ETSI EN 303 387 V1.1.1 (2015-11)

Reconfigurable Radio Systems (RRS);
Signalling Protocols and information exchange
for Coordinated use of TV White Spaces;
Interface between Cognitive Radio System (CRS)
and Spectrum Coordinator (SC)

Reference
DEN/RRS-0142

Keywords
control, CRS, performance, white space

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

Important notice

The present document can be downloaded from:
http://www.etsi.org/standards-search

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the only prevailing document is the print of the Portable Document Format (PDF) version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status.
Information on the current status of this and other ETSI documents is available at
http://portal.etsi.org/tb/status/status.asp

If you find errors in the present document, please send your comment to one of the following services:
https://portal.etsi.org/People/CommiteeSupportStaff.aspx

Copyright Notification

No part may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm except as authorized by written permission of ETSI.
The content of the PDF version shall not be modified without the written authorization of ETSI.
The copyright and the foregoing restriction extend to reproduction in all media.

© European Telecommunications Standards Institute 2015.
All rights reserved.

DECT™, **PLUGTESTS**™, **UMTS**™ and the ETSI logo are Trade Marks of ETSI registered for the benefit of its Members.
3GPP™ and **LTE**™ are Trade Marks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners.
GSM® and the GSM logo are Trade Marks registered and owned by the GSM Association.

Contents

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 European Standard (EN) has been produced by ETSI Technical Committee Reconfigurable Radio Systems (RRS).

National transposition dates	
Date of adoption of this EN:	17 November 2015
Date of latest announcement of this EN (doa):	29 February 2016
Date of latest publication of new National Standard or endorsement of this EN (dop/e):	31 August 2016
Date of withdrawal of any conflicting National Standard (dow):	31 August 2016

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**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.

1 Scope

The present document provides the parameters and procedures for such information exchange between Cognitive Radio System (CRS) and Spectrum Coordinator (SC) in Coordinated use of TV White Spaces. This work is based on the system architecture and high level procedures for coordinated and uncoordinated use of TV White Spaces as defined in ETSI EN 303 145 [i.1].

2 References

2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the reference document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at http://docbox.etsi.org/Reference.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are necessary for the application of the present document.

Not applicable.

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the reference document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] ETSI EN 303 145: "Reconfigurable Radio Systems (RRS); System Architecture and High Level Procedures for Coordinated and Uncoordinated Use of TV White Spaces".

[i.2] ETSI EN 301 598: "White Space Devices (WSD); Wireless Access Systems operating in the 470 MHz to 790 MHz TV broadcast band; Harmonized EN covering the essential requirements of article 3.2 of the R&TTE Directive".

[i.3] IETF PAWS: "Protocol to Access White-Space (PAWS) Databases", (draft-ietf-paws-protocol-20).

[i.4] ECC Report 159 - January 2011: "Technical and operational requirements for the possible operation of cognitive radio systems in the "white spaces" of the frequency band 470-790 MHz".

[i.5] ISO/IEC 10731 (1994): "Information Technology - Open Systems Interconnection - Basic Reference Model: Conventions for the Definition of OSI Services".

[i.6] Recommendation ITU-T X.680: "Information technology - Abstract Syntax Notation One (ASN.1): Specification of basic notation".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in ETSI EN 303 145 [i.1], ECC Report 159 [i.4] and the following apply:

channel usage parameters: parameters sent by a CRS to inform the spectrum coordinator (SC) / Geo-location database (GLDB) of the actual radio resources that will be used by CRSs

Cognitive Radio System (CRS): white spaces device (WSD) or network of WSDs (i.e. a master WSD and some slave WSDs)

device parameters: parameters that specify the technical characteristics of an individual CRS, and its location

Geo-location Database (GLDB): entity whose operation is mandated or authorized by a regulatory authority

NOTE: It also provides a WSD in a CRS with location specific information on the available frequencies and associated maximum EIRP values that the WSD is permitted to use which allow for protection of the incumbent service and are derived from information provided by the WSD and the minimum required Adjacent Channel Leakage Ratio (ACLR) of the WSD. The GLDB consists of database and geo-location functions.

operational parameters: transmission parameters communicated from SC/GLDB to a CRS to allow the CRS to operate as required

Spectrum Coordinator (SC): entity that coordinates spectrum usage of CRS based on the information obtained from geo-location database as well as supplemental spectrum usage data from different CRSs using its service

White Space Device (WSD): WSD controlled by a SC/GLDB and which operates in white spaces

3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

ACLR	Adjacent Channel Leakage Ratio
ACS	Adjacent Channel Selectivity
ASN	Abstract SyntaxNotation
CRS	Cogitative Radio System
EIRP	Effective Isotropic Radiated Power
GLDB	Geo-location Database
ID	IDentifier
LTE	Long Term Evolution
PHY	Physical layer
SAP	Service Access Point
SC	Spectrum Coordinator
SCGLDB	Spectrum Coordinator and Geo-location Database
SC-SAP	Spectrum Coordinator Service Access Point
TV	TeleVision
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
WSD	White Space Device

4 Service access points

4.1 General

This following clauses describe the primitives related to the Service Access Points as identified in ETSI EN 303 145 [i.1] and also shown in Figure 4.1 below.

Clause 4.2 describes the primitives associated to the control function (C-SAP in Figure 4.1).

Clause 4.3 describes the primitives associated to the spectrum coordination function (SC-SAP in Figure 4.1).

Clause 4.4 describes the primitives associated to the communication function (Com-SAP in Figure 4.1).

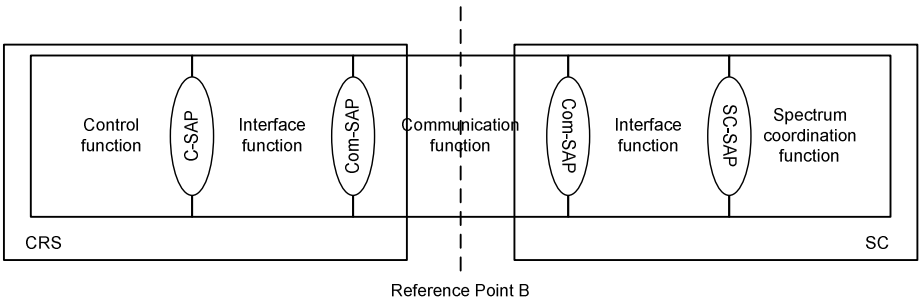

Figure 4.1: Reference model for the information exchange between a CRS and a SC (Reference point B) [i.1]

4.2 Control Service Access Point (C-SAP)

4.2.1 Initialization

The initialization service provides a set of primitives or method through which the control function provides the device parameters of a CRS and obtains initialization of a CRS.

Initialization.request

Function

This primitive is used by the control function to request initialization of a CRS.

Semantics of the service primitive

Initialization.request(
initializationID,
deviceDescriptor,
geolocation,
deviceCapabilities
)

Parameters

Name	Type	Description
initializationID	TransactionID	initializationID uniquely identifies one transaction of requesting initialization for a CRS.
deviceDescriptor	DeviceDescriptor	Device descriptor describes the physical profile of a CRS.
geolocation	Geolocation	Geo-location information of CRS.
deviceCapabilities	DeviceCapabilities	Device capabilities provide the additional information needed for initialization.

When used

This primitive shall be used by the control function when a CRS needs to provide CRS information as described by the physical profile, geo-location and device capabilities of CRS.

Effect of receipt

The interface function receives the request for device parameters and geo-location information of a CRS.

Initialization.response

Function

This primitive is used by the interface function to provide the results of initialization to the control function as a response to the *Initialization.request* primitive.

Semantics of the service primitive

Initialization.response(
initializationID,
rulesetInformation,
scgldbInformation
)

Parameters

Name	Type	Description
initializationID	TransactionID	initializationID uniquely identifies one transaction of requesting initialization for a CRS.
rulesetInformation	RulesetInformation	Ruleset Information contains parameters for the ruleset of a regulatory domain described in [i.3].
scgldbInformation	SCGLDBInformation	SCGLDB Information contains list of name and URI of SC and GLDB.

When used

This primitive shall be used by the interface function when it receives the request of initialization.

Effect of receipt

The control function receives the response (e.g. ruleset information of a regulatory domain, the list of name and URI of SC and GLDB, etc.) of the initialization request of a CRS.

4.2.2 Subscription

The subscription service provides a set of primitives or method through which the control function provides the subscription request of a CRS and obtains subscription of a CRS.

Service_Subscription.request

Function

This primitive is used by the control function to request subscription of a CRS.

Semantics of the service primitive

Service_Subscription.request(
subscriptionID,
subscriptionRequest
)

Parameters

Name	Type	Description
subscriptionID	TransactionID	subscriptionID uniquely identifies one transaction of requesting subscription for a CRS.
subscriptionRequest	SubscriptionRequest	CRS subscription to coordination service (management or information).

When used

This primitive shall be used by the control function when a CRS needs to request its subscription as described by the subscription request.

Effect of receipt

The interface function receives the request for subscription of a CRS.

Service_Subscription.response

Function

This primitive is used by the interface function to provide the results of the subscription to the control function as a response to the *Service_Subscription.request* primitive.

Semantics of the service primitive

Service_Subscription.response(
subscriptionID,
status
)

Parameters

Name	Type	Description
subscriptionID	TransactionID	subscriptionID uniquely identifies one transaction of requesting subscription for a CRS.
status	Status	Status of subscription.

When used

This primitive shall be used by the interface function when it receives the status of a subscription.

Effect of receipt

The control function receives the results of subscription of a CRS.

4.2.3 Subscription update

The subscription update service provides a set of primitives or method through which the control function provides the subscription update request of a CRS and obtains subscription update of a CRS.

Service_Subscription_Update.request

Function

This primitive is used by the control function to request service subscription update of a CRS.

Semantics of the service primitive

Service_Subscription_Update.request(
subscriptionUpdateID,
subscriptionRequest
)

Parameters

Name	Type	Description
subscriptionUpdateID	TransactionID	subscriptionUpdateID uniquely identifies one transaction of updating service subscription for a CRS.
subscriptionRequest	SubscriptionRequest	CRS subscription to coordination service (management, information or no service).

When used

This primitive shall be used by the control function when a CRS needs to request its subscription update as described by the subscription update request.

Effect of receipt

The interface function receives the request for subscription update of a CRS.

Service_Subscription_Update.response

Function

This primitive is used by the interface function to provide the results of the subscription update to the control function as a response to the *Service_Subscription_Update.request* primitive.

Semantics of the service primitive

Service_Subscription_Update.response(
subscriptionUpdateID,
status
)

Parameters

Name	Type	Description
subscriptionUpdateID	TransactionID	subscriptionUpdateID uniquely identifies one transaction of updating service subscription for a CRS.
status	Status	Status of the subscription update.

When used

This primitive shall be used by the interface function when it receives the status of the subscription update.

Effect of receipt

The control function receives the result of the subscription update of a CRS.

4.2.4 Subscription change

The subscription change service provides a set of primitives or method through which the interface function provides the subscription change of a CRS.

Service_Subscription_Change.request

Function

This primitive is used by the interface function to request a subscription change of a CRS.

Semantics of the service primitive

Service_Subscription_Change.request(
subscriptionChangeID,
subscriptionChangeRequest
)

Parameters

Name	Type	Description
subscriptionChangeID	TransactionID	subscriptionChangeID uniquely identifies one transaction of requesting subscription change for a CRS.
subscriptionChangeRequest	SubscriptionChangeRequest	CRS subscription to coordination service (management or information).

When used

This primitive shall be used by the interface function when it needs to request the subscription change for a CRS as described by the subscription change request.

Effect of receipt

The control function receives the request for subscription change of a CRS.

Service_Subscription_Change.response

Function

This primitive is used by the control function to provide the result of service a subscription change to the interface function as a response to the *Service_Subscription_Change.request* primitive.

Semantics of the service primitive

Service_Subscription_Change.response(
subscriptionChangeID,
status
)

Parameters

Name	Type	Description
subscriptionChangeID	TransactionID	subscriptionChangeID uniquely identifies one transaction of requesting subscription change for a CRS.
status	Status	Status of subscription change.

When used

This primitive shall be used by the control function to generate a response to the subscription change request.

Effect of receipt

The interface function receives the result of subscription change request of a CRS.

4.2.5 Registration

The registration service provides a set of primitives or method through which the control function provides the device parameters and geo-location information of a CRS and obtains the registration of a CRS.

Network_Registration.request

Function

This primitive is used by the control function to request registration of a CRS.

Semantics of the service primitive

Network_Registration.request(
registrationID,
deviceDescriptor,
geolocation,
deviceCharacteristics
)

Parameters

Name	Type	Description
registrationID	TransactionID	registrationID uniquely identifies one transaction of requesting registration for a CRS.
deviceDescriptor	DeviceDescriptor	Device descriptor describes the physical profile of a CRS.
geolocation	Geolocation	Geo-location information of CRS.
deviceCharacteristics	DeviceCharacteristics	Device characteristics provide CRS's device information (e.g. antenna, installation, etc.).

When used

This primitive shall be used by the control function when a CRS needs to request its registration as described by the physical profile, geo-location and characteristics of CRS.

Effect of receipt

The interface function receives the request for registration of a CRS.

Network_Registration.response

Function

This primitive is used by the interface function to provide the results of registration of a CRS to the control function as a response to the *Network_Registration.request* primitive.

Semantics of the service primitive

Network_Registration.response(
registrationID,
rulesetInformation,
scgldbInformation
)

Parameters

Name	Type	Description
registrationID	TransactionID	registrationID uniquely identifies one transaction of requesting registration for a CRS.
rulesetInformation	RulesetInformation	Ruleset Information contains parameters for the ruleset of a regulatory domain described in [i.3].
scgldbInformation	SCGLDBInformatoin	SCGLDB Information contains list of name and URI of SC and GLDB.

When used

This primitive shall be used by the interface function when it receives the status of registration.

Effect of receipt

The control function receives the result (e.g. ruleset information of a regulatory domain, the list of name and URI of SC and GLDB, etc.) of registration of a CRS.

4.2.6 Registration update

The CRS registration update service provides a set of primitives or method through which the control function provides the device parameters and geo-location information of a CRS and obtains the registration of a CRS.

Network_Registration_Update.request

Function

This primitive is used by the control function to request registration update of a CRS.

Semantics of the service primitive

Network_Registration_Update.request(
registrationUpdateID,
deviceDescriptor,
geolocation,
deviceCharacteristics
)

Parameters

Name	Type	Description
registrationUpdateID	TransactionID	registrationUpdateID uniquely identifies one transaction of requesting registration update for a CRS.
deviceDescriptor	DeviceDescriptor	Device descriptor describes the physical profile of a CRS.
geolocation	Geolocation	Geo-location information of CRS.
deviceCharacteristics	DeviceCharacteristics	Device characteristics provide CRS's device information (e.g. antenna, installation, etc.).

When used

This primitive shall be used by the control function when a CRS needs to request its registration as described by the physical profile, geo-location and characteristics of CRS.

Effect of receipt

The interface function receives the request for registration of a CRS.

Network_Registration_Update.response

Function

This primitive is used by the interface function to provide the results of registration update of a CRS to the control function as a response to the *Network_Registration_Update.request* primitive.

Semantics of the service primitive

Network_Registration_Update.response(
registrationUpdateID,
rulesetInformation,
scgldbInformation
)

Parameters

Name	Type	Description
registrationUpdateID	TransactionID	registrationUpdateID uniquely identifies one transaction of requesting registration update for a CRS.
rulesetInformation	RulesetInformation	Ruleset Information contains parameters for the ruleset of a regulatory domain described in [i.3].
scgldbInformation	SCGLDBInformatoin	SCGLDB Information contains list of name and URI of SC and GLDB.

When used

This primitive shall be used by the interface function when it receives the status of registration update.

Effect of receipt

The control function receives the result of registration update of a CRS.

4.2.7 Requesting channel access

The requesting channel access service provides a set of primitives or method through which the control function provides the device parameters and geo-location information of a CRS and requests spectrum usage for a CRS.

Coordinated_Channel.request

Function

This primitive is used by the control function to request spectrum usage for a CRS.

Semantics of the service primitive

Coordinated_Channel.request(
channelAccessID,
deviceDescriptor,
locationInfo,
deviceCharacteristics,
deviceCapabilities
)

Parameters

Name	Type	Description
channelAccessID	TransactionID	channelAccessID uniquely identifies one transaction of requesting the channel for a CRS.
deviceDescriptor	DeviceDescriptor	Device descriptor describes the physical profile of a CRS.
locationInfo	LocationInfo	One or multiple geo-location information of CRS or an active region of CRS.
deviceCharacteristics	DeviceCharacteristics	Device characteristics provide CRS's device information (e.g. antenna, installation, etc.).
deviceUsageRequirements	DeviceUsageRequirements	Device usage requirements provide the information that be used by CRS to provide additional information to the SC and/or GLDB that can help it to determine operational parameters.

When used

This primitive shall be used by the control function when a CRS needs to obtain spectrum usage as described by the physical profile, geo-location, characteristics and usage requirements of CRS.

Effect of receipt

The interface function receives the channel request of a CRS.

Coordinated_Channel.response

Function

This primitive is used by the interface function to provide the confirmation of the channel request of a CRS to the control function as a confirmation to the *Coordinated_Channel.request* primitive.

Semantics of the service primitive

Coordinated_Channel.response(
channelAccessID,
status
)

Parameters

Name	Type	Description
channelAccessID	TransactionID	channelAccessID uniquely identifies one transaction of requesting the channel for a CRS.
status	Status	Status of channel request.

When used

This primitive shall be used by the interface function when it receives the status of channel request.

Effect of receipt

The control function receives the result of the channel request of a CRS.

4.2.8 Providing available channel list

Coordinated_Available_Channel.indication

Function

This primitive is used by the interface function to indicate available channel of a CRS.

Semantics of the service primitive

Coordinated_Available_Channel.indication(
availableChannelID,
operationalParameters
)

Parameters

Name	Type	Description
availableChannelID	TransactionID	availableChannelID uniquely identifies one transaction of indicating available channel for a CRS.
operationalParameters	OperationalParameters	Operational parameters describe the available spectrum for a CRS and any additional information to allow the CRS to utilize the spectrum.

When used

This primitive shall be used by the interface function when it needs to indicate the available spectrum for a CRS as described by the operational parameters.

Effect of receipt

The control function receives the indication for available channels of a CRS.

Coordinated_Channel_Usage.response

Function

This primitive is used by the control function to provide the channel usage parameters to the interface function as a response to the *Coordinated_Available_Channel.indication* primitive.

Semantics of the service primitive

Coordinated_Channel_Usage.response(
availableChannelID,
deviceDescriptor,
geolocation,
channelUsageParameters
)

Parameters

Name	Type	Description
availableChannelID	TransactionID	availableChannelID uniquely identifies one transaction of indicating available channel for a CRS.
deviceDescriptor	DeviceDescriptor	Device descriptor describes the physical profile of a CRS.
geolocation	Geolocation	Geo-location information of CRS.
channelUsageParameters	ChannelUsageParameters	Channel usage parameters describe the operation information of a CRS.

When used

This primitive shall be used by the control function to generate channel usage parameters as a response to *Coordinated_Available_Channel.indication.*

Effect of receipt

The interface function receives the channel usage parameters of a CRS.

4.2.9 Reconfiguration

The reconfiguration service provides a set of primitives or method through which the interface function provides the operational parameters of a CRS.

Reconfiguration.request

Function

This primitive is used by the interface function to request the reconfiguration of a CRS.

Semantics of the service primitive

Reconfiguration.request(
reconfigurationID,
operationalParameters
)

Parameters

Name	Type	Description
reconfigurationID	TransactionID	reconfigurationID uniquely identifies one transaction of requesting the reconfiguration for a CRS.
operationalParameters	OperationalParameters	Operational parameters describe the available spectrum of a CRS and any additional information to allow the CRS to use it.

When used

This primitive shall be used by the interface function when it needs to request the reconfiguration of a CRS.

Effect of receipt

The control function receives the request of the reconfiguration of a CRS.

Reconfiguration.response

Function

This primitive is used by the control function to provide the result of the reconfiguration of a CRS to the interface function as a response to the *Reconfiguration.request* primitive.

Semantics of the service primitive

Reconfiguration.response(
reconfigurationID,
status
)

Parameters

Name	Type	Description
reconfigurationID	TransactionID	reconfigurationID uniquely identifies one transaction of requesting the channel for a CRS.
status	Status	Status of reconfiguration request.

When used

This primitive shall be used by the control function when it receives the reconfiguration request.

Effect of receipt

The interface function receives the result of the reconfiguration of a CRS.

4.2.10 Providing coordination report

Coordination_Report.indication

Function

This primitive is used by the interface function to send the coordination report to a CRS.

Semantics of the service primitive

Coordination_Report.indication(
coordinationReportID,
coordinationReport
)

Parameters

Name	Type	Description
coordinationReportID	TransactionID	coordinationReportID uniquely identifies one transaction of indicating coordination report for a CRS.
coordinationReport	CoordinationReport	Coordination report describes the information needed by the CRS to allow it to make decisions related to its operational parameters which allow coexistence.

When used

This primitive shall be used by the interface function when it needs to send the coordination report to a CRS.

Effect of receipt

The control function receives the coordination report.

Coordination_Report.response

Function

This primitive is used by the control function to confirm receiving the coordination report to the interface function as a response to the *Coordination_Report.indication* primitive.

Semantics of the service primitive

Coordination_Report.response(
coordinationReportID,
status
)

Parameters

Name	Type	Description
coordinationReportID	TransactionID	coordinationReportID uniquely identifies one transaction of indicating coordination report for a CRS.
status	Status	Status of coordination report indication.

When used

This primitive shall be used by the control function to generate a response to the *Coordination_Report.indication*.

Effect of receipt

The interface function receives the result of the coordination report indication of a CRS.

4.2.11 Operational parameters update

The operational parameters update service provides a set of primitives or method through which the interface function provides the request of the operational parameters update of a CRS.

Operational_Parameters_Update.request

Function

This primitive is used by the interface function to request the operational parameters update of a CRS.

Semantics of the service primitive

Operational_Parameters_Update.request(
operationalParametersUpdateID,
validTime
)

Parameters

Name	Type	Description
operational ParametersUpdateID	TransactionID	operationalParametersUpdateID uniquely identifies one transaction of requesting the operational parameters update for a CRS.
validTime	ValidTime	ValidTime describes the time limit for the update of CRS's operational parameters.

When used

This primitive shall be used by the interface function when it needs to request the operational parameters update of a CRS.

Effect of receipt

The control function receives the request of the operational parameter update of a CRS.

Operational_Parameters_Update.response

Function

This primitive is used by the control function to provide the response of the operational parameters update of a CRS to the interface function as a response to the *Operational_Parameters_Update.request* primitive.

Semantics of the service primitive

Operational_Parameters_Update.response(
operationalParametersUpdateID,
status
)

Parameters

Name	Type	Description
operationalParamtersUpdateID	TransactionID	operationalParametersUpdateID uniquely identifies one transaction of requesting the operational parameters update for a CRS.
status	Status	Status of operational parameters update request.

When used

This primitive shall be used by the control function when it receives the request of operational parameters update.

Effect of receipt

The interface function receives the result of the request of the operational parameters update.

4.2.12 Requesting measurement

The measurement request service provides a set of primitives or method through which the interface function provides the sensing and measurement request of a CRS.

Measurement.request

Function

This primitive is used by the interface function to request sensing and measurement of a CRS.

Semantics of the service primitive

Measurement.request(
measurementID,
measurementType,
measurementConfiguration,
reportConfiguration
)

Parameters

Name	Type	Description
measurementID	TransactionID	measurementID uniquely identifies one transaction of requesting sensing and measurement of a CRS.
measurementType	MeasurementType	Measurement type describes frequency of measurement report (single, periodic) as well as the type of measurement that needs to be taken (e.g. PHY layer sensing, QoS, etc.).
measurementConfiguration	MeasurementConfiguration	Measurement configuration provides the information that used by SC to provide measurement configuration to the CRS.
reportConfiguration	ReportConfiguration	Format and configuration of measurement report.

When used

This primitive shall be used by the interface function when the SC requests the CRS to perform measurements of a specific type, and with a specific timing, frequency, or condition.

Effect of receipt

The control function receives the measurements request for a specific TransactionID and configures/starts measurements or sensing in the CRS associated with the measurementType and measurementConfiguration associated with the request.

Measurement.response

Function

This primitive is used by the control function as a confirmation to the *Measurement.request* primitive.

Semantics of the service primitive

Measurement.response(
measurementID,
status,
)

Parameters

Name	Type	Description
measurementID	TransactionID	measurementID uniquely identifies one transaction of requesting sensing and measurement of a CRS.
status	Status	Status of measurement request.

When used

This primitive shall be used by the control function to generate a confirmation of the measurement request as *Measurement.request*.

Effect of receipt

The interface function receives the result of measurement request of a CRS.

4.2.13 Periodic measurement

The periodic measurement service provides a set of primitives or method through which the control function provides the sensing and measurement report of a CRS based on schedule.

MeasurementResults.indication

Function

This primitive is used by the control function to provide the sensing and measurement report of a CRS.

Semantics of the service primitive

MeasurementResults.indication(
measurementResultsID,
geolocation
measurementReport
)

Parameters

Name	Type	Description
measurementResultsID	TransactionID	measurementResultsID uniquely identifies one transaction of providing the sensing and measurement results of a CRS.
geolocation	Geolocation	Geo-location information of CRS.
measurementReport	MeasurementReport	Measurement Report describes the results of the sensing and measurement of a CRS.

When used

This primitive shall be used by the control function when it needs to provide the sensing and measurement report of a CRS.

Effect of receipt

The interface function receives the measurement report of a CRS.

MeasurementResults.response

Function

This primitive is used by the interface function as a response to the *MeasurementResults.indication* primitive.

Semantics of the service primitive

MeasurementResults.response(
measurementResultsID,
measurementAction,
status,
)

Parameters

Name	Type	Description
measurementResultsID	TransactionID	measurementRequestID uniquely identifies one transaction of providing the sensing and measurement results of a CRS.
measurementAction	MeasurementAction	measurementAction indicates whether measurements should continue following the current measurements period or stop measurements.
status	Status	Status of measurement results.

When used

This primitive shall be used by the interface function to generate a confirmation of receiving the measurement report as *MeasurementReport.indication*.

Effect of receipt

The control function receives the result of receiving the measurement report from SC.

4.2.14 Single measurement

The single measurement service provides a set of primitives or method through which the control function provides the sensing and measurement report of a CRS once.

MeasurementResults.indication

Function

This primitive is used by the control function to provide the sensing and measurement report of a CRS.

Semantics of the service primitive

MeasurementResults.indication(
measurementResultsID,
geolocation
measurementReport
)

Parameters

Name	Type	Description
measurementResultsID	TransactionID	measurementResultsID uniquely identifies one transaction of providing the sensing and measurement results of a CRS.
geolocation	Geolocation	Geo-location information of CRS.
measurementReport	MeasurementReport	Measurement Report describes the results of the sensing and measurement of a CRS.

When used

This primitive shall be used by the control function when it needs to provide the sensing and measurement report of a CRS.

Effect of receipt

The interface function receives the measurement report of a CRS.

MeasurementResults.response

Function

This primitive is used by the interface function as a response to the *MeasurementResults.indication* primitive.

Semantics of the service primitive

MeasurementResults.response(
measurementResultsID,
status,
)

Parameters

Name	Type	Description
measurementResultsID	TransactionID	measurementResultsID uniquely identifies one transaction of providing the sensing and measurement results of a CRS.
status	Status	Status of measurement results.

When used

This primitive shall be used by the interface function to generate a confirmation of receiving the measurement report as *MeasurementReport.indication*.

Effect of receipt

The control function receives the result of receiving the measurement report from SC.

4.2.15 Device parameters reconfiguration

The device parameter reconfiguration service provides a set of primitives or method through which the interface function provides the suggested device parameters of a CRS.

Device_Parameter_Reconfiguration.request

Function

This primitive is used by the interface function to request the reconfiguration of device parameters of a CRS.

Semantics of the service primitive

Device_Parameter_Reconfiguration.request(
deviceParameterReconfigurationID,
suggestedDeviceParameters
)

Parameters

Name	Type	Description
deviceParameterReconfigurationID	TransactionID	deviceParameterReconfigurationID uniquely identifies one transaction of requesting device parameter reconfiguration for a CRS.
suggestedDeviceParameters	DeviceParameters	suggestedDeviceParameters describe the SC-suggested device parameters of a CRS.

When used

This primitive shall be used by the interface function when it needs to request the device parameter reconfiguration of a CRS.

Effect of receipt

The control function receives the device parameters reconfiguration request.

Device_Parameter_Reconfiguration.response

Function

This primitive is used by the control function to provide the result of the device parameters reconfiguration of a CRS to the interface function as a response to the *Device_Parameter_Reconfiguration.request* primitive.

Semantics of the service primitive

Device_Parameter_Reconfiguration.response(
deviceParameterReconfigurationID,
status
)

Parameters

Name	Type	Description
deviceParameterReconfigurationID	TransactionID	deviceParameterReconfigurationID uniquely identifies one transaction of requesting the device parameter reconfiguration for a CRS.
status	Status	Status of device parameters reconfiguration.

When used

This primitive shall be used by the control function when it receives the device parameter reconfiguration request.

Effect of receipt

The interface function receives the result of the device parameter reconfiguration of a CRS.

4.3 Spectrum Coordination Service Access Point (SC-SAP)

4.3.1 CRS initialization

The CRS initialization service provides a set of primitives or method through which the interface function provides the device parameters of a CRS and obtains the information for initialization of a CRS.

CRS_Initialization.request

Function

This primitive is used by the interface function to request initialization from a CRS to spectrum coordination function.

Semantics of the service primitive

CRS_Initialization.request(
cRSInitializationID,
deviceDescriptor,
geolocation,
deviceCapabilities
)

Parameters

Name	Type	Description
cRSInitializationID	TransactionID	cRSInitializationID uniquely identifies one transaction of requesting initialization from a CRS.
deviceDescriptor	DeviceDescriptor	Device descriptor describes the physical profile of a CRS.
geolocation	Geolocation	Geo-location information of CRS.
deviceCapabilities	DeviceCapabilities	Device capabilities provide the additional information needed for initialization.

When used

This primitive shall be used by the interface function when it needs to provide CRS information as described by the physical profile, geo-location and device capabilities of CRS.

Effect of receipt

The spectrum coordination function receives the request for initialization of a CRS.

CRS_Initialization.response

Function

This primitive is used by the spectrum coordination function to provide the results of initialization to the interface function as a response to the *CRS_Initialization.request* primitive.

Semantics of the service primitive

CRS_Initialization.response(
cRSInitializationID,
rulesetInformation,
scgldbInformation
)

Parameters

Name	Type	Description
cRSInitializationID	TransactionID	cRSInitializationID uniquely identifies one transaction of requesting initialization from a CRS.
rulesetInformation	RulesetInformation	Ruleset Information contains parameters for the ruleset of a regulatory domain described in [i.3].
scgldbInformation	SCGLDBInformatoin	SCGLDB Information contains list of name and URI of SC and GLDB.

When used

This primitive shall be used by the spectrum coordination function when it receives the request of initialization from a CRS.

Effect of receipt

The interface function receives the response (e.g. ruleset information of a regulatory domain, the list of name and URI of SC and GLDB, etc.) of the initialization request of a CRS from the spectrum coordination function.

4.3.2 CRS subscription

The CRS subscription service provides a set of primitives or method through which the interface function provides the subscription request of a CRS and obtains subscription of a CRS to the spectrum coordinator function.

CRS_Service_Subscription.request

Function

This primitive is used by the interface function to request subscription of a CRS.

Semantics of the service primitive

CRS_Service_Subscription.request(
cRSSubscriptionID,
subscriptionRequest
)

Parameters

Name	Type	Description
cRSSubscriptionID	TransactionID	cRSSubscriptionID uniquely identifies one transaction of requesting subscription for a CRS.
subscriptionRequest	SubscriptionRequest	CRS subscription to coordination service (management or information).

When used

This primitive shall be used by the interface function when it needs to request CRS subscription as described by the subscription request.

Effect of receipt

The spectrum coordination function receives the request for subscription of a CRS.

CRS_Service_Subscription.response

Function

This primitive is used by the spectrum coordination function to provide the results of the subscription to the interface function as a response to the *CRS_Service_Subscription.request* primitive.

Semantics of the service primitive

CRS_Service_Subscription.request(
cRSSubscriptionID,
status
)

Parameters

Name	Type	Description
cRSSubscriptionID	TransactionID	cRSSubscriptionID uniquely identifies one transaction of requesting subscription for a CRS.
status	Status	Status of subscription.

When used

This primitive shall be used by the spectrum coordination function when it receives the request of a subscription from a CRS.

Effect of receipt

The interface function receives the result of subscription of a CRS from the spectrum coordination function.

4.3.3 CRS subscription update

The CRS subscription update service provides a set of primitives or method through which the interface function provides the subscription update request of a CRS and obtains subscription update of a CRS.

CRS_Service_Subscription_Update.request

Function

This primitive is used by the interface function to request service subscription update from a CRS.

Semantics of the service primitive

CRS_Service_Subscription_Update.request(
cRSSubscriptionUpdateID,
subscriptionRequest
)

Parameters

Name	Type	Description
cRSSubscriptionUpdateID	TransactionID	cRSSubscriptionUpdateID uniquely identifies one transaction of updating service subscription for a CRS.
subscriptionRequest	SubscriptionRequest	CRS subscription to coordination service (management, information or no service).

When used

This primitive shall be used by the interface function when it receives the CRS subscription update request as described by the subscription update request from a CRS.

Effect of receipt

The spectrum coordination function receives the request for subscription update from a CRS.

CRS_Service_Subscription_Update.response

Function

This primitive is used by the spectrum coordination function to provide the results of the subscription update to the interface function as a response to the *CRS_Service_Subscription_Update.request* primitive.

Semantics of the service primitive

CRS_Service_Subscription_Update.response(
cRSSubscriptionUpdateID,
status
)

Parameters

Name	Type	Description
cRSSubscriptionUpdateID	TransactionID	cRSSubscriptionUpdateID uniquely identifies one transaction of updating service subscription for a CRS.
status	Status	Status of the subscription update.

When used

This primitive shall be used by the spectrum coordination function when it receives the status of the subscription update.

Effect of receipt

The interface function receives the status of the subscription update of a CRS from the spectrum coordination function.

4.3.4 CRS subscription change

The CRS subscription change service provides a set of primitives or method through which the spectrum coordination function provides the subscription change of a CRS.

CRS_Service_Subscription_Change.request

Function

This primitive is used by the spectrum coordination function to request a subscription change of a CRS.

Semantics of the service primitive

CRS_Service_Subscription_Change.request(
cRSSubscriptionChangeID,
subscriptionChangeRequest
)

Parameters

Name	Type	Description
cRSSubscriptionChangeID	TransactionID	cRSSubscriptionChangeID uniquely identifies one transaction of requesting subscription change for a CRS.
subscriptionChangeRequest	SubscriptionChangeRequest	CRS subscription to coordination service (management or information).

When used

This primitive is used by the spectrum coordination function when it needs to request the subscription change for a CRS as described by the subscription change request.

Effect of receipt

The interface function receives the request for subscription change of a CRS.

CRS_Service_Subscription_Change.response

Function

This primitive is used by the interface function to provide the result of service a subscription change to the spectrum coordination function as a response to the *CRS_Service_Subscription_Change.request* primitive.

Semantics of the service primitive

CRS_Service_Subscription_Change.response(
cRSSubscriptionChangeID,
status
)

Parameters

Name	Type	Description
cRSSubscriptionChangeID	TransactionID	cRSSubscriptionChangeID uniquely identifies one transaction of requesting subscription change for a CRS.
status	Status	Status of subscription change.

When used

This primitive shall be used by the interface function to generate a response to the subscription change request.

Effect of receipt

The spectrum coordination function receives the result of subscription change request of a CRS.

4.3.5 CRS registration

The CRS registration service provides a set of primitives or method through which the interface function provides the device parameters and geo-location information of a CRS and obtains the registration of a CRS.

CRS_Network_Registration.request

Function

This primitive is used by the interface function to request registration of a CRS.

Semantics of the service primitive

CRS_Network_Registration.request(
cRSRegistrationID,
deviceDescriptor,
geolocation,
deviceCharacteristics
)

Parameters

Name	Type	Description
cRSRegistrationID	TransactionID	cRSRegistrationID uniquely identifies one transaction of requesting registration for a CRS.
deviceDescriptor	DeviceDescriptor	Device descriptor describes the physical profile of a CRS.
geolocation	Geolocation	Geo-location information of CRS.
deviceCharacteristics	DeviceCharacteristics	Device characteristics provide CRS's device information (e.g. antenna, installation, etc.).

When used

This primitive shall be used by the interface function when it needs to request its registration as described by the physical profile, geo-location and characteristics of CRS.

Effect of receipt

The spectrum coordination function receives the request for registration from a CRS.

CRS_Network_Registration.response

Function

This primitive is used by the spectrum coordination function to provide the results of registration of a CRS to the interface function as a response to the *CRS_Network_Registration.request* primitive.

Semantics of the service primitive

CRS_Network_Registration.response(
cRSRegistrationID,
rulesetInformation,
scgldbInformation
)

Parameters

Name	Type	Description
cRSRegistrationID	TransactionID	cRSRegistrationID uniquely identifies one transaction of requesting registration for a CRS.
rulesetInformation	RulesetInformation	Ruleset Information contains parameters for the ruleset of a regulatory domain described in [i.3].
scgldbInformation	SCGLDBInformatoin	SCGLDB Information contains list of name and URI of SC and GLDB.

When used

This primitive shall be used by the spectrum coordination function when it receives the status of registration.

Effect of receipt

The interface function receives the result (e.g. ruleset information of a regulatory domain, the list of name and URI of SC and GLDB, etc.) of registration of a CRS.

4.3.6 CRS registration update

The CRS registration update service provides a set of primitives or method through which the interface function provides the device parameters and geo-location information of a CRS and obtains the registration of a CRS.

CRS_Network_Registration_Update.request

Function

This primitive is used by the interface function to request registration update of a CRS.

Semantics of the service primitive

```
CRS_Network_Registration_Update.request(
                                        cRSRegistrationUpdateID,
                                        deviceDescriptor,
                                        geolocation,
                                        deviceCharacteristics
                                        )
```

Parameters

Name	Type	Description
cRSRegistrationUpdateID	TransactionID	cRSRegistrationUpdateID uniquely identifies one transaction of requesting registration update for a CRS.
deviceDescriptor	DeviceDescriptor	Device descriptor describes the physical profile of a CRS.
geolocation	Geolocation	Geo-location information of CRS.
deviceCharacteristics	DeviceCharacteristics	Device characteristics provide CRS's device information (e.g. antenna, installation, etc.).

When used

This primitive is used by the interface function when a CRS needs to request its registration as described by the physical profile, geo-location and characteristics of CRS.

Effect of receipt

The spectrum coordination function receives the request for registration of a CRS.

CRS_Network_Registration_Update.response

Function

This primitive is used by the spectrum coordination function to provide the results of registration update of a CRS to the interface function as a response to the *CRS_Network_Registration_Update.request* primitive.

Semantics of the service primitive

```
CRS_Network_Registration_Update.response(
                                        cRSRegistrationUpdateID,
                                        rulesetInformation,
                                        scgldbInformation
                                        )
```

Parameters

Name	Type	Description
cRSRegistrationUpdateID	TransactionID	cRSRegistrationUpdateID uniquely identifies one transaction of requesting registration update for a CRS.
rulesetInformation	RulesetInformation	Ruleset Information contains parameters for the ruleset of a regulatory domain described in [i.3].
scgldbInformation	SCGLDBInformatoin	SCGLDB Information contains list of name and URI of SC and GLDB.

When used

This primitive shall be used by the spectrum coordination function when it receives the status of registration update.

Effect of receipt

The interface function receives the result of registration update of a CRS.

4.3.7 Requesting CRS channel access

The requesting CRS channel access service provides a set of primitives or method through which the interface function provides the device parameters and geo-location information of a CRS and requests spectrum usage for a CRS.

CRS_Coordinated_Channel.request

Function

This primitive is used by the interface function to request spectrum usage from a CRS.

Semantics of the service primitive

```
CRS_Coordinated_Channel.request(
                                cRSChannelAccessID,
                                spectrumQueryInfo
                                )
```

Parameters

Name	Type	Description
cRSChannelAccessID	TransactionID	cRSChannelAccessID uniquely identifies one transaction of requesting the channel for a CRS.
spectrumQueryInfo	SpectrumQueryInfo	A set of geo-location information, device descriptor and device characteristics of one or multiple CRS(s).

When used

This primitive shall be used by the interface function when it needs to obtain spectrum usage as described by the physical profile, geo-location, characteristics and usage requirements of CRS.

Effect of receipt

The spectrum coordination function receives the channel request from a CRS.

CRS_Coordinated_Channel.response

Function

This primitive is used by the spectrum coordination function to provide the response of the channel request of a CRS to the interface function as a response to the *CRS_Coordinated_Channel.request* primitive.

Semantics of the service primitive

CRS_Coordinated_Channel.response(
cRSChannelAccessID,
status
)

Parameters

Name	Type	Description
cRSChannelAccessID	TransactionID	cRSChannelAccessID uniquely identifies one transaction of requesting the channel for a CRS.
status	Status	Status of channel request.

When used

This primitive shall be used by the spectrum coordination function when it receives the status of channel request.

Effect of receipt

The interface function receives the result of the channel request of a CRS.

4.3.8 Providing available channel list for CRS

CRS_Coordinated_Available_Channel.indication

Function

This primitive is used by the spectrum coordination function to provide available channels to a CRS.

Semantics of the service primitive

CRS_Coordinated_Available_Channel.indication(
cRSAvailableChannelID,
operationalParameters
)

Parameters

Name	Type	Description
cRSAvailableChannelID	TransactionID	cRSAvailableChannelID uniquely identifies one transaction of indicating available channel for a CRS.
operationalParameters	OperationalParameters	Operational parameters describe the available spectrum for a CRS and any additional information to allow the CRS to utilize the spectrum.

When used

This primitive shall be used by the spectrum coordination function when it needs to indicate the available spectrum for a CRS as described by the operational parameters.

Effect of receipt

The interface function receives the indication for available channels of a CRS.

CRS_Coordinated_Channel_Usage.response

Function

This primitive is used by the interface function to provide the channel usage parameters from the spectrum coordination function as a response to the CRS_Coordinated_Available_Channel.indication primitive.

Semantics of the service primitive

CRS_Coordinated_Channel_Usage.response(
cRSAvailableChannelID,
deviceDescriptor,
geolocation,
channelUsageParameters
)

Parameters

Name	Type	Description
cRSAvailableChannelID	TransactionID	cRSAvailableChannelID uniquely identifies one transaction of indicating available channel for a CRS.
deviceDescriptor	DeviceDescriptor	Device descriptor describes the physical profile of a CRS.
geolocation	Geolocation	Geo-location information of CRS.
channelUsageParameters	ChannelUsageParameters	Channel usage parameters describe the operation information of a CRS.

When used

This primitive shall be used by the interface function to generate channel usage parameters as a response to CRS_Coordinated_Available_Channel.indication.

Effect of receipt

The spectrum coordination function receives the channel usage parameters of a CRS.

4.3.9 CRS Reconfiguration

The CRS reconfiguration service provides a set of primitives or method through which the spectrum coordination function provides the operational parameters of a CRS.

CRS_Reconfiguration.request

Function

This primitive is used by the spectrum coordination function to request the reconfiguration of a CRS.

Semantics of the service primitive

CRS_Reconfiguration.request(
cRSReconfigurationID,
operationalParameters
)

Parameters

Name	Type	Description
cRSReconfigurationID	TransactionID	cRSReconfigurationID uniquely identifies one transaction of requesting the reconfiguration for a CRS.
operationalParameters	OperationalParameters	Operational parameters describe the available spectrum of a CRS and any additional information to allow the CRS to use it.

When used

This primitive is used by the spectrum coordination function when it needs to request the reconfiguration of a CRS.

Effect of receipt

The interface function receives the request of the reconfiguration of a CRS.

CRS_Reconfiguration.response

Function

This primitive is used by the interface function to provide the result of the reconfiguration of a CRS to the spectrum coordination function as a response to the *CRS_Reconfiguration.request* primitive.

Semantics of the service primitive

CRS_Reconfiguration.response(
cRSReconfigurationID,
status
)

Parameters

Name	Type	Description
cRSReconfigurationID	TransactionID	cRSReconfigurationID uniquely identifies one transaction of requesting the channel for a CRS.
status	Status	Status of reconfiguration request.

When used

This primitive is used by the interface function when it receives the reconfiguration request.

Effect of receipt

The spectrum coordination function receives the result of the reconfiguration of a CRS.

4.3.10 Providing coordination report for CRS

CRS_Coordination_Report.indication

Function

This primitive is used by the spectrum coordination function to send the coordination report to a CRS.

Semantics of the service primitive

CRS_Coordination_Report.indication(
cRSCoordinationReportID,
coordinationReport
)

Parameters

Name	Type	Description
cRSCoordinationReportID	TransactionID	cRSCoordinationReportID uniquely identifies one transaction of indicating coordination report for a CRS.
coordinationReport	CoordinationReport	Coordination report describes the information needed by the CRS to allow it to make decisions related to its operational parameters which allow coexistence.

When used

This primitive shall be used by the spectrum coordination function when it needs to send the coordination report to a CRS.

Effect of receipt

The interface function receives the coordination report.

CRS_Coordination_Report.response

Function

This primitive is used by the spectrum coordination function to confirm receiving the coordination report to the spectrum coordination function as a response to the CRS_Coordination_Report.indication primitive.

Semantics of the service primitive

CRS_Coordination_Report.response(
cRSCoordinationReportID,
status
)

Parameters

Name	Type	Description
cRSCoordinationReportID	TransactionID	cRSCoordinationReportID uniquely identifies one transaction of indicating coordination report for a CRS.
status	Status	Status of coordination report indication.

When used

This primitive shall be used by the interface function to generate a response to the CRS_Coordination_Report.indication.

Effect of receipt

The spectrum coordination function receives the result of the coordination report indication of a CRS.

4.3.11 CRS operational parameters update

The CRS operational parameters update service provides a set of primitives or method through which the spectrum coordination function provides the request of the operational parameters update of a CRS.

CRS_Operational_Parameters_Update.request

Function

This primitive is used by the spectrum coordination function to request the operational parameters update of a CRS.

Semantics of the service primitive

CRS_Operational_Parameters_Update.request(
cRSOperationalParametersUpdateID,
validTime
)

Parameters

Name	Type	Description
cRSOperational ParametersUpdateID	TransactionID	cRSOperationalParametersUpdateID uniquely identifies one transaction of requesting the operational parameters update for a CRS.
validTime	ValidTime	ValidTime describes the time limit for the update of CRS's operational parameters.

When used

This primitive sahll be used by the spectrum coordination function when it needs to request the operational parameter update of a CRS.

Effect of receipt

The interface function receives the request of the operational parameter update of a CRS.

CRS_Operational_Parameters_Update.response

Function

This primitive is used by the interface function to provide the response of the operational parameters update of a CRS to the spectrum coordination function as a response to the *CRS_Operational_Parameters_Update.request* primitive.

Semantics of the service primitive

```
CRS_Operational_Parameters_Update.response(
                                        cRSOperationalParametersUpdateID,
                                        status
                                        )
```

Parameters

Name	Type	Description
cRSOperationalParamtersUpdateID	TransactionID	cRSOperationalParametersUpdateID uniquely identifies one transaction of requesting the operational parameters update for a CRS.
status	Status	Status of operational parameters update request.

When used

This primitive shall be used by the interface function when it receives the request of operational parameters update.

Effect of receipt

The spectrum coordination function receives the result of the request of the operational parameters update.

4.3.12 Requesting measurement for CRS

The requesting measurement for CRS service provides a set of primitives or method through which the spectrum coordination function provides the sensing and measurement request of a CRS.

CRS_Measurement.request

Function

This primitive is used by the spectrum coordination function to request sensing and measurement of a CRS.

Semantics of the service primitive

```
CRS_Measurement.request(
                        cRSMeasurementID,
                        measurementType,
                        measurementConfiguration,
                        reportConfiguration
                        )
```

Parameters

Name	Type	Description
cRSMeasurementID	TransactionID	cRSMeasurementID uniquely identifies one transaction of requesting sensing and measurement of a CRS.
measurementType	MeasurementType	Measurement type describes frequency of measurement report (single or periodic) as well as the type of measurement that needs to be taken (e.g. PHY layer sensing, QoS, etc.).
measurementConfiguration	MeasurementConfiguration	Measurement configuration provides the information that used by SC to provide measurement configuration to the CRS.
reportConfiguration	ReportConfiguration	Format and configuration of measurement report.

When used

This primitive shall be used by the spectrum coordination function when it requests the CRS to perform measurements of a specific type, and with a specific timing, frequency, or condition.

Effect of receipt

The interface function receives the measurements request for a specific TransactionID and configures/starts measurements or sensing in the CRS associated with the measurementType and measurementConfiguration associated with the request.

CRS_Measurement.response

Function

This primitive is used by the interface function as a confirmation to the *CRS_Measurement.request* primitive.

Semantics of the service primitive

CRS_Measurement.response(
cRSMeasurementID,
status
)

Parameters

Name	Type	Description
cRSMeasurementID	TransactionID	cRSMeasurementID uniquely identifies one transaction of requesting sensing and measurement of a CRS.
status	Status	Status of measurement request.

When used

This primitive shall be used by the interface function to generate a confirmation of the measurement request as *Measurement.request*.

Effect of receipt

The spectrum coordination function receives the result of measurement request of a CRS.

4.3.13 Periodic measurement for CRS

The periodic measurement for CRS service provides a set of primitives or method through which the interface function provides the sensing and measurement report of a CRS based on schedule.

CRS_MeasurementResults.indication

Function

This primitive is used by the interface function to provide the sensing and measurement report of a CRS.

Semantics of the service primitive

CRS_MeasurementResults.indication(
cRSMeasurementResultsID,
geolocation,
measurementReport
)

Parameters

Name	Type	Description
cRSMeasurementResultsID	TransactionID	cRSMeasurementResultsID uniquely identifies one transaction of providing the sensing and measurement results of a CRS.
geolocation	Geolocation	Geo-location information of CRS.
measurementReport	MeasurementReport	Measurement Report describes the results of the sensing and measurement of a CRS.

When used

This primitive shall be used by the interface function when it needs to provide the sensing and measurement report of a CRS.

Effect of receipt

The spectrum coordination function receives the measurement report of a CRS.

CRS_MeasurementResults.response

Function

This primitive is used by the spectrum coordination function as a confirmation to the *CRS_MeasurementResults.indication* primitive.

Semantics of the service primitive

CRS_MeasurementResults.response(
cRSMeasurementResultsID,
measurementAction,
status
)

Parameters

Name	Type	Description
cRSMeasurementResultsID	TransactionID	cRSMeasurementRequestID uniquely identifies one transaction of providing the sensing and measurement results of a CRS.
measurementAction	MeasurementAction	measurementAction indicates whether measurements should continue following the current measurements period or stop measurements.
status	Status	Status of measurement results.

When used

This primitive shall be used by the spectrum coordination function to generate a confirmation of receiving the measurement report as *CRS_MeasurementReport.indication*.

Effect of receipt

The interface function receives the result of receiving the measurement report from SC.

4.3.14 Single measurement for CRS

The single measurement for CRS service provides a set of primitives or method through which the interface function provides the sensing and measurement report of a CRS once.

CRS_MeasurementResults.indication

Function

This primitive is used by the interface function to provide the sensing and measurement report of a CRS.

Semantics of the service primitive

CRS_MeasurementResults.indication(
cRSMeasurementResultsID,
geolocation,
measurementReport
)

Parameters

Name	Type	Description
cRSMeasurementResultsID	TransactionID	cRSMeasurementResultsID uniquely identifies one transaction of providing the sensing and measurement results of a CRS.
geolocation	Geolocation	Geo-location information of CRS.
measurementReport	MeasurementReport	Measurement Report describes the results of the sensing and measurement of a CRS.

When used

This primitive shall be used by the interface function when it needs to provide the sensing and measurement report of a CRS.

Effect of receipt

The spectrum coordination function receives the measurement report of a CRS.

CRS_MeasurementResults.response

Function

This primitive is used by the interface function as a response to the *CRS_MeasurementResults.indication* primitive.

Semantics of the service primitive

CRS_MeasurementResults.response(
cRSMeasurementResultsID,
status
)

Parameters

Name	Type	Description
cRSMeasurementResultsID	TransactionID	cRSMeasurementResultsID uniquely identifies one transaction of providing the sensing and measurement results of a CRS.
status	Status	Status of measurement results.

When used

This primitive shall be used by the spectrum coordination function to generate a confirmation of receiving the measurement report as *CRS_MeasurementReport.indication*.

Effect of receipt

The interface function receives the result of receiving the measurement report from SC.

4.3.15 CRS device parameters reconfiguration

The CRS device parameter reconfiguration service provides a set of primitives or method through which the spectrum coordination function provides the suggested device parameters of a CRS.

CRS_Device_Parameter_Reconfiguration.request

Function

This primitive is used by the spectrum coordination function to request the reconfiguration of device parameters of a CRS.

Semantics of the service primitive

CRS_Device_Parameter_Reconfiguration.request(
deviceParameterReconfigurationID,
suggestedDeviceParameters
)

Parameters

Name	Type	Description
deviceParameterReconfigurationID	TransactionID	deviceParameterReconfigurationID uniquely identifies one transaction of requesting device parameter reconfiguration for a CRS.
suggestedDeviceParameters	DeviceParameters	suggestedDeviceParameters describe the SC-suggested device parameters of a CRS.

When used

This primitive shall be used by the spectrum coordination function when it needs to request the device parameter reconfiguration of a CRS.

Effect of receipt

The interface function receives the device parameters reconfiguration request.

CRS_Device_Parameter_Reconfiguration.response

Function

This primitive is used by the interface function to provide the result of the device parameters reconfiguration of a CRS to the spectrum coordination function as a response to the *CRS_Device_Parameter_Reconfiguration.request* primitive.

Semantics of the service primitive

CRS_Device_Parameter_Reconfiguration.response(
deviceParameterReconfigurationID,
status
)

Parameters

Name	Type	Description
cRSDeviceParameterReconfigurationID	TransactionID	cRSDeviceParameterReconfigurationID uniquely identifies one transaction of requesting the device parameter reconfiguration for a CRS.
status	Status	Status of device parameters reconfiguration.

When used

This primitive shall be used by the interface function when it receives the device parameter reconfiguration request.

Effect of receipt

The spectrum coordination function receives the result of the device parameter reconfiguration of a CRS.

4.4 Communication Service Access Point (Com-SAP)

4.4.1 General

The Com-SAP is used for the exchange of information between CRS and SC. It abstracts services of the communication by providing a set of primitives or method and mapping these primitives to transport protocols. The terminology used for describing primitives at service access points for peer OSI-service-users follows the model in Figure 4.2 [i.5].

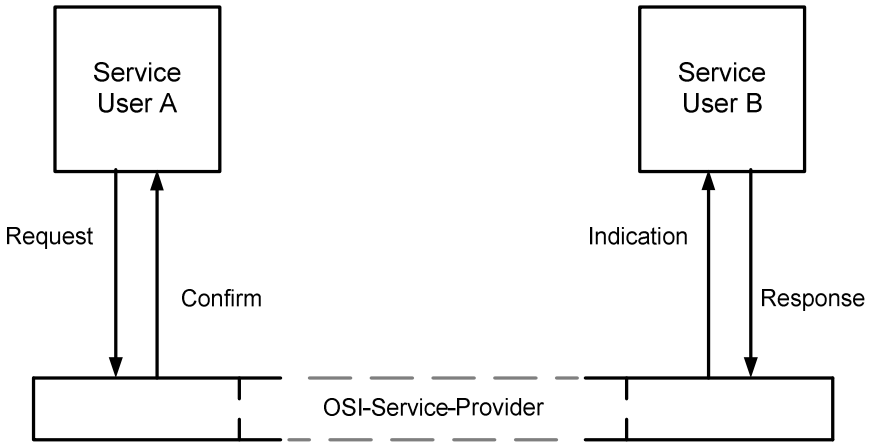

Figure 4.2: Example of a peer-to-peer connection-mode service

4.4.2 Operation-related Information send service

The operation-related information send service provides a set of primitives or method through which the interface functions send information to another interface function utilizing their local Com-SAP.

Operation_Related_Information_Share.request

Function

This primitive is used by the interface function to send operation-related information through its Com-SAP to another interface function that has Com-SAP.

Semantics of the service primitive

```
Operation_Related_Information_Share.request(
                                        operationRelatedInfoShareID,
                                        inforSource,
                                        inforDestination,
                                        route,
                                        secLevel,
                                        operationRelatedInfo
                                        )
```

Parameters

Name	Type	Description
operationRelatedInfoShareID	TransactionID	operationRelatedInfoShareID uniquely identifies one transaction of operation related information sharing request.
inforSource	InforSource	InfoSource gives an ID that uniquely defines one logical entity as the source of operation-related information.
inforDestination	InforDestination	InfoDestination gives an ID that uniquely defines one logical entity as the destination of operation-related information.
route	Route	Route gives a list of IDs of interface functions through which the operation-related information flows from the interface function to the other.
secLevel	SecLevel	Seclevel indicates the level of security depending on the available levels of security for authentication and data verification.
oprationRelatedInfo	OperationRelatedInfo	OperationRelatedInfo describes the operation-related information that is sent through the Com-SAP. Primitives/commands can also be sent as a payload via this Com-SAP.

When used

This primitive shall be used by the interface function when it needs to send operation-related information to another interface function through the Com-SAP.

Effect of receipt

The communication function subsequently uses an *Operation_Related_Information_Share.response* primitive to reflect the results of the request.

Operation_Related_Information_Share.confirm

Function

This primitive returns the results of the request to send operation-related information through the Com-SAP.

Semantics of the service primitive

Operation_Related_Information_Share.confirm(
operationRelatedInfoShareID,
status,
timeout,
lockStatus
)

Parameters

Name	Type	Description
operationRelatedInfoShareID	TransactionID	operationRelatedInfoShareID uniquely identifies one transaction of operation related information sharing request.
status	Status	Status of operation.
timeout	Timeout	Timeout value in second indicates the timeout of the link.
lockStatus	LockStatus	LockStatus indicates whether the entity (i.e. the sender of this primitive) has been locked by an interface function for exclusive service.

When used

The primitive shall be used in response to the *Operation_Related_Information_Share.request* primitive.

Effect of receipt

Upon receipt, the interface function obtains the results of sending operation-related information by the communication function.

Operation_Related_Information_Share.indication

Function

This primitive is used by the interface function to receive operation-related information through its Com-SAP.

Semantics of the service primitive

Operation_Related_Information_Share.indication(
operationRelatedInfoShareID,
inforSource,
operationRelatedInfo
)

Parameters

Name	Type	Description
operationRelatedInfoShareID	TransactionID	operationRelatedInfoShareID uniquely identifies one transaction of operation related information sharing request.
inforSource	InforSource	InfoSource gives an ID that uniquely defines one logical entity as the source of operation-related information.
operationRelatedInfo	OperationRelatedInfo	OperationRelatedInfo describes the operation-related information that is sent through the Com-SAP. Primitives/commands can also be sent as a payload via this Com-SAP.

When used

This primitive shall be used by the interface function when it needs to receive operation-related information from another interface function through the Com-SAP.

Effect of receipt

The communication function subsequently uses an *Operation_Related_Information_Share.request* primitive to reflect the results of the request.

Operation_Related_Information_Share.response

Function

This primitive returns the results of the request to receive operation-related information through the Com-SAP.

Semantics of the service primitive

Operation_Related_Information_Receive.response(
operationRelatedInfoShareID,
status,
)

Parameters

Name	Type	Description
operationRelatedInfoShareID	TransactionID	operationRelatedInfoShareID uniquely identifies one transaction of operation related information sharing request.
status	Status	Status of operation.
timeout	Timeout	Timeout value in second indicates the timeout of the link.
lockStatus	LockStatus	LockStatus indicates whether the entity (i.e. the sender of this primitive) has been locked by an interface function for exclusive service.

When used

The primitive shall be used in response to the *Operation_Related_Information_Share.indication* primitive.

Effect of receipt

Upon receipt, the interface function which sends the request obtains the results of requesting to receive operation-related information through the communication function.

4.4.3 Information services

The information services provide a set of primitives or method through which the interface functions obtain information such as IDs and capabilities of the communication function.

Get_ComSubsys_Profile.request

Function

This primitive is used by the interface function to obtain the information related to the capabilities of a communication function.

Semantics of the service primitive

Get_ComSubsys_Profile.response()

No parameters are used in this primitive.

When used

This primitive shall be used by the interface function when it needs to obtain the information related to the capabilities of the communication function.

Effect of receipt

The communication function subsequently uses a Get_ComSubsys_Profile.response primitive to reflect the results of the request.

Get_ComSubsys_Profile.response

Function

This primitive returns the results of the request to get the communication function profile.

Semantics of the service primitive

```
Get_ComSubsys_Profile.response(
                                status,
                                com_Subsys_ID,
                                com_Subsys_Capability,
                                lockStatus
                                )
```

Parameters

Name	Type	Description
status	Status	Status of operation.
com_Subsys_ID	Com_Subsys_ID	Com_Subsys_ID uniquely defines one compliant communication function.
com_Subsys_Capability	Com_Subsys_Capability	Com_Subsys_Capability describes the capability of the communication function.
lockStatus	LockStatus	LockStatus indicates whether the entity (i.e. the sender of this primitive) has been locked by an interface function for exclusive service.

When used

The primitive shall be used in response to the Get_ComSubsys_Profile.request primitive.

Effect of receipt

Upon receipt, the interface function sending the request obtains the profile of the communication function.

Notify

Function

This primitive is used by the communication function to notify a status change to interface function through the Com-SAP.

Semantics of the service primitive

Notify(
type,
status,
reason
)

Parameters

Name	Type	Description
type	Type	Type parameter specifies the type of the notification.
status	Status	Status of operation.
reason	Reason	Reason parameter expresses the reason of the notification.

When used

This primitive shall be used by the communication function to report a status change to an interface function.

Effect of receipt

The interface function obtains the status of the communication function.

5 Information elements

5.1 General

Clause 5.2 below describes the data types as defined in the SAP primitives described in clause 4. Data types are described using ASN.1 [i.6] but alternative ways are possible and are not excluded.

5.2 Data format

- TransactionID

```
--ASN1START

TransactionID ::= OCTET STRING

--ASN1STOP
```

- DeviceDescriptor

DeviceDescriptor is specified in ETSI EN 301 598 [i.2].

```
--ASN1START

DeviceDescriptor ::= SEQUENCE{

    --Device type
    deviceType          ENUMERATED{
                                typeA,
                                typeB
},
    --Device category
    deviceCategory      ENUMERATED{
                                master,
                                slave
},
    --Unique device ID
    deviceID            ,
    --Technology identifier
    technologyIdentifier    OCTET STRING,
    --Device emission class
    deviceEmissionClass INTEGER(1..5)
}

--ASN1STOP
```

- DeviceID

```
-- ASN1START

DeviceID ::= SEQUENCE{

    --Manufacturer identifier
    manufacturerIdentifier  OCTET STRING,
    --Model identifier
    modelIdentifier     OCTET STRING,
    --Serial number
    serialNumber        OCTET STRING
}

--ASN1STOP
```

- Geolocation

Geolocation includes location information of one CRS.

```
--ASN1START

Geolocation ::= SEQUENCE{
    --Longitude [degree]
    longitude           REAL,
    --Latitude [degree]
    latitude            REAL,
    --Altitude [m]
    altitude            REAL
}

--ASN1STOP
```

- DeviceCapabilities

DeviceCapabilities provides the additional information needed for initialization.

```
--ASN1START

DeviceCapabilities ::= SEQUENCE{

    --The number of antennas at the requesting CRS indicating the ability of mitigating the effect
of interference spatially
    numberOfAntennas        INTEGER,
    --Capability to access GLDB via another CRS. "0" unable to access GLDB via another CRS; "1" able
to access GLDB via another CRS.
    accessRoutingEnabled    BOOLEAN,
    --Routing information for accessing GLDB
    routeCRS            OCTET STRING,
    --Priority access of the CRS
    priotityAccessTrue  BOOLEAN,
    --Expected QoS of priority access
    expectedQoS         QoS,
    ...
}

--ASN1STOP
```

- QoS

```
--ASN1START

QoS ::= CHOICE{
    --Expected bit error rate
    bitErrorRate        REAL,
    --Expected signal to interference ratio [dB]
    sIR                 REAL
}

--ASN1STOP
```

- Rulesetinformation

Rulesetinformation contains parameters for the ruleset of a regulatory domain described in IETF PAWS [i.3].

```
--ASN1START

RulesetInformation ::= SEQUENCE{

    authority           OCTET STRING,
    rulesetId           OCTET STRING,
    maxLocationChange       REAL,
    maxPollingSecs      INTEGER,
    ...
}

--ASN1STOP
```

- SCGLDBInformation

SCGLDBInformation contains list of name and URL of SC and GLDB.

```
--ASN1START

SCGLDBInformation ::= SEQUENCE{

    scgldbSpec  SCGLDBSpec
}

--ASN1STOP
```

- SCGLDBSpec

SCGLDBSpec contains the name and URI of a SC and a GLDB.

```
--ASN1START

SCGLDBSpec ::= SEQUENCE{

    name    OCTET STRING,
    uri OCTET STRING
}

--ASN1STOP
```

- SubscriptionRequest

```
--ASN1START

SubscriptionRequest ::= ENUMERATED{

    -- Management Service
    management,
    -- Information Service
    information,
    -- No Service
    noService
}

--ASN1STOP
```

- Status

```
--ASN1START

Status ::= ENUMERATED{

    --Operation is successful
    success,
    --Operation has encountered unspecified failure
    unspecifiedFailure,
    --Operation has been rejected
    rejection,
    --Operation has failed due to authentication failure
    authenticationFailure,
    --Operation is rejected because the expected operation such as expected QoS cannot be supported
    unableToSupport,
    --Operation limits the CRS to use the estimated required spectrum resources based on its
expected QoS
    requestAccept,
    ...
}

--ASN1STOP
```

- SubscriptionChangeRequest

```
--ASN1START

SubscriptionChangeRequest ::= ENUMERATED{

    -- Management Service
    management,
    -- Information Service
    information
}

--ASN1STOP
```

- LocationInfo

LocationInfo includes a set of location information of multiple CRSs, a region where a CRS is active or a region represented by a rectangular.

```
--ASN1START

LocationInfo ::= CHOICE{
    --Geolocations of one or multiple CRSs
    geolocations        SEQUENCE OF Geolocation,
    --Region where a CRS is active represent by an area defined by geolocations
    region          Region,
    --Region where a CRS is active represented by a rectangular
    rectangularRegion
}

--ASN1STOP
```

- Region

Region includes multiple location information of a CRS where the CRS is active.

```
--ASN1START

--Information of the bounded area defined by the multiple geolocations
minNumGeolocInfo            INTEGER ::= 3

Region ::= SEQUENCE{
    numGeolocInfo       INTEGER,
    geolocation         Geolocation(SIZE(minNumGeolocInfo..numGeolocInfo))
}

--ASN1STOP
```

- RectangularRegion

RectangularRegion is the information represented by a rectangular.

```
--ASN1START

RectangularRegion ::= SEQUENCE{

    --Geolocation of the upper-left point of the rectangular
    geolocationUpper            Geolocation,
    --Geolocation of the lower-right point of the rectangular
    geolocationLower            Geolocation
}

--ASN1STOP
```

- DeviceCharacteristics

DeviceCharacteristics provides CRS's device information.

```
--ASN1START

DeviceCharacteristics ::= SEQUENCE{

    --Antenna information of master device
    masterAntennaInfo                ,
    --The maximum transmission power level of master device [dBm]
    txPower                 REAL        OPTIONAL,
    --Antenna information of slave device
    slaveAntennaInfo                SlaveAntennaInfo        OPTIONAL,
    --Adjacent channel selectivity
    aCS                         ACS     OPTIONAL,
    --Adjacent channel leakage ratio
    aCLR                        ACLR        OPTIONAL,
    --Guaranteed QoS of backhaul connection
    guaranteedQoSOfBackhaulConnection       OPTIONAL,
    ...
}
```

```
--ASN1STOP
```

- MasterAntennaInfo

MasterAntennaInfo includes antenna information of master device.

```
--ASN1START

MasterAntennaInfo   ::= SEQUENCE {

    --The number of antennas at the requesting CRS indicating the ability of mitigating the effect
of interference spatially
    numberOfAntennas        INTEGER,

    --Antenna height above ground level of master device [m]
    masterAntennaHeight REAL,

    --Antenna gain of master device [dBi]
    masterAntennaGain       REAL,

    --The type of antenna array. Present if the number of antenna is two or more.
    antennaType         AntennaType     OPTIONAL,

    -- MIMO type. Present if the number of antenna is two or more.
    mimoType        ENUMERATED {
                        twoDimentional,
                        threeDimentional
}           OPTIONAL,
--Antenna processing capability includes directional beam forming and multiantenna precoding.
Present if the number of antenna is two or more.
    multiAntProCap      MultiAntProCap  OPTIONAL,
    --Antenna boresight azimuth angle direction measured in degree against longitude facing north in
clockwise direction. (i.e. an azimuth angle of zero degrees) is a horizontal line in the direction
to the north pole, starting from the antenna. Present if the number of antenna is two or more.
    azimuthAngle        REAL            OPTIONAL,

    --Start angle of range of boresight azimuth angle
    startRangeAzimuthAngle      REAL            OPTIONAL,

    --Stop angle of range of boresight azimuth angle
    stopRangeAzimuthAngle       REAL            OPTIONAL,

--The elevation angle is the angle between the horizontal plane and the boresight of the planar
array, measured in the vertical plane. The reference direction (i.e. an elevation angle of zero
degrees) is a horizontal line in the direction to the horizon, starting from the antenna. Present if
the number of antenna is two or more.
elevationAngle      REAL            OPTIONAL,

    --Start angle of range of elevation angle
    startRangeElevationAngle        REAL            OPTIONAL,

    --Stop angle of range of elevation angle
    stopRangeElevationAngle         REAL            OPTIONAL,
    ...

}

--ASN1STOP
```

- AntennaType

```
--ASN1START

AntennaType ::= ENUMERATED{

    --Linear array
    linear,
    --Planar array
    planar,
    --Circular
    circular,
    ...
}
```

```
--ASN1STOP
```

- Multiple antenna processing capability

```
--ASN1START

MultiAntProCap ::= ENUMERATED{

    --Directional beam forming capability
    beamforming,
    --Multiple antenna precoding capability
    precoding
}

--ASN1STOP
```

- SlaveAntennaInfo

SlaveAntennaInfo includes antenna information of slave device.

```
--ASN1START

SlaveAntennaInfo    ::= SEQUENCE {

    --The number of antennas at slave device
    numberOfAntennas        INTEGER,

    --Antenna height above ground level of slave device [m]
    slaveAntennaHeight  REAL,

    --Antenna gain of slave device [dBi]
    slaveAntennaGain        REAL

}

--ASN1STOP
```

- ACS

```
--ASN1START

ACS ::= SEQUENCE{
    --ACS on 1st adjacent channel [dB]
    acsRatio1           REAL,
    --ACS on 2nd adjacent channel [dB]
    acsRatio2           REAL,
    --ACS on 3rd adjacent channel [dB]
    acsRatio3           REAL,
    --ACS on 4th adjacent channel [dB]
    acsRatio4           REAL,
    --ACS on 5th adjacent channel [dB]
    acsRatio5           REAL,
    --ACS on 6th adjacent channel [dB]
    acsRatio6           REAL,
    --ACS on 7th adjacent channel [dB]
    acsRatio7           REAL,
    --ACS on 8th adjacent channel [dB]
    acsRatio8           REAL,
    ...
}

--ASN1STOP
```

- ACLR

```
--ASN1START

ACLR ::= SEQUENCE{
    --ACLR on 1st adjacent channel [dB]
    aclrRatio1          REAL,
    --ACLR on 2nd adjacent channel [dB]
    aclrRatio2          REAL        OPTIONAL,
```

```
    ...
}

--ASN1STOP
```

- GuaranteedQoSOfBackhaulConnection

```
--ASN1START

GuaranteedQoSOfBackhaulConnection ::= SEQUENCE{
    --Backhaul type ID
    backhaulTypeID  ENUMEATED{
        xDSL,
        opticalFibre,
        ...
}       OPTIONAL,
    --Guaranteed minimum bit rates of backhaul connection [Mb/s]
    guaranteedMinimumBitRates           REAL        OPTIONAL,
    ...
}

--ASN1STOP
```

- DeviceUsageRequirements

DeviceUsageRequirements provides the information that be used by CRS to provide additional information to the SC and/or GLDB that can help it to determine operational parameters.

```
--ASN1START

DeviceUsageRequirements ::= SEQUENCE{

    --Minimum required SNR [dB]
    minReqSNR               REAL        OPTIONAL,
    ...
}

--ASN1STOP
```

- OperationalParameters

```
--ASN1START

OperationalParameters ::= SEQUENCE{

    --Ruleset Information
    rulesetInformation      RulesetInformation,
    --List of available frequencies
    listOfavailableFrequencies  ListOfAvailableFrequencies,
    --Time validity
    timeValidity            TimeValidity,
    --Location validity [m]
    locationValidity            REAL,
    --GLDB access routing and timing information
    databaseAccessTiming        DatabaseAccessTiming,
    --Routing information for accessing GLDB
    routeCRS                OCTET STRING,   --Interference leakage weighting factor describes the
weight on the interference of a CRS to co-channel CRSs, where the value is limited from 0 to
    intLeakageFactor            REAL,
    --List of reference point locations of the incumbent for each available channels that can be
used to generate null pattern towards incumbents, e.g., reduced directivity gain.
    listOfSpecUsageInfoOfRefPoints  ListOfSpecUsageInfo,
    --List of cochannel neighbor CRSs location information for available channels that can be used
to generate null pattern toward cochannel CRS for better coexistence.
    listOfSpecUsageInfoOfNeightborCRSs  ListOfSpecUsageInfo
}

--ASN1STOP
```

- DatabaseAccessTIming

```
--ASN1START

DatabaseAccessTiming ::= SEQUENCE{

    --Start time of GLDB in UTC
    startTime           UTCTime,
    --Update timer [seconds]
    updateTimer         REAL,
    ...
}

--ASN1STOP
```

- ListOfAvailableFrequencies

```
--ASN1START

ListOfAvailableFrequencies ::= SEQUENCE OF AvailableFrequency

--ASN1STOP
```

- AvailableFrequency

```
--ASN1START

AvailableFrequency ::= SEQUENCE{

    --DTT channel edge frequency pair start edge frequency [MHz]
    startFreq               REAL,
    --DTT channel edge frequency pair stop edge frequency [MHz]
    stopFreq                REAL,
    --Maximum in-block RF EIRP spectrum density [dBm/0.1MHz]
    maximumEIRPDensity      REAL,
    --Maximum in-block RF EIRP [dBm]
    maximumEIRP             REAL,
    --Information of priority of the available frequency. High level means that the available
frequency is more preferable to be used than the others. "0" is the highest level.
    priorityLevel           REAL            OPTIONAL
}

--ASN1STOP
```

- TimeValidity

```
--ASN1START

TimeValidity ::= SEQUENCE{

    --Time validity start in UTC
    startTime           UTCTime,
    --Time validity end in UTC
    stopTime            UTCTime
}

--ASN1STOP
```

- ListOfSpecUsageInfo

```
--ASN1START
    -- List of spectrum usage information of neighbor CRSs or reference points of incumbents in each
available frequency channel.

    ListOfSpecUsageInfo ::= SEQUENCE OF SpecUsageInfo

--ASN1STOP
```

- SpecUsageInfo

```
--ASN1START

-- Spectrum usage information of neighbor CRSs or reference points of incumbents in each frequency
channel.

SpecUsageInfo   ::= SEQUENCE{
    -- System type
    systemType      SourceType,
    -- Start frequency
    startFreq       REAL,
    -- Stop frequency
    stopFreq        REAL,
    -- Geolocation information
    geolocation     SEQUENCE OF Geolocation
}

--ASN1STOP
```

- ChannelUsageParameters

```
--ASN1START

ChannelUsageParameters ::= SEQUENCE{

    --List of usage frequencies
    listOfUsageFrequencies      ListOfUsageFrequencies
}

--ASN1STOP
```

- ListOfUsageFrequencies

```
--ASN1START

ListOfUsageFrequencies  ::= SEQUENCE OF UsageFrequency

--ASN1STOP
```

- UsageFrequency

```
--ASN1START

UsageFrequency ::= SEQUENCE{

    --DTT channel edge frequency pair start edge frequency [MHz]
    startFreq               REAL,
    --DTT channel edge frequency pair stop edge frequency [MHz]
    stopFreq                REAL,
    --Maximum in-block RF EIRP spectrum density [dBm/0.1MHz]
    maximumEIRPDensity      REAL,
    --Maximum in-block RF EIRP [dBm]
    maximumEIRP             REAL
}

--ASN1STOP
```

- CoordinationReport

CoordinationReport describes the information needed by the CRS to allow it to make decisions related to its operational parameters which allow coexistence.

```
--ASN1START

CoordinationReport ::= SEQUENCE OF SEQUENCE{

    networkID               OCTET STRING            OPTIONAL,
    technologyIdentifier        OCTET STRING            OPTIONAL,
    listOfAvailableFrequencies  ListOfAvailableFrequencies  OPTIONAL
}
```

```
--ASN1STOP
```

- ValidTime

```
--ASN1START

ValidTime ::= SEQUENCE{

    --Time validity end in UTC
    stopTime            UTCTime
}

--ASN1STOP
```

- MeasurementType

```
--ASN1START

MeasurementType ::= SEQUENCE{

    --Frequency of measurement report
    reportType          ReportType,
    --Type of measurement method
    measurementMethod   MeasurementMethod
}

--ASN1STOP
```

- ReportType

```
--ASN1START

ReportType ::= ENUMERATED{

    --Single Measurement
    single,
    --Periodic Measurement
    periodic
}

--ASN1STOP
```

- MeasurementMethod

```
--ASN1START

MeasurementMethod ::= ENUMERATED{

    --PHY Layer Sensing
    pHYLayerSensing,
    --QoS Measurement
    qoSMeasurement,
    --Interference Measurement
    interferenceLevel,
    --Throughput Measurement
    throughput,
    ...
}

--ASN1STOP
```

- MeasurementConfiguration

```
--ASN1START

MeasurementConfiguration ::= SEQUENCE {

    --Measurement time
    measurementTime         MeasurementTime,
    --Time between measurements [second]
    timeBetweenMeasurements     REAL            OPTIONAL,
    --Number of measurement
    numberOfMeasurements        INTEGER     OPTIONAL,
    --Measurement frequency
    measurementFrequency        MeasurementFrequency,
    ...
}

--ASN1STOP
```

- MeasurementFrequency

```
--ASN1START

MeasurementFrequency ::= SEQUENCE{

    --Measurement start frequency [MHz]
    startFreq               REAL,
    --Measurement stop frequency [MHz]
    stopFreq                REAL,
    ...
}

--ASN1STOP
```

- MeasurementTime

```
--ASN1START

MeasurementTime ::= SEQUENCE{

    --Measurement start time in UTC
    measStartTime           UTCTime,
    --Measurement stop time in UTC
    measStopTime            UTCTime,
    ...
}

--ASN1STOP
```

- ReportConfiguration

```
--ASN1START

ReportConfiguration ::= CHOICE{

    formatA             OCTET STRING,
    formatB             OCTET STRING,
    formatC             OCTET STRING,
    ...
}

--ASN1STOP
```

- MeasurementReport

```
--ASN1START

MeasurementReport ::= SEQUENCE{

--Measurement method
    measurementMethod           MeasurementMethod           OPTIONAL,
--Measurement configuration
    measurementConfiguration    MeasurementConfiguration    OPTIONAL,
--Measurement data
    measurementData         MeasurementData
}

--ASN1STOP
```

- MeasurementData

```
--ASN1START

MeasurementData ::= SEQUENCE{

    --Measurement Time
    measurementTime         MeasurementTime,
    --Measurement dataset
    measurementDataset      CHOICE{
        --Interference Report
        listOfInterferenceReport    SEQUENCE OF InterferenceReport,
        --Throughput Report
        listOfThroughputReport      SEQUENCE OF ThroughputReport,
        ...
    }
}

--ASN1STOP
```

- InterferenceReport

```
--ASN1START

InterferenceReport ::= SEQUENCE{

    --Interference source type
    sourceType                  SourceType,
    --Measurement Frequency
    measurementFrequency        MeasurementFrequency,
    --Interference Level [dBm]
    interferenceLevel               REAL,
    ...
}

--ASN1STOP
```

- Source Type

```
--ASN1START

SourceType ::= ENUMERATED{

    --Incumbent
    incumbent,
    --CRS
    cRS,
    --Unknown
    unknown,
    ...
}

--ASN1STOP
```

- Throughput Report

```
--ASN1START

ThroughputReport ::= SEQUENCE{

    --Measurement Frequency
    measurementFrequency           MeasurementFrequency,
    --Throughput value [Mbps]
    throughputValue            REAL,
    ...
}

--ASN1STOP
```

- MeasurementAction

```
--ASN1START

MeasurementAction ::= ENUMERATED{

    --Measurement Stop
    measurementStop,
    --Measurement Again
    measurementAgain,
    ...
}

--ASN1STOP
```

- DeviceParameters

```
--ASN1START

DeviceParameters ::= SEQUENCE{

    --Antenna information
    masterAntennaInfo              MasterAntennaInfo,
    --The maximum transmission power level of master device [dBm]
    txPower             REAL            OPTIONAL,
    --Geolocation of master device
    geolocation             Geolocation     OPTIONAL,
    ...
}

--ASN1STOP
```

- SpectrumQueryInfo

```
--ASN1START

SpectrumQueryInfo ::= SEQUENCE{

    cRSInfo     CRSInfo
}

--ASN1STOP
```

- CRSInfo

```
--ASN1START

CRSInfo ::= SEQUENCE{

    deviceDescriptor           DeviceDescriptor,
    locationInfo           LocationInfo,
    deviceCharacteristics       DeviceCharacteristics
}

--ASN1STOP
```

- InforSource

```
--ASN1START

InforSource ::= SEQUENCE{

    --ID of source entity
    sourceID            OCTET STRING
}

--ASN1STOP
```

- InforDestination

```
--ASN1START

InforDestination ::= SEQUENCE{

    --ID of destination entity
    destinationID           OCTET STRING
}

--ASN1STOP
```

- Route

```
-- ASN1START

Route ::= SEQUENCE{

    --Route information for transporting operation related information
    route               OCTET STRING,
    ...
}

--ASN1STOP
```

- SecLevel

```
--ASN1START

SecLevel ::= INTEGER

--ASN1STOP
```

- OperationRelatedInfo

```
--ASN1START

OperationRelatedInfo ::= CHOICE{

    initializationRequest           InitializationRequest,
    initializationResponse          InitializationResponse,
    serviceSubscriptionRequest      ServiceSubscriptionRequest,
    serviceSubscriptionResponse     ServiceSubscriptionResponse,
    serviceSubscriptionUpdateRequest    ServiceSubscriptionUpdateRequest,
    serviceSubscriptionUpdateResponse   ServiceSubscriptionUpdateResponse,
    serviceSubscriptionChangeRequest    ServiceSubscriptionChangeRequest,
    serviceSubscriptionChangeResponse   ServiceSubscriptionChangeResponse,
    networkRegistrationRequest      NetworkRegistrationRequest,
    networkRegistrationResponse     NetworkRegistrationResponse,
    networkRegistrationUpdateRequest    NetworkRegistrationUpdateRequest,
    networkRegistrationUpdateResponse   NetworkRegistrationUpdateResponse,
    coordinatedChannelRequest       CoordinatedChannelRequest,
    coordinatedChannelResponse      CoordinatedChannelResponse,
    coordinatedAvailableChannelIndication   CoordinatedAvailableChannelIndication,
    coordinatedChannelUsageResponse CoordinatedChannelUsageResponse,
    reconfigurationRequest          ReconfigurationRequest,
    reconfigurationResponse         ReconfigurationResponse,
    coordinationReportIndication        CoordinationReportIndication,
    coordinationReportResponse      CoordinationReportResponse,
```

```
    operationalParametersUpdateRequest  OperationalParametersUpdateRequest,
    operationalParametersUpdateResponse OperationalParametersUpdateResponse,
    measurementRequest          MeasurementRequest,
    measurementResponse         MeasurementResponse,
    measurementResultsIndication        MeasurementResultsIndication,
    measurementResultsResponse      MeasurementResultsResponse,
    deviceParameterReconfigurationRequest   DeviceParameterReconfigurationRequest,
    deviceParameterReconfigurationResponse  DeviceParameterReconfigurationResponse,
    ...
}

--ASN1STOP
```

- InitializationRequest

```
--ASN1START

InitializationRequest ::= SEQUENCE{

    deviceDescriptor        DeviceDescriptor,
    geolocation         Geolocation,
    deviceCapabilities  DeviceCapabilities
}

--ASN1STOP
```

- InitializationResponse

```
--ASN1START

InitializationResponse ::= SEQUENCE{

    rulesetInformation  RulesetInformation,
    scgldbInformation       SCGLDBInformation
}

--ASN1STOP
```

- ServiceSubscriptionRequest

```
--ASN1START

ServiceSubscriptionRequest ::= SEQUENCE{

    subscriptionRequest SubscriptionRequest
}

--ASN1STOP
```

- ServiceSubscriptionResponse

```
--ASN1START

ServiceSubscriptionResponse ::= SEQUENCE{

    status          Status
}

--ASN1STOP
```

- ServiceSubscriptionUpdateRequest

```
--ASN1START

ServiceSubscriptionUpdateRequest ::= SEQUENCE{

    subscriptionRequest     SubscriptionRequest
}

--ASN1STOP
```

- ServiceSubscriptionUpdateResponse

```
--ASN1START

ServiceSubscriptionUpdateResponse ::= SEQUENCE{

    status          Status
}

--ASN1STOP
```

- ServiceSubscriptionChangeRequest

```
--ASN1START

ServiceSubscriptionChangeRequest ::= SEQUENCE{

    subscriptionChangeRequest   SubscriptionChangeRequest
}

--ASN1STOP
```

- ServiceSubscriptionChangeResponse

```
--ASN1START

ServiceSubscriptionChangeResponse ::= SEQUENCE{

    status          Status
}

--ASN1STOP
```

- NetworkRegistrationRequest

```
--ASN1START

NetworkRegistrationRequest ::= SEQUENCE{

    deviceDescriptor        DeviceDescriptor,
    geolocation         Geolocation,
    deviceCharacteristics   DeviceCharacteristics
}

--ASN1STOP
```

- NetworkRegistrationResponse

```
--ASN1START

NetworkRegistrationResponse ::= SEQUENCE{

    rulesetInformation  RulesetInformation,
    scgldbInformation       SCGLDBInformation
}

--ASN1STOP
```

- NetworkRegistrationUpdateRequest

```
--ASN1START

NetworkRegistrationUpdateRequest ::= SEQUENCE{

    deviceDescriptor        DeviceDescriptor,
    geolocation         Geolocation,
    deviceCharacteristics   DeviceCharacteristics
}
```

```
--ASN1STOP
```

- NetworkRegistrationUpdateResponse

```
--ASN1START

NetworkRegistrationUpdateResponse ::= SEQUENCE{

    rulesetInformation  RulesetInformation,
    scgldbInformation       SCGLDBInformation
}

--ASN1STOP
```

- CoordinatedChannelRequest

```
--ASN1START

CoordinatedChannelRequest ::= SEQUENCE{

    deviceDescriptor        DeviceDescriptor,
    locationInfo        LocationInfo,
    deviceCharacteristics   DeviceCharacteristics,
    deviceUsageRequirements DeviceUsageRequirements
}

--ASN1STOP
```

- CoordinatedChannelResponse

```
--ASN1START

CoordinatedChannelResponse ::= SEQUENCE{

    status          Status
}

--ASN1STOP
```

- CoordinatedAvailableChannelIndication

```
--ASN1START

CoordinatedAvailableChannelIndication ::= SEQUENCE{

    operationalParameters   OperationalParameters
}

--ASN1STOP
```

- CoordinatedChannelUsageResponse

```
--ASN1START

CoordinatedChannelUsageResponse ::= SEQUENCE{

    deviceDescriptor        DeviceDescriptor,
    geolocation         Geolocation,
    channelUsageParameters  ChannelUsageParameters
}

--ASN1STOP
```

- ReconfigurationRequest

```
--ASN1START

ReconfigurationRequest ::= SEQUENCE{

    operationalParameters   OperationalParameters
}

--ASN1STOP
```

- ReconfigurationResponse

```
--ASN1START

ReconfigurationResponse ::= SEQUENCE{

    status          Status
}

--ASN1STOP
```

- CoordinationReportIndication

```
--ASN1START

CoordinationReportIndication ::= SEQUENCE{

    coordinationReport  CoordinationReport
}

--ASN1STOP
```

- CoordinationReportResponse

```
--ASN1START

CoordinationReportResponse ::= SEQUENCE{

    status          Status
}

--ASN1STOP
```

- OperationalParametersUpdateRequest

```
--ASN1START

OperationalParametersUpdateRequest ::= SEQUENCE{

    validTime           ValidTime

}

--ASN1STOP
```

- OperationalParametersUpdateResponse

```
--ASN1START

OperationalParametersUpdateResponse ::= SEQUENCE{

    status          Status
}

--ASN1STOP
```

- MeasurementRequest

```
--ASN1START

MeasurementRequest ::= SEQUENCE{

    measurementType MeasurementType,
    measurementConfiguration    MeasurementConfiguration,
    reportConfiguration ReportConfiguration
}

--ASN1STOP
```

- MeasurementResponse

```
--ASN1START

MeasurementResponse ::= SEQUENCE{

    status          Status
}

--ASN1STOP
```

- MeasurementResultsIndication

```
--ASN1START

MeasurementResultsIndication ::= SEQUENCE{

    geolocation         Geolocation,
    measurementReport       MeasurementReport
}

--ASN1STOP
```

- MeasurementResultsResponse

```
--ASN1START

MeasurementResultsResponse ::= SEQUENCE{

    measurementAction       MeasurementAction,
    status          Status
}

--ASN1STOP
```

- DeviceParameterReconfigurationRequest

```
--ASN1START

DeviceParameterReconfigurationRequest ::= SEQUENCE{

    suggestedDeviceParameters   DeviceParameters
}

--ASN1STOP
```

- DeviceParameterReconfigurationResponse

```
--ASN1START

DeviceParameterReconfigurationResponse ::= SEQUENCE{

    status          Status
}

--ASN1STOP
```

- Timeout

```
--ASN1START


-- Time out [second] of the link
Timeout ::= REAL

--ASN1STOP
```

- LockStatus

```
--ASN1START

LockStatus ::= ENUMERATED{

    --The status of the entity is locked. The entity is locked by client for exclusive service.
    exclusiveUse,
    -- The status of the entity is unlocked. The entity has been unlocked by
its client, or has been unlocked by a self-generated timeout event.
    unlocked
}

--ASN1STOP
```

- Com_Subsys_ID

```
--ASN1START

Com_Subsys_ID ::= OCTET STRING

--ASN1STOP
```

- Com_Subsys_Capability

```
--ASN1START

Com_Subsys_Capability ::= ENUMERATED{

    --Support IEEE 802.11 protocol
    ieee802dot11,
    --Support LTE system
    lte,
    ...
}

--ASN1STOP
```

- Type

```
--ASN1START

Type ::= ENUMERATED{

    periodic,
    ...
}

--ASN1STOP
```

- Reason

```
--ASN1START

Reason ::= ENUMERATED{

    --Operation is successful
    success,
    --Operation has encountered unspecified failure
    unspecifiedFailure,
    --Operation has been rejected
    rejection,
    --Operation has failed due to authentication failure
    authenticationFailure,
    --Operation is rejected because the expected operation cannot be supported
    unableToSupport
}

--ASN1STOP
```

History

Document history			
V1.0.1	July 2015	EN Approval Procedure	AP 20151117: 2015-07-20 to 2015-11-17
V1.1.1	November 2015	Publication	