IEEE Standard for Spectrum Sensing Interfaces and Data Structures for Dynamic Spectrum Access and Other Advanced Radio Communication Systems

Sponsored by the
IEEE Communications Society
and
IEEE Standards Coordinating Committee 41 on Dynamic Spectrum Networks

IEEE
3 Park Avenue
New York, NY 10016-5997
USA

22 April 2011

IEEE Std 1900.6™-2011

IEEE Standard for Spectrum Sensing Interfaces and Data Structures for Dynamic Spectrum Access and Other Advanced Radio Communication Systems

Sponsor

IEEE Communications Society

and

IEEE Standards Coordinating Committee 41 on Dynamic Spectrum Networks

Approved 2 February 2011

IEEE-SA Standards Board

Approved 4 August 2011

American National Standards Institute

Abstract: The interfaces and data structures required to exchange sensing-related information in order to increase interoperability between sensors and their clients developed by different manufacturers are defined in this standard. The logical interface and supporting data structures are defined abstractly without constraining the sensing technology, client design, or data link between sensor and client. The entities involved and parameters exchanged in this process. It further elaborates on the service access points, service primitives, as well as generic procedures used to realize this information exchange, are defined by this standard.

Keywords: cognitive radio, data structure, distributed spectrum sensing, dynamic spectrum access, IEEE 1900.6, logical interface, sensing information, spectrum sensor

---

The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA

PDF: ISBN 978-0-7381-6555-4 STD97079
Print: ISBN 978-0-7381-6556-1 STDPD97079

Introduction

This introduction is not part of IEEE Std 1900.6-2011, IEEE Standard for Spectrum Sensing Interfaces and Data Structures for Dynamic Spectrum Access and Other Advanced Radio Communication Systems.

Given the increasing proliferation of devices that use radio spectrum and the resulting shortage of capacity in allocated spectrum bands, new technologies are being introduced that allow devices to access other unused spectrum bands dynamically to serve traffic demands. These new technologies require reliable, dependable, and trusted spectrum sensing capabilities in order to make accurate assessments of spectrum availabilities in the surrounding operational area. Such capabilities will assist devices and associated radio equipment in identifying locally/temporally available spectrum that can be accessed without affecting the incumbent users of that spectrum.

The IEEE 1900 Standards Committee was established in the first quarter 2005 jointly by the IEEE Communications Society and the IEEE Electromagnetic Compatibility Society. The objective of this effort is to develop supporting standards dealing with new technologies and techniques being developed for next generation radio and advanced spectrum management. On March 22, 2007, the IEEE Standards Board approved the reorganization of the IEEE 1900 effort as Standards Coordinating Committee 41 (SCC41) on Dynamic Spectrum Access Networks (DySPAN). The IEEE Communications Society and Electromagnetic Compatibility Society are sponsoring societies for this effort, as they were for the IEEE 1900 effort.

The IEEE 1900.6 working group was launched on September 26, 2008 to address the development of spectrum sensing interfaces and data structures for the exchange of sensing-related information to increase interoperability between sensors and their clients provided by different manufacturers.

Notice to users

Laws and regulations

Users of these documents should consult all applicable laws and regulations. Compliance with the provisions of this standard does not imply compliance to any applicable regulatory requirements. Implementers of the standard are responsible for observing or referring to the applicable regulatory requirements. IEEE does not, by the publication of its standards, intend to urge action that is not in compliance with applicable laws, and these documents may not be construed as doing so.

Copyrights

This document is copyrighted by the IEEE. It is made available for a wide variety of both public and private uses. These include both use, by reference, in laws and regulations, and use in private self-regulation, standardization, and the promotion of engineering practices and methods. By making this document available for use and adoption by public authorities and private users, the IEEE does not waive any rights in copyright to this document.

Updating of IEEE documents

Users of IEEE standards should be aware that these documents may be superseded at any time by the issuance of new editions or may be amended from time to time through the issuance of amendments, corrigenda, or errata. An official IEEE document at any point in time consists of the current edition of the document together with any amendments, corrigenda, or errata then in effect. In order to determine whether a given document is the current edition and whether it has been amended through the issuance of amendments, corrigenda, or errata, visit the IEEE Standards Association web site at http://ieeexplore.ieee.org/xpl/standards.jsp, or contact the IEEE at the address listed previously.

For more information about the IEEE Standards Association or the IEEE standards development process, visit the IEEE-SA web site at http://standards.ieee.org.

Errata

Errata, if any, for this and all other standards can be accessed at the following URL: http://standards.ieee.org/reading/ieee/updates/errata/index.html. Users are encouraged to check this URL for errata periodically.

Interpretations

Current interpretations can be accessed at the following URL: http://standards.ieee.org/reading/ieee/interp/index.html.

Patents

Attention is called to the possibility that implementation of standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken with respect to the existence or validity of any patent rights in connection therewith. The IEEE is not responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patents Claims or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance, if any, or in any licensing agreements are reasonable or non-discriminatory. Users of this standard are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. Further information may be obtained from the IEEE Standards Association.

Participants

At the time this standard was submitted to the IEEE-SA Standards Board for approval, the IEEE 1900.6 Working Group had the following membership:

Klaus Moessner, *Chair*
Ha Nguyen Tran, *Secretary*
Chen Sun, Bernd Bochow, and Darcy Swain, *Technical Editors*

Yohannes D. Alemseged
Masayuki Ariyoshi
Stefan Aust
Hiroshi Harada
Oliver Holland
Tetsuya Ito
Paulo Marques
Maurizio Murroni
Dominique Noguet
Sofie Pollin
Venkatesha Prasad
Naotaka Sato
Ryo Sawai
Akira Yamaguchi

The following members of the individual balloting committee voted on this standard. Balloters may have voted for approval, disapproval, or abstention.

Stefan Aust
Tuncer Baykas
H. Stephen Berger
Harry Bims
Bernd Bochow
Keith Chow
Yohannes Demessie
Thomas Dineen
Carlo Donati
Marc Emmelmann
Stanislav Filin
Randall Groves
Hiroshi Harada
Marco Hernandez
James Hoffmeyer
Oliver Holland
Kentaro Ishizu
Toru Kabe
Masahiko Kaneko
Piotr Karocki
Stuart J. Kerry
Thomas Kurihara
Joseph Kwak
Daniel Levesque
Arthur Light
Daniel Lubar
Greg Luri
Edward McCall
Klaus Moessner
Michael S. Newman
Przemyslaw Pawelczak
Ulrich Pohl
Yuriy Posherstnik
Venkatesha Prasad
Leonard Pucker
Mohammad Rahman
Maximilian Riegel
Robert Robinson
Kazuyuki Sakoda
Osman Sakr
Naotaka Sato
Ryo Sawai
Bartien Sayogo
Gil Shultz
Chunyi Song
Thomas Starai
Dennis Stewart
Walter Struppler
Chin-Sean Sum
Chen Sun
Jun Ichi Takada
Ha Nguyen Tran
Junyi Wang
Hung-Yu Wei
Akira Yamaguchi
Oren Yuen

When the IEEE-SA Standards Board approved this standard on 2 February 2011, it had the following membership:

Robert M. Grow, *Chair*
Richard H. Hulett, *Vice Chair*
Steve M. Mills, *Past Chair*
Judith Gorman, *Secretary*

Karen Bartleson
Victor Berman
Ted Burse
Clint Chaplin
Andy Drozd
Alexander Gelman
Jim Hughes
Young Kyun Kim
Joseph L. Koepfinger*
John Kulick
David J. Law
Hung Ling
Oleg Logvinov
Ted Olsen
Ronald C. Petersen
Thomas Prevost
Jon Walter Rosdahl
Sam Sciacca
Mike Seavey
Curtis Siller
Don Wright

*Member Emeritus

Also included are the following nonvoting IEEE-SA Standards Board liaisons:

Satish Aggarwal, *NRC Representative*
Richard DeBlasio, *DOE Representative*
Michael Janezic, *NIST Representative*

Don Messina
IEEE Standards Program Manager, Document Development

Matthew J. Ceglia
IEEE Standards Program Manager, Technical Program Development

Contents

IEEE Standard for Spectrum Sensing Interfaces and Data Structures for Dynamic Spectrum Access and Other Advanced Radio Communication Systems

IMPORTANT NOTICE: This standard is not intended to ensure safety, security, health, or environmental protection. Implementers of the standard are responsible for determining appropriate safety, security, environmental, and health practices or regulatory requirements.

This IEEE document is made available for use subject to important notices and legal disclaimers. These notices and disclaimers appear in all publications containing this document and may be found under the heading "Important Notice" or "Important Notices and Disclaimers Concerning IEEE Documents." They can also be obtained on request from IEEE or viewed at http://standards.ieee.org/IPR/disclaimers.html.

1. Overview

Background

Given the increasing proliferation of devices that use radio spectrum and the resulting shortage of capacity in allocated spectrum bands, new technologies are being introduced that allow devices to access unused spectrum bands dynamically to serve traffic demands. These new technologies require reliable, dependable, and trusted spectrum sensing capabilities in order to make accurate assessments of spectrum availability in the surrounding operational area. Such capabilities will assist devices and associated radio equipment in identifying locally/temporally available spectrum that can be accessed without causing harmful interference to the incumbent users of that spectrum.

Problem

Recently proposed advanced radio systems based on sensing technology (e.g., those being worked on within IEEE P802.22™ [B2][1]) combine sensing and the protocols and cognitive engines (CEs) that use the

[1] The numbers in brackets correspond to those of the bibliography in Annex F.

sensing results into proprietary architectures. This model of development reduces innovation and limits the opportunities for integrating new component technologies for better system performance. Furthermore, the results of sensing extend beyond the activities of a single system and are ideally integrated into the larger spectrum management process including the development of spectrum use monitoring and enforcement activities.

Many different sensing techniques have been defined and implemented, yet there has been no effort to provide interoperability between sensors and clients developed by different manufacturers.

Solution

This standard defines the interfaces and data structures required to exchange sensing-related information for increasing interoperability between sensors and their clients developed by different manufacturers. The clients can be cognitive engines as in the focus of this standard or can be any other type of algorithms or devices (e.g., adaptive radio) that use sensing-related information. Being aware of evolving technologies, interfaces are developed to accommodate future extensions, new service primitives, and parameters.

How this standard applies

This standard provides a formal definition of data structures and interfaces for exchange of spectrum sensing-related information.

1.1 Scope

This standard defines the information exchange between spectrum sensors and their clients in radio communication systems. The logical interface and supporting data structures used for information exchange are defined abstractly without constraining the sensing technology, client design, or data link between the sensor and client.

1.2 Purpose

The purpose of this standard is to define spectrum sensing interfaces and data structures for dynamic spectrum access (DSA) and other advanced radio communications systems that will facilitate interoperability between independently developed devices and thus allow for separate evolution of spectrum sensors and other system functions.

1.3 Interfaces and sample application areas

1.3.1 IEEE 1900.6 interfaces

Figure 1 illustrates IEEE 1900.6 interfaces between spectrum sensors and their clients (cf. 3.1).

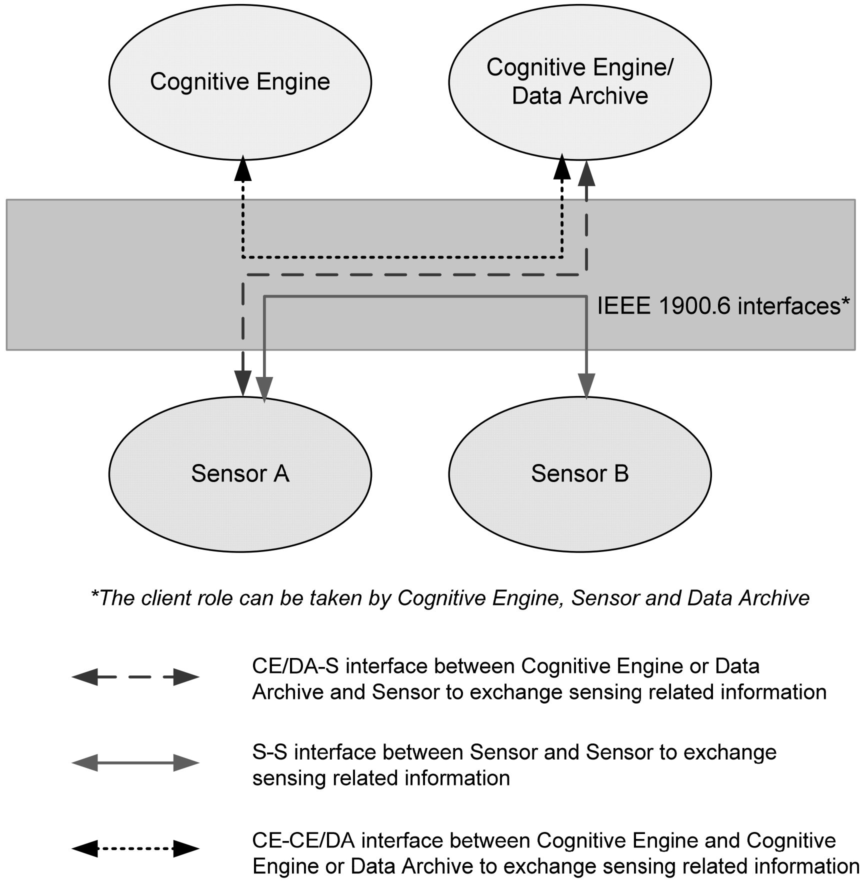


Figure 1—IEEE 1900.6 interfaces between spectrum sensors and their clients

As shown in Figure 1, there are three possible instances of the logical interface within the scope of this standard depending on the logical entities using the IEEE 1900.6 interface:

- CE/DA–S[2] interface between CE or data archive (DA) and Sensor
- S–S interface between Sensor and Sensor
- CE–CE/DA interface between CE and CE or DA

The CE/DA–S interface is used for exchanging sensing-related information between a CE or DA and a Sensor. As an example, the CE/DA–S interface is used in scenarios where a given CE or DA obtains sensing-related information from one or several Sensors or a given Sensor provides sensing-related information to one or several CEs or DAs.

The S–S interface is used for exchanging sensing-related information between Sensors. As an example, the S–S interface is used in scenarios when multiple Sensors exchange sensing-related information for distributed sensing (see the definition of distributed sensing in 3.1).

The CE–CE/DA interface is used for exchanging sensing-related information between CEs or between CE and DA. The CE–CE/DA interface is used in scenarios where CEs exchange sensing-related information for distributed sensing. The CE–CE/DA interface is also used in scenarios where a CE obtains sensing-

[2] “CE/DA” denotes CE or DA.

related information and or policy/regulatory information from a DA. For CE–CE/DA communication, a CE/DA shall be able to take the role of Sensors in terms of providing the CE/DA–S interface for the duration and purpose of exchanging sensing-related information.

The following subclauses detail sample application areas of this standard.

1.3.2 Reporting data format of the sensing-related information

The standard defines the set of sensing parameters that will allow for spectral opportunities exploitation. For example, binary values "0" for idle and "1" for occupied band or soft parameters (e.g., detected received power, cyclo-stationary features level, and direction of arrival of the licensed user signal) shall be reported by the Sensors to their clients. The standard specifies the data formats of these parameters in a protocol-independent manner.

1.3.3 Single-device sensing and multiple-device sensing

Single device sensing denotes spectrum sensing capabilities or performance measurements based on the information obtained from a single sensor. This can be considered as the reference case for more sophisticated sensing techniques based on multiple sensing devices. Multiple device sensing includes distributed sensing between different radio nodes or between sensors. Sensing information obtained from distributed sensors shall be qualified either implicitly or explicitly by time and location of the acquisition of measurement data as well as by the attributes or characteristics of the acquisition method utilized. Some concise examples are given in A.2. The standard has been developed considering different use cases which rely on the exchange of sensing-related information between multiple sensors and a client by referring to existing sensing methods, techniques, and their approaches (see Noguet et al. [B7]). The detailed study of protocols related to multiple device sensing is outside the scope of this standard.

1.3.4 Usage of spectrum sensing cognitive radio (CR) for the investigation of policy violations

The standard defines a logical interface to support exchange of a complete set of sensing-related information between Sensors and their clients. The logical interface defined in this standard may be applied to use cases such as the radio scene analysis and investigation of policy violations. Verification algorithms are outside the scope of this standard.

1.4 Conformance keywords

In this document, the word *shall* is used to indicate a mandatory requirement. The word *should* is used to indicate a recommendation. The word *may* is used to indicate a permissible action. The word *can* is used for statements of possibility and capability.

2. Normative references

The following referenced documents and URLs are indispensable for the application of this document (i.e., they must be understood and used, so each referenced document is cited in text and its relationship to this document is explained). For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments or corrigenda) applies.

IEEE Std 1900.1™-2008, IEEE Standard Definitions and Concepts for Dynamic Spectrum Access: Terminology Relating to Emerging Wireless Networks, System Functionality, and Spectrum Management.[3,4]

IEEE Std 1900.4™-2009, IEEE Standard for Architectural Building Blocks Enabling Network-Device Distributed Decision Making for Optimized Radio Resource Usage in Heterogeneous Wireless Access Networks.

3. Definitions, acronyms, and abbreviations

3.1 Definitions

For the purposes of this document, the following terms and definitions apply. *The IEEE Standards Dictionary: Glossary of Terms & Definitions* should be consulted for terms not defined in this clause.[5] For any definition not given in this subclause, IEEE Std 1900.1™-2008 shall apply.

cognitive engine (CE): A logical entity that contains cognitive control mechanisms, which may include policy-based control mechanisms, and uses sensing information to assess spectrum availability.

data archive (DA): A logical entity storing systematically sensing-related information.

distributed sensing: The process of sensing where sensors are distributed in acquisition time, space, frequency, and function.

IEEE 1900.6 application service access point (A-SAP): A service access point used by applications accessing services of an IEEE 1900.6 compliant server.

IEEE 1900.6 client[6]: A logical entity, application, or device (compliant to this standard) that receives sensing information and spectrum usage-related information from an IEEE 1900.6 compliant server. In general, the information exchange between IEEE 1900.6 compliant clients and servers applies to sensing-related information.

NOTE—The term “client” is used in the context of describing the IEEE 1900.6 service and denotes a conceptual functional role of an IEEE 1900.6 logical entity. It is the counterpart to an IEEE 1900.6 server.[7]

IEEE 1900.6 communication service access point (C-SAP): Used for the exchange of sensing-related information between remote IEEE 1900.6 servers and clients by means of a communication subsystem.

---

[3] The IEEE standards or products referred to in this clause are trademarks owned by the Institute of Electrical and Electronics Engineers, Incorporated.
[4] IEEE publications are available from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, Piscataway, NJ 08854, USA (http://standards.ieee.org/).
[5] *The IEEE Standards Dictionary: Glossary of Terms & Definitions* is available at http://shop.ieee.org/.
[6] The terms "client" and "servers" refer to roles of IEEE 1900.6 logical entities that are defined by the direction of the flow of sensing-related information. These terms are used when addressing the exchange of chunks of information between logical entities and are not related to the use of service primitives and subsequent exchange of protocol messages required to realize the exchange of sensing-related information.
[7] Notes in text, tables, and figures of a standard are given for information only and do not contain requirements needed to implement this standard.

IEEE 1900.6 logical entity: Part of the underlying logical entity–relationship model. It is defined by its functional role(s) and interfaces with other IEEE 1900.6 logical entities. The three types of logical entities in the IEEE 1900.6 logical model are Sensor, CE, and DA. Logical entities are realized through an IEEE 1900.6 service.

IEEE 1900.6 logical interface: A conceptual boundary between two or more IEEE 1900.6 logical entities. Logical interfaces exist between Sensors (S–S interface), between CE and/or DA and Sensors (CE/DA–S interface) and between CE and DA (CE–CE/DA interface). These interfaces are realized by the services or by a subset of the services available at the IEEE 1900.6 SAPs.

IEEE 1900.6 measurement service access point (M-SAP): Used for the exchange of sensing-related information between the spectrum measurement module and its M-SAP user.

IEEE 1900.6 server[6]: An IEEE 1900.6 logical entity, application, or device that provides sensing information and spectrum usage-related information to IEEE 1900.6 clients. In general, the information exchange between IEEE 1900.6 clients and IEEE 1900.6 compliant servers applies to sensing-related information.

NOTE—The term "IEEE 1900.6 server" is used in the context of describing the IEEE 1900.6 service and denotes a conceptual functional role of an IEEE 1900.6 logical entity. It is the counterpart to an IEEE 1900.6 client.

IEEE 1900.6 service: An abstraction of the totality of those functional blocks inside IEEE 1900.6 servers and clients realizing the IEEE 1900.6 logical interface.

IEEE 1900.6 service access point (SAP): A conceptual location at which an IEEE 1900.6 service user can interact with the IEEE 1900.6 service provider by means of utilizing IEEE 1900.6 service primitives. A service access point makes an IEEE 1900.6 service accessible to an IEEE 1900.6 client. The three types of service access points are IEEE 1900.6 communication service access point (C-SAP), IEEE 1900.6 application service access point (A-SAP), and IEEE 1900.6 measurement service access point (M-SAP).

IEEE 1900.6 service primitive: Describes the interaction with an IEEE 1900.6 service provider through an IEEE 1900.6 service access point (SAP) in an abstract implementation-independent way.

NOTE—The term "IEEE 1900.6 service primitive" is adopted from ITU-T Recommendation X.210 [B5].

IEEE 1900.6 service provider: An abstraction of the totality of those entities that provide an IEEE 1900.6 service to the IEEE 1900.6 service user. The IEEE 1900.6 service provider includes the IEEE 1900.6 service and the associated IEEE 1900.6 service access points (SAPs) that instantiate the logical interface for the exchange of sensing-related information between IEEE 1900.6 servers and clients.

NOTE—The term "service provider" is used in the context of protocol and procedure specifications and denotes a conceptual functional role of an IEEE 1900.6 logical entity. It is the counterpart to a service user.

IEEE 1900.6 service user: The IEEE 1900.6 logical entity that shall use the IEEE 1900.6 service and the associated IEEE 1900.6 SAPs that instantiate the logical interface with the purpose to exchange and use sensing-related information.

NOTE—The term "service user" is used in the context of protocol and procedure specifications and denotes a conceptual functional role of an IEEE 1900.6 logical entity. It is the counterpart to a service provider.

message transport service: A communication service commonly associated with the functionality of an open systems interconnection (OSI) transport layer protocol that supports transaction based exchange of messages between communication peers. In this standard, the message transport service implements the exchange of sensing-related information between remote IEEE 1900.6 servers and clients across a platform-provided communication subsystem.

regulatory requirements: A category of sensing-related information derived from regulations, including definitions of the sensing range, accuracy requirements, granularity, required measurement bandwidth, periodicity, and detection sensitivity, as well as information about sensing permission and synchronization. Regulatory requirements are expressed in the form of sensing and sensing control parameters.

NOTE—The term "regulatory information" is used synonymously in conjunction with the exchange of sensing-related information.

sensing: In the context of radio frequency spectrum, it refers to the act of measuring information indicative of spectrum occupancy (information may include frequency ranges, signal power levels, bandwidth, location information, etc.). Sensing may include determining how the sensed spectrum is used (cf. Clause 6).

sensing control information: Information describing the status and configuration of IEEE 1900.6 logical entities as well as information controlling and configuring the acquisition and processing of sensing information.

sensing information: Any information acquired by and obtained from sensors, including related spatio-temporal state information such as position, time, and confidence of acquisition.

sensor: A logical entity that performs sensing (see the definition of sensing in 3.1) within a radio system. Sensors may also act as clients of other Sensors.

NOTE—For the remainder of the document, any entity described as "IEEE 1900.6 logical entity" denotes a logical entity compliant to this standard.

3.2 Acronyms and abbreviations

A-SAP	application service access point
A/D	analog-to-digital
ADC	analog-to-digital conversion
AP	access point
BS	base station
C-SAP	communication service access point
CE	cognitive engine
CR	cognitive radio
D/A	digital-to-analog
DA	data archive
DAC	digital-to-analog conversion
DLC	data link control
DRRUO	distributed radio resource usage optimization
DSA	dynamic spectrum access
DSM	distributed sensing models
DSS	dynamic spectrum sharing
ISM	industrial, scientific, and medical
LTE	long -term evolution
M-SAP	measurement service access point
MAC	media access control
MCM	mobile node spectrum coordination model
NCM	network node spectrum coordination model
NRM	network reconfiguration manager
OSI	open systems interconnection
PHY	physical layer
PDA	personal digital assistant

PDU	protocol data unit
RAN	radio access network
RAT	radio access technology
RF	radio frequency
SAP	service access point
SeEM	sensing enhanced model
SEM	service enhanced model
SME	station management entity
SDU	service data unit
SPOLD	substitute for primary operation of longer duration
SPOSD	substitute for primary operation of shorter duration
TRM	terminal reconfiguration manager
ULME	upper layer management entity
UML	Unified Modeling Language
UTC	Coordinated Universal Time
WLAN	wireless local area network

4. System model

The IEEE 1900.6 system model describes the relationship of logical entities and interfaces detailed by subsequent clauses of this standard. The description of the interaction of these entities across their interfaces shall be the purpose of the IEEE 1900.6 reference model (Clause 5).

The system model has been derived from the requirements implied by the use cases given in Annex A. It respects various spectrum usage models, distributed sensing models, and sensing topologies as elaborated further by Annex A. The scenarios given by this clause complement each other and shall be seen as applications of the IEEE 1900.6 system model, each of them satisfying a given purpose and a certain subset of requirements as detailed by Annex A.

The two main categories of spectrum usage models are a long-term spectrum usage model and a short-term spectrum usage model. The terms "long term" and "short term" are understood within the context of use cases as related to the observed period of activity of the licensed/primary spectrum user.

— Long-term spectrum usage model: Spectrum is used over a relative long period (e.g., emergency services; please refer to A.1.1.1).

— Short-term spectrum usage model: Spectrum is used over short periods (e.g., ad hoc licensee service; please refer to A.1.2.2).

The abstract, high-level system model presented in this subclause is not dependent on topology or the spectrum usage model. Based on this system model, the use cases detailed in Annex A have been classified within each usage as shown in 4.1.

4.1 Scenario 1: Single CE/DA and single Sensor

In this scenario as shown in Figure 2, a single Sensor provides sensing-related information to one CE/DA. It is denoted as "1:1 scenario." The following instance of the IEEE 1900.6 logical interface is involved:

— CE/DA–S interface

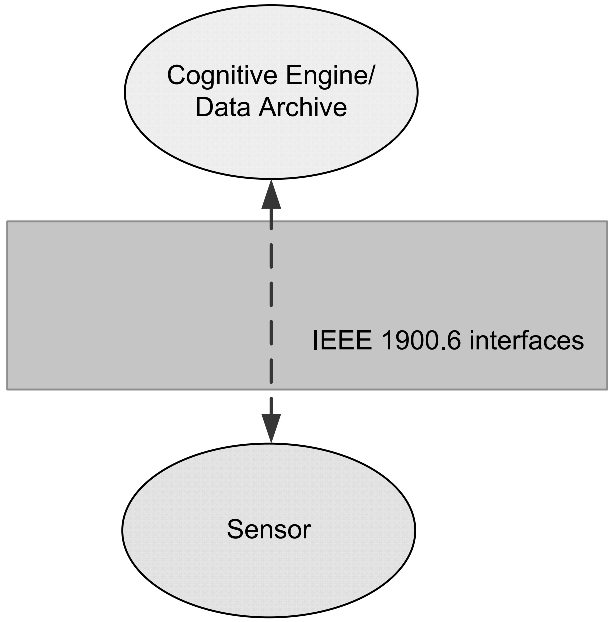


Figure 2—Single CE/DA and single Sensor scenario

4.2 Scenario 2: Single CE/DA and multiple Sensors

In this scenario as shown in Figure 3, multiple Sensors provide sensing-related information to a CE/DA. One CE/DA can access sensing-related information from multiple Sensors. It is denoted as “1:N scenario” where N is the number of Sensors.

The following instances of the IEEE 1900.6 logical interface are involved:

- CE/DA–S interface
- S–S interface

In distributed sensing, the CE/DA can access multiple Sensors and make dynamic spectrum access decisions based on sensing-related information from distributed Sensors. Also, multiple Sensors can exchange information and provide the CE/DA with an improved sensing result.

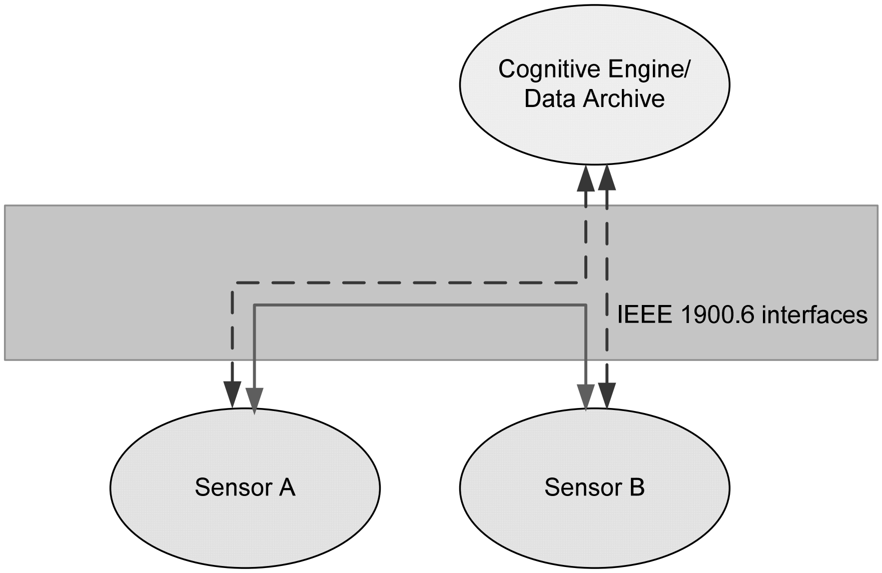


Figure 3—Single CE/DA and multiple Sensors scenario

4.3 Scenario 3: Multiple CE/DA and single Sensor

In this scenario as shown in Figure 4, multiple CE/DAs access sensing-related information from one Sensor. One Sensor provides sensing-related information to multiple CE/DAs. If some of the multiple CE/DAs are capable of accessing a Sensor, the CE/DA can exchange sensing-related information with other CE/DAs. This scenario is denoted as "M:1 scenario" where M is the number of CE/DAs.

The following instances of the IEEE 1900.6 logical interface are involved:

- CE/DA–S interface
- CE–CE/DA interface

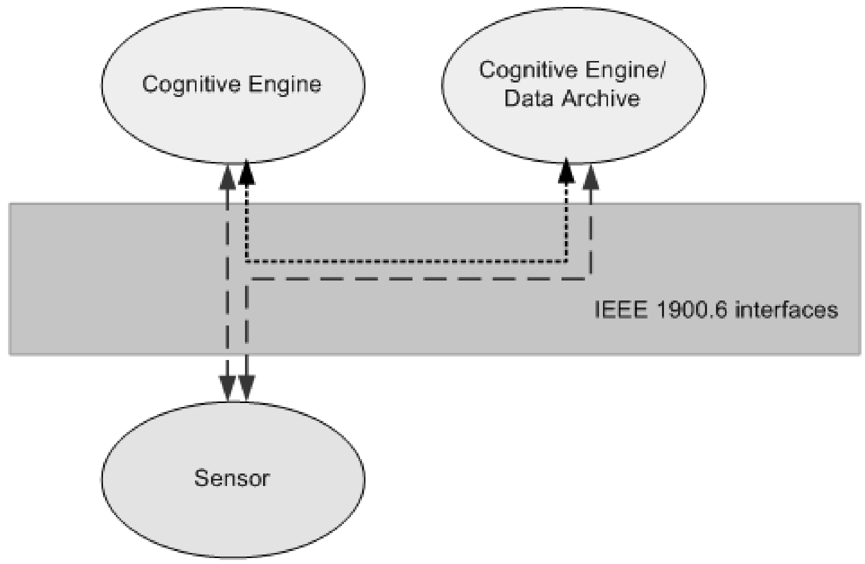


Figure 4—Multiple CE/DA and single Sensor scenario

Note that for the case of multiple CE/DAs and multiple Sensors, (i.e., the M:N scenario referred to in A.2.3), the system can be decomposed into a combination of the above three scenarios. This is understood within the context of model description. This does not imply decomposition at the implementation level.

5. The IEEE 1900.6 reference model

5.1 General description

Figure 5 shows the reference model for the IEEE 1900.6 interface and defined entities (i.e., Sensor, CE, and DA). The three SAPs shown in Figure 5 are a realization of the IEEE 1900.6 interface. Sensors and their clients may have all of the three SAPs or a subset of the SAPs depending on the implementation. The IEEE 1900.6 logical entities may also follow reference model views as depicted in Figure 6 through Figure 8, which show the instantiation of the IEEE 1900.6 interface as CE/DA–S, CE–CE/DA, S–S given in Figure 1. They are utilizing distinct SAPs to realize the IEEE 1900.6 logical interface. A compliant realization of the interface shall be sufficient to comply with the standard. No assumption is made on the realization of other functions of the device or devices involved. The figures show how the previously mentioned SAPs are involved in the sensing-related information exchange between Sensors and their clients. Subclause 5.1 provides formal definitions for each of the entities identified in Figure 5 through Figure 8.

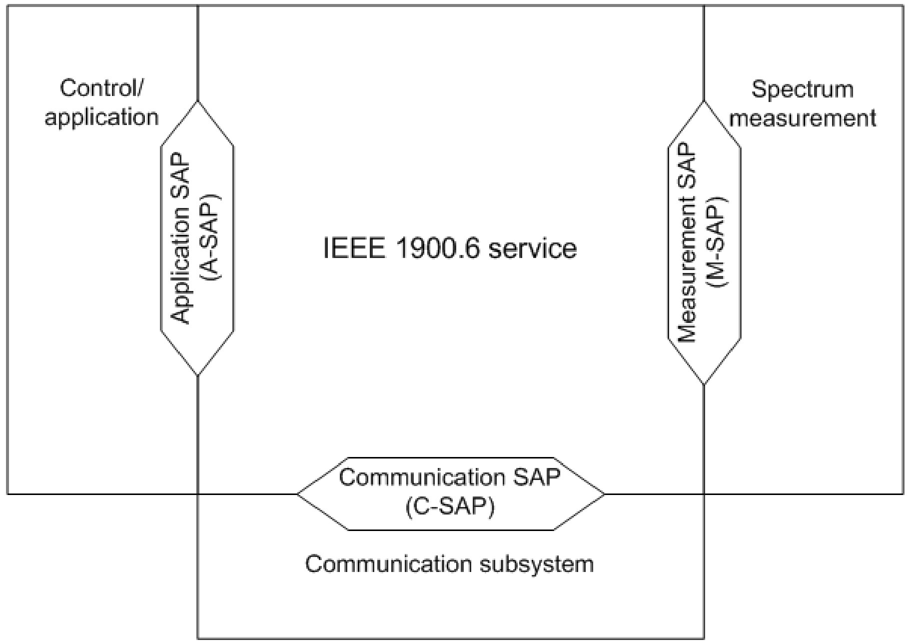


Figure 5—Reference model of the IEEE 1900.6 interface

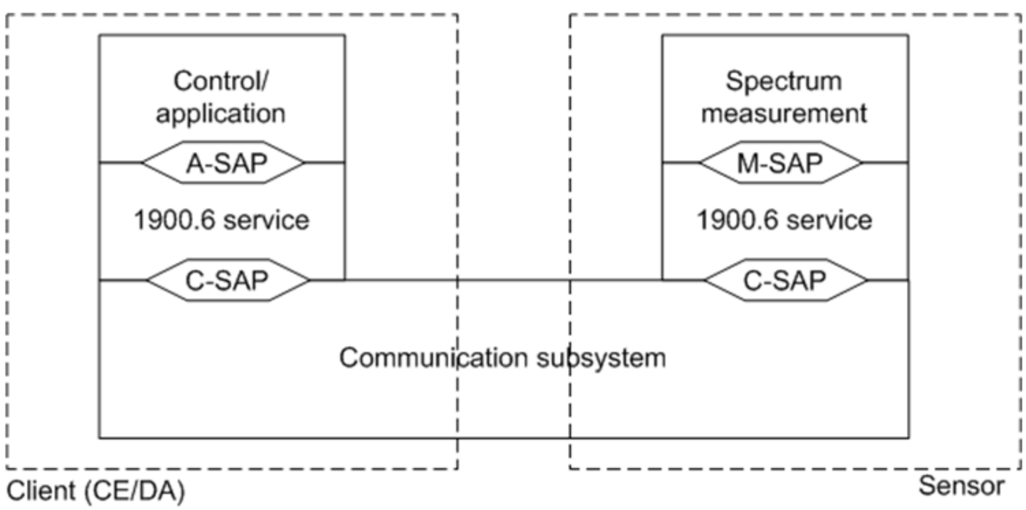


Figure 6—View of reference model when the client of Sensor is a CE or DA

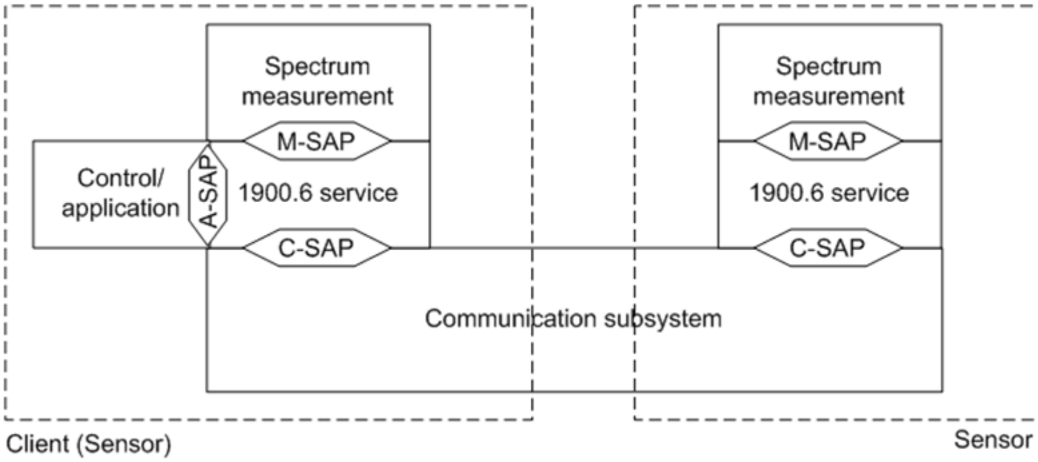


Figure 7—View of reference model when the client of Sensor is a Sensor

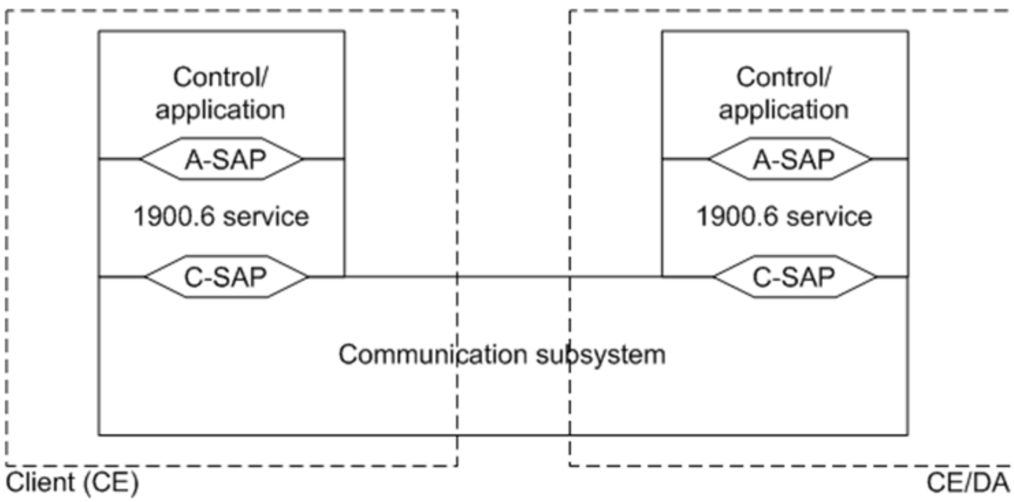


Figure 8—View of reference model for sensing-related information exchange between CE and CE/DA

The IEEE 1900.6 service is responsible for realizing IEEE 1900.6 service primitives and for generating data structures defined by this standard for the exchange of sensing-related information between IEEE 1900.6 servers and clients.

An **IEEE 1900.6 service access point (SAP)** is a conceptual location where an IEEE 1900.6 logical entity or an IEEE 1900.6 service user, which is not an IEEE 1900.6 logical entity by itself, can request the services provided by another IEEE 1900.6 logical entity or by an IEEE 1900.6 compliant station service (see 3.1). For example, an IEEE 1900.6 logical entity “Sensor” can utilize physical radio frequency (RF) sensor capabilities provided by the station (e.g., by the terminal radio hardware and firmware) and can provide RF measurement data to an IEEE 1900.6 logical entity CE upon request. The term “IEEE 1900.6 compliant service” here refers to the fact that primitives, parameters, and data structures defined throughout this standard are understood by this service and that measurement data returned are formatted accordingly.

The application service access point (A-SAP) is used by applications to access services of an IEEE 1900.6 compliant server. It defines a set of generic primitives and data structures to control the

IEEE 1900.6 logical entity and/or to obtain the sensing results for application purposes. The IEEE 1900.6 client utilizing the A-SAP takes the role of an information consumer and of a control application.

The A-SAP is used by an application (i.e., the user of the A-SAP) to utilize sensing-related information for its purpose (e.g., for policy investigation and analysis of spectrum usage). The A-SAP may provide functions to set up a configuration of IEEE 1900.6 logical entities (e.g., Sensors and CE), to configure these for collaborative sensing, to start the data acquisition and processing (e.g., policy processing), and to obtain the results of the processing in order to configure the RF interface accordingly. The A-SAP shall be instantiated by the IEEE 1900.6 logical entity.

The measurement service access point (M-SAP) is used for the exchange of sensing-related information between the spectrum measurement module and its M-SAP user. The user of the measurement module is realized by dedicated functions of the IEEE 1900.6 service. It is used by IEEE 1900.6 logical entities to access IEEE 1900.6 compliant services provided by the station's hardware and/or firmware to control the spectrum measurement module (such as a collocated physical spectrum measurement module, i.e., ADC/DAC, filtering, signal conditioning, etc.) and to acquire spectrum measurement data. For example, a station (terminal) utilizes its RF interface during idle times for spectrum measurement and provides RF spectrum data to collocated IEEE 1900.6 "sensor" entities that are registered at the local M-SAP. The M-SAP shall be instantiated by the station's IEEE 1900.6 compliant measurement function.

The communication service access point (C-SAP) exchanges sensing-related information (sensing information, sensor information, control information, and requirements derived from regulation) between remote IEEE 1900.6 servers and their clients. It abstracts communication mechanisms for use by IEEE 1900.6 service providers and service users through a dedicated set of generic service primitives. The client role can be taken by a Sensor, CE, or DA. It abstracts communication mechanisms for use by IEEE 1900.6 services through defining a set of generic primitives and mapping these primitives to transport protocols.

An IEEE 1900.6 compliant message transport service shall be provided by the station in order to locate a remote IEEE 1900.6 peer entity and to establish a communication link with this entity. Message exchange then takes place as defined by this standard and it is the responsibility of the transport service to map these message transfers to a suitable transport, network, or link layer communication. For example, an IEEE 1900.6 logical entity CE can take the role of a client to a remote IEEE 1900.6 Sensor by sending a request to configure the sensor for delivering measurement data to the requestor utilizing its C-SAP functionality. The C-SAP shall be instantiated by the IEEE 1900.6 compliant message transport service.

Control/application in the reference model refers to a hardware/software module that utilizes functions of the A-SAP to obtain sensing information or other information or to control and configure the behavior and functions of other IEEE 1900.6 logical entities and functions.

Communication subsystem in the reference model refers to a hardware/software module that provides the levels of the communications protocol stack and other communication services required by the IEEE 1900.6 services. The communication subsystem can be implemented with the IEEE 1900.6 compliant message transport service.

Spectrum measurement in the reference model is realized by the spectrum measurement module, a hardware/software module that provides radio spectrum measurement functions (e.g., analog-to-digital conversion [ADC]/digital-to-analog conversion [DAC], filtering, and signal conditioning).

5.2 An implementation example of the IEEE 1900.6 reference model

This subclause presents an informative example of realizing the IEEE 1900.6 reference model presented in 5.1 based on the reference model for an IEEE 802.11 station (see IEEE Std 802.11™-2007 [B3]). Figure 9 shows a case where the IEEE 1900.6 service and corresponding SAPs are implemented by the station management entity (SME). The IEEE 1900.6 application (control/application) is implemented on the application layer and the sensor (spectrum measurement) is implemented on the physical layer (PHY). The IEEE 1900.6 SAPs may be provided through different sublayer management entities and their related SAPs. This figure applies to instantiation of both (IEEE 1900.6) client and server.

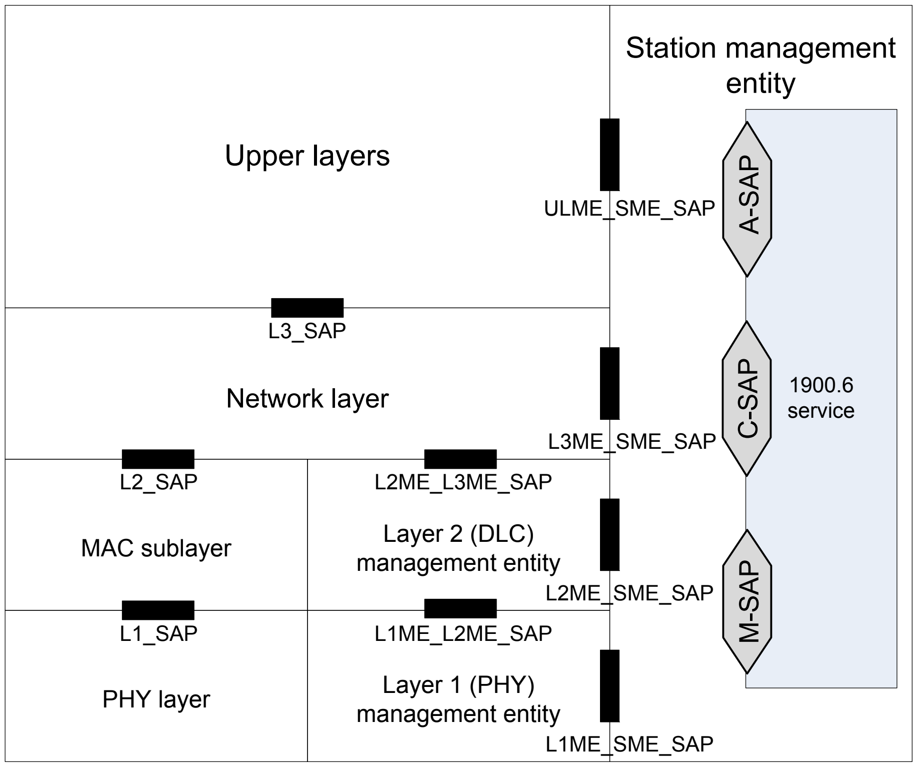


Figure 9—An implementation example of the IEEE 1900.6 service

5.3 Service access points

This standard defines the following SAPs that are involved in the exchange of sensing-related information between Sensors and their clients. Abstract service primitives describe the interaction with the IEEE 1900.6 services through SAPs in an implementation independent way.

NOTE—This standard does not distinguish between the SAPs at the local and remote entities. Hence, interaction at the SAP is only described using request and response primitives. If for implementation purposes, distinguishing of local and remote entities is necessary, the additional primitives, indication, and confirmation have the same semantics of the request and response primitives, correspondingly.

A number of parameters to primitives used throughout this clause are specified but not qualified nor detailed further in their effect on the behavior of a potential implementation. This has been done intentionally to remain neutral with respect to specific technologies and the demand for specific functions and algorithms of entities and services considered. Detailed descriptions of parameters and data representation are given in 6.3 and 6.4, respectively.

This standard specifies the information flow between sensors and their clients. The specification of a related message exchange protocol is left to future versions. Subclause 6.1 addresses the information exchange by giving the structure of the commands and parameters in that flow. The information structure given in this subclause defines how the primitives can be implemented for the information exchange. The current standard does not cover addressing of logical entities. It is assumed that a platform service exists that performs the mapping between the IDs of logical entities and proper device or network addresses.

5.3.1 Measurement service access point

The M-SAP provides a set of generic primitives or method for IEEE 1900.6 SAP users to control spectrum sensing and to obtain sensing-related information from spectrum sensing.

The IEEE 1900.6 logical entities obtain the following services from the M-SAP:

— Measurement capabilities discovery services
— Measurement configuration discovery services
— Measurement configuration services
— Information services

5.3.1.1 Measurement capabilities discovery services

The measurement capabilities discovery services provide a set of primitives or method through which the IEEE 1900.6 SAP user obtains the information related to the measurement capabilities of the associated measurement module.

The primitives of the measurement capabilities discovery services defined as part of the M-SAP are described in the following table.

Primitives	Description
Get_Supported_Spectrum_Measurement_Description	This primitive is used by the IEEE 1900.6 SAP user to obtain the information related to the capability of the spectrum measurement module.

5.3.1.1.1 Get_Supported_Spectrum_Measurement_Description.request

Function

This primitive is used by the IEEE 1900.6 SAP user to obtain the supported measurement of the measurement module.

Semantics of the service primitive

```
Get_Supported_Spectrum_Measurement_Description.request
                              (
                              MeasurementCapabilityTransID
                              )
```

Parameters

Name	Type	Description
MeasurementCapabilityTransID	Unsigned integer	MeasurementCapabilityTransID uniquely identifies one transaction of requesting a measurement feature (type and range) within the capability of the measurement module.

When used

This primitive is used by the IEEE 1900.6 SAP user when it needs to discover the spectrum measurement capabilities of the entity.

Effect of receipt

The spectrum measurement module subsequently uses a Get_Supported_Spectrum_Measurement_Description.reponse to reflect the results of the request.

5.3.1.1.2 Get_Supported_Spectrum_Measurement_Description.response

Function

This primitive returns the results of the request to obtain the description of spectrum measurement capabilities.

Semantics of the service primitive

```
Get_Supported_Spectrum_Measurement_Description.response
                                              (
                                              MeasurementCapabilityTransID,
                                              Status,
                                              MeasuRange,
                                              SensingMode,
                                              DataSheet.ADDAResolution,
                                              DataSheet.AngleResolution,
                                              DataSheet.FrequencyResolution,
                                              DataSheet.LocationTimeCapability,
                                              DataSheet.LoggingFunctions,
                                              DataSheet.RecordingCapability,
                                              DataSheet.SweepTime,
                                              LockStatus
                                              )
```

Parameters

Name	Type	Description
MeasurementCapabilityTransID	Unsigned integer	MeasurementCapabilityTransID uniquely identifies one transaction of requesting a measurement feature (type and range) within the capability of the measurement module.
Status	Enumeration	Status of operation. 0: success 1: unspecified failure 2: rejection 3: authorization failure
MeasuRange	Bandwidth	The MeasuRange parameter indicates the range of frequency that can be measured according to the manufacturer specification (cf. 6.3.27).
SensingMode	Enumeration	The SensingMode parameter indicates the numbers and types of standard sensing methods supported by the sensor (cf. 6.3.28).
DataSheet.ADDAResolution	Unsigned integer	The resolution of AD/DA convertor of the spectrum measurement module (cf. 6.3.32).
DataSheet.AngleResolution	Angle	The angle resolution of the spectrum measurement module (cf. 6.3.32).
DataSheet.FrequencyResolution	Frequency	Frequency resolution of the spectrum measurement module (cf. 6.3.32).
DataSheet.LocationTimeCapability	String	The capability of spectrum module in determining the location and time of the measurement (cf. 6.3.32).
DataSheet.LoggingFunctions	Unsigned integer	The ability of access log files (cf. 6.3.32).
DataSheet.RecordingCapability	String	The capability to recording data at the spectrum measurement module (cf. 6.3.32).
DataSheet.SweepTime	Microsecond	The time to sweep between two frequencies (cf. 6.3.32).
LockStatus	Enumeration	The LockStatus indicates whether the entity (i.e., the sender of this primitive) has been locked by a client for exclusive service. 1: The status of the entity is locked. The entity is locked by a client for exclusive service. 0: 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.

When used

The primitive is used in response to the Get_Supported_Spectrum_Measurement_Description.request primitive.

Effect of receipt

Upon receipt, the IEEE 1900.6 SAP user obtains the information related to the spectrum measurement capabilities of the spectrum measurement module.

5.3.1.2 Measurement configuration discovery services

The measurement configuration discovery services provide a set of primitives or method through which the IEEE 1900.6 SAP user obtains the information related to the measurement configuration of the spectrum measurement module. The following information is discovered through these services:

- Sensor PHY profile
- Sensor antenna profile
- Sensor location

The primitives of the measurement configuration services defined as part of the M-SAP are described in the following table.

Primitives	Description
Sensor PHY profile	
Get_Sensor_PHY _Description	This primitive is used by the IEEE 1900.6 SAP user to obtain the information related to physical configuration of the spectrum measurement module.
Sensor antenna profile	
Get_Sensor_Antenna _Description	This primitive is used by the IEEE 1900.6 SAP user to obtain the information related to the antenna configuration of the spectrum measurement module.
Sensor location	
Get_Sensor_Location _Description	This primitive is used by the IEEE 1900.6 SAP user to obtain the information related to the location of the spectrum measurement module. Note that the spectrum measurement module represents the location of the measurement.

5.3.1.2.1 Get_Sensor_PHY_Description.request

Function

This primitive is used by the IEEE 1900.6 SAP user to obtain the PHY description of the spectrum measurement module.

Semantics of the service primitive

```
Get_Sensor_PHY_Description.request
                                        (
                                        SensorPHYProfileID
                                        )
```

Parameters

Name	Type	Description
SensorPHYProfileID	Unsigned integer	SensorPHYProfileID uniquely defines the PHY profile of the spectrum measurement module.

When used

This primitive is used by the IEEE 1900.6 SAP user when it needs to discover the PHY configuration of the spectrum measurement module.

Effect of receipt

The spectrum measurement module subsequently uses a Get_Sensor_PHY_Description.response to reflect the results of the request.

5.3.1.2.2 Get_Sensor_PHY_Description.response

Function

This primitive returns the results of the request to obtain the description of the PHY profile of the spectrum measurement module.

Semantics of the service primitive

```
Get_Sensor_PHY_Description.response
                                    (
                                    SensorPHYProfileID,
                                    Status,
                                    Datasheet.CalibrationData,
                                    Datasheet.CalibrationMethod,
                                    Datasheet.ChannelFiltering,
                                    Datasheet.DynamicRange,
                                    Datasheet.NoiseFactor,
                                    Datasheet.PhaseNoise,
                                    LockStatus
                                    )
```

Parameters

Name	Type	Description
SensorPHYProfileID	Unsigned integer	SensorPHYProfileID uniquely defines the PHY profile of the spectrum measurement module.
Status	Enumeration	Status of operation. 0: success 1: unspecified failure 2: rejection 3: authorization failure
Datasheet.CalibrationData	String	Data used to calibrate the spectrum measurement module (cf. 6.3.32).
Datasheet.CalibrationMethod	String	Method to calibrate the spectrum measurement module. (cf. 6.3.32).
Datasheet.ChannelFiltering	String	The filtering function of the spectrum measurement module (cf. 6.3.32).
Datasheet.DynamicRange	Float	Dynamic range in dB of signal detection of the spectrum measurement module (cf. 6.3.32).
Datasheet.NoiseFactor	Float	The ratio in linear scale of the noise produced by the spectrum measurement module to the thermal noise (cf. 6.3.32).
Datasheet.PhaseNoise	Float	Phase noise in dBc/Hz produced by the spectrum measurement module (cf. 6.3.32).
LockStatus	Enumeration	The LockStatus indicates whether the entity (i.e., the sender of this primitive) has been locked by a client for exclusive service. 1: The status of the entity is locked. The entity is locked by a client for exclusive service. 0: 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.

When used

The primitive is used in response to the Get_Sensor_PHY_Description.request primitive.

Effect of receipt

Upon receipt, the IEEE 1900.6 SAP user obtains the information related to the PHY profile of the spectrum measurement module.

5.3.1.2.3 Get_Sensor_Antenna_Description.request

Function

This primitive is used by the IEEE 1900.6 SAP user to obtain the antenna description of the spectrum measurement module.

Semantics of the service primitive

```
Get_Sensor_Antenna_Description.request
                                        (
                                        SensorAntennaProfileID
                                        )
```

Parameters

Name	Type	Description
SensorAntennaProfileID	Unsigned integer	SensorAntennaProfileID uniquely defines the antenna profile of the spectrum measurement module.

When used

This primitive is used by the IEEE 1900.6 SAP user when it needs to discover the antenna configuration of the spectrum measurement module.

Effect of receipt

The spectrum measurement module subsequently uses a Get_Sensor_Antenna_Description.response to reflect the results of the request.

5.3.1.2.4 Get_Sensor_Antenna_Description.response

Function

This primitive returns the results of the request to obtain the description of the antenna profile of the spectrum measurement module.

Semantics of the service primitive

```
Get_Sensor_Antenna_Description.response
                                        (
                                        SensorAntennaProfileID,
                                        Status,
                                        Datasheet.AntennaBandwidth,
                                        Datasheet.AntennaBeamPointing,
```

Datasheet.AntennaBeamwidth,
Datasheet.AntennaDirectivityGain,
Datasheet.AntennaGain,
Datasheet.AntennaHeight,
Datasheet.AntennaPolarization,
LockStatus
)

Parameters

Name	Type	Description
SensorAntennaProfileID	Unsigned integer	SensorAntennaProfileID uniquely defines the antenna profile of the spectrum measurement module.
Status	Enumeration	Status of operation. 0: success 1: unspecified failure 2: rejection 3: authorization failure
Datasheet.AntennaBandwidth	Array(Frequency)	Bandwidth of the antenna used at the spectrum measurement module (cf. 6.3.32).
Datasheet.AntennaBeamPointing	Array(Angle)	The DataSeet.AntennaBeamPointing parameter specifies the beam pointing direction of the antenna used at the spectrum measurement module by giving the azimuthal angle with respect to North and elevation angle with respect to the horizon (cf. 6.3.32).
Datasheet.AntennaBeamwidth	Array(Angle)	Beamwidth of the antenna used at the spectrum measurement module, normally specified as half-power horizontal and vertical beamwidth (cf. 6.3.32).
Datasheet.AntennaDirectivityGain	Float	Directivity gain in dBi of the antenna radiation pattern at the spectrum measurement module (cf. 6.3.32).
Datasheet.AntennaGain	Float	Power gain in dB of the antenna used at the spectrum measurement module (cf. 6.3.32).
Datasheet.AntennaHeight	Float	Height of the antenna in meters with respect to sea level (cf. 6.3.32).
Datasheet.AntennaPolarization	Enumeration	Polarization of the antenna used at the spectrum measurement module (cf. 6.3.32). 0: Linear polarization 1: Circular polarization 2: Elliptical polarization
LockStatus	Enumeration	The LockStatus indicates whether the entity (i.e., the sender of this primitive) has been locked by a client for exclusive service. 1: The status of the entity is locked. The entity is locked by a client for exclusive service. 0: 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.

When used

The primitive is used in response to the Get_Sensor_Antenna_Description.request primitive.

Effect of receipt

Upon receipt, the IEEE 1900.6 SAP user obtains the information related to the antenna profile of the spectrum measurement module.

5.3.1.2.5 Get_Sensor_Location_Description.request

Function

This primitive is used by the IEEE 1900.6 SAP user to obtain the location information of the spectrum measurement module.

Semantics of the service primitive

Get_Sensor_Location_Description.request
(
SensorLocationTransID
)

Parameters

Name	Type	Description
SensorLocationTransID	Unsigned integer	SensorLocationTransID uniquely defines one transaction of a requesting the location of the spectrum measurement module.

When used

This primitive is used by the IEEE 1900.6 SAP user when it needs to discover the location of the spectrum measurement module.

Effect of receipt

The spectrum measurement module subsequently uses a Get_Sensor_Location.response to reflect the results of the request.

5.3.1.2.6 Get_Sensor_Location_Description.response

Function

This primitive returns the results of the request to obtain the location information of the spectrum measurement module.

Semantics of the service primitive

Get_Sensor_Location_Description.response
(
SensorLocationTransID,
Status,
AbsSensorLocation,
RelSensorLocation,
LockStatus
)

Parameters

Name	Type	Description
SensorLocationTransID	Unsigned integer	SensorLocationTransID uniquely defines one transaction of a requesting the location of the spectrum measurement module.
Status	Enumeration	Status of operation. 0: success 1: unspecified failure 2: rejection 3: authorization failure
AbsSensorLocation	RGeolocation	The absolute location of the spectrum measurement module (cf. 6.3.25).
RelSensorLocation	RGeolocation	The RelSensorLocation parameter is used when the position of the sensor is measured by triangulation with respect to known reference positions (cf. 6.3.26).
LockStatus	Enumeration	The LockStatus indicates whether the entity (i.e., the sender of this primitive) has been locked by a client for exclusive service. 1: The status of the entity is locked. The entity is locked by a client for exclusive service. 0: 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.

When used

The primitive is used in response to the Get_Sensor_Location_Description.request primitive.

Effect of receipt

Upon receipt, the IEEE 1900.6 SAP user obtains the location information of the spectrum measurement module. The sensor location can be expressed by either AbsSensorLocation or RelSensorLocation. If RelLocation is used, the reference positions are assumed to be known.

5.3.1.3 Measurement configuration services

The measurement configuration services provide a set of primitives or method through which the IEEE 1900.6 SAP user configures the spectrum measurement module. The following configuration is set through these services:

- Measurement objective
- Measurement profile
- Measurement performance

The primitives of the measurement configuration services are defined as part of the M-SAP and are described in the following table.

Primitives	Description
Measurement objective	
Set_Sensor_Measurement_Obj	This primitive is used by the IEEE 1900.6 SAP user to set the measurement objective of the spectrum measurement module.
Measurement profile	
Set_Sensor_Measurement_Profile	This primitive is used by the IEEE 1900.6 SAP user to set the measurement profile of the spectrum measurement module.
Measurement performance	
Set_Sensor_Measurement_Performance	This primitive is used by the IEEE 1900.6 SAP user to set the target performance of the spectrum measurement module.

5.3.1.3.1 Set_Sensor_Measurement_Obj.request

Function

This primitive is used by the IEEE 1900.6 SAP user to set the measurement objective of the spectrum measurement module.

Semantics of the service primitive

```
Set_Sensor_Measurement_Obj.request
                                   (
                                   SensorMeasurementObjID,
                                   StartTime,
                                   EndTime,
                                   Bandwidth
                                   )
```

Parameters

Name	Type	Description
SensorMeasurementObjID	Unsigned integer	SensorMeasurementObjID uniquely defines the measurement objective of the spectrum measurement module.
StartTime	TimeStamp	The StartTime parameter specifies the time stamp when spectrum measurement starts (cf. 6.3.8).
EndTime	TimeStamp	The EndTime parameter specifies the time stamp when spectrum measurement finishes (cf. 6.3.8).
Bandwidth	Array(Frequency)	Bandwidth parameter specifies the bandwidth, by giving the start frequency and stop frequency, to perform spectrum measurement (cf. 6.3.11).

When used

This primitive is used by the IEEE 1900.6 SAP user to set the measurement objective of the spectrum measurement module.

Effect of receipt

Parameters of the spectrum measurement module are subsequently set to the values given by the primitive. The spectrum measurement module subsequently uses a Set_Sensor_Measurement_Obj.response primitive to reflect the results of the request.

5.3.1.3.2 Set_Sensor_Measurement_Obj.response

Function

This primitive returns the results of the request to set the measurement objective of the spectrum measurement module.

Semantics of the service primitive

```
Set_Sensor_Measurement_Obj.response
                                    (
                                    SensorMeasurementObjID,
                                    Status,
                                    LockStatus
                                    )
```

Parameters

Name	Type	Description
SensorMeasurementObjID	Unsigned integer	SensorMeasurementObjID uniquely defines the measurement objective of the spectrum measurement module.
Status	Enumeration	Status of operation. 0: success 1: unspecified failure 2: rejection 3: authorization failure
LockStatus	Enumeration	The LockStatus indicates whether the entity (i.e., the sender of this primitive) has been locked by a client for exclusive service. 1: The status of the entity is locked. The entity is locked by a client for exclusive service. 0: The status of the entity is locked. The entity has been unlocked by its client, or has been unlocked by a self-generated timeout event.

When used

The primitive is used in response to the Set_Sensor_Measurement_Obj.request primitive.

Effect of receipt

Upon receipt, the spectrum measurement module sets the measurement objective.

5.3.1.3.3 Set_Sensor_Measurement_Profile.request

Function

This primitive is used by the IEEE 1900.6 SAP user to set the measurement profile of the spectrum measurement module.

Semantics of the service primitive

```
Set_Sensor_Measurement_Profile.request
                                        (
                                        SensorMeasurementProfileID,
                                        SensingMode,
                                        ChOrder,
                                        ChList,
                                        ReportRate,
                                        Scan.LowerThreshold,
                                        ReportingMode
                                        )
```

Parameters

Name	Type	Description
SensorMeasurementProfileID	Unsigned integer	SensorMeasurementProfileID uniquely defines the measurement profile of the spectrum measurement module.
SensingMode	Enumeration	The SensingMode parameter indicates the numbers and types of standard sensing methods supported by the sensor (cf. 6.3.28).
ChOrder	Vector(Unsigned integer)	The ChOrder specifies the scanning order of the channels. The channel order is represented by a predefined vector of step indices (sequence vector should use a bounded type rather than the generic Vector type). The default for this parameter is “sequential scanning in ascending order” (cf. 6.3.13).
ChList	Structured	The ChList specifies a list of frequency bands by providing a vector of channel numbers along with a corresponding vector of bandwidth parameters (cf. 6.3.32).
ReportRate	Structured	The ReportRate parameter is issued by the IEEE 1900.6 client to specify how often the measurement result should be reported or updated (e.g., every 2 s or every 100 ms) (cf. 6.3.14).
Scan.LowerThreshold	NoisePower	The Scan.LowerThreshold parameter specifies the noise floor (cf. 6.3.24).
ReportingMode	Enumeration	The ReportMode parameter is issued by the IEEE 1900.6 client to indicate in what form the sensing-related information should be reported (cf. 6.3.15).

When used

This primitive is used by the IEEE 1900.6 SAP user to set the measurement profile of the spectrum measurement module.

Effect of receipt

Parameters of the spectrum measurement module are subsequently set to the values given by the primitive. The spectrum measurement module subsequently uses a Set_Sensor_Measurement_Profile.response primitive to reflect the results of the request.

5.3.1.3.4 Set_Sensor_Measurement_Profile.response

Function

This primitive returns the results of the request to set the measurement profile of the spectrum measurement module.

Semantics of the service primitive

Set_Sensor_Measurement_Profile.response
(
SensorMeasurementProfileID,
Status,
LockStatus
)

Parameters

Name	Type	Description
SensorMeasurementProfileID	Unsigned integer	SensorMeasurementProfileID uniquely defines the measurement profile of the spectrum measurement module.
Status	Enumeration	Status of operation. 0: success 1: unspecified failure 2: rejection 3: authorization failure
LockStatus	Enumeration	The LockStatus indicates whether the entity (i.e., the sender of this primitive) has been locked by a client for exclusive service. 1: The status of the entity is locked. The entity is locked by a client for exclusive service. 0: 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.

When used

The primitive is used in response to the Set_Sensor_Measurement_Profile.request primitive.

Effect of receipt

Upon receipt, the IEEE 1900.6 SAP user sets the measurement profile of the spectrum measurement module.

5.3.1.3.5 Set_Sensor_Measurement_Performance.request

Function

This primitive is used by the IEEE 1900.6 SAP user to set the measurement performance of the spectrum measurement module.

Semantics of the service primitive

Set_Sensor_Measurement_Performance.request
(
SensorMeasurementPerformanceID,
PerfMetric.Pd,
PerfMetric.Pf
)

Parameters

Name	Type	Description
SensorMeasurementPerformanceID	Unsigned integer	SensorMeasurementPerformanceID uniquely defines the measurement performance of the spectrum measurement module.
PerfMetric.Pd	Unsigned integer	The PerfMetric.Pd specifies the rate of detection that the spectrum measurement should achieve (cf. 6.3.16).
PerfMetric.Pf	Unsigned integer	The PerfMetric.Pf specifies the rate of false alarm that the spectrum measurement should achieve (cf. 6.3.16).

When used

This primitive is used by the IEEE 1900.6 SAP user to set the measurement performance of the spectrum measurement module.

Effect of receipt

Parameters of the spectrum measurement module are subsequently set to the values given by the primitive. The spectrum measurement module subsequently uses a Set_Sensor_Measurement_Performance.response to reflect the results of the request.

5.3.1.3.6 Set_Sensor_Measurement_Performance.response

Function

This primitive returns the results of the request to set the measurement performance of the spectrum measurement module.

Semantics of the service primitive

Set_Sensor_Measurement_Performance.response
(
SensorMeasurementPerformanceID,
Status,
LockStatus
)

Parameters

Name	Type	Description
SensorMeasurementPerformanceID	Unsigned integer	SensorMeasurementPerformanceID uniquely defines the measurement performance of the spectrum measurement module.
Status	Enumeration	Status of operation. 0: success 1: unspecified failure 2: rejection 3: authorization failure 4: function not supported
LockStatus	Enumeration	The LockStatus indicates whether the entity (i.e., the sender of this primitive) has been locked by a client for exclusive service. 1: The status of the entity is locked. The entity is locked by a client for exclusive service. 0: 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.

When used

The primitive is used in response to the Set_Sensor_Measurement_Performance.request primitive.

Effect of receipt

Upon receipt, the spectrum measurement module sets the measurement performance.

5.3.1.4 Information services

The information services provide a set of primitives or methods through which the IEEE 1900.6 SAP user collects the information related to the measurement configuration of the spectrum measurement module. The following information is collected through these services:

- Sensor profile
- Measurement profile
- Signal measurement
- Channel measurement
- Radio access technology (RAT) measurement
- Management

The primitives of information services are defined as part of the M-SAP in the following table.

Primitives	Description
Sensor profile collection	
Get_Sensor_Manufacturer_Profile	This primitive is used by the IEEE 1900.6 SAP user to collect the information related to the manufacturer of the spectrum measurement module.
Get_Sensor_Power_Profile	This primitive is used by the IEEE 1900.6 SAP user to collect the information related to the power consumption of the spectrum measurement module.
Measurement profile collection	
Get_Measurement_Profile	This primitive is used by the IEEE 1900.6 SAP user to obtain the information related to the spectrum measurement that has been carried out.
Get_Measurement_Location_Information	This primitive is used by the IEEE 1900.6 SAP user to collect the information related to the location where the spectrum measurement has been carried out.
Signal measurement collection	
Get_Signal_Measurement_Value	This primitive is used by the IEEE 1900.6 SAP user to collect the measured value related to the measured signals.
Channel measurement collection	
Get_Channel_Measurement_Value	This primitive is used by the IEEE 1900.6 SAP user to collect the measured value related to the measured channels.
RAT measurement collection	
Get_RAT_ID_Value	This primitive is used by the IEEE 1900.6 SAP user to collect the measured value related to the measured RAT.
Management	
Notify	This primitive is used by the measurement module to report its status to IEEE 1900.6 SAP user.

5.3.1.4.1 Get_Sensor_ Manufacturer_Profile.request

Function

This primitive is used by the IEEE 1900.6 SAP user to obtain the manufacturer information of the spectrum measurement module.

Semantics of the service primitive

```
Get_Sensor_Manufacturer_Profile.request
                                        (
                                        SensorManufacturerProfileID,
                                        )
```

Parameters

Name	Type	Description
SensorManufacturerProfileID	Unsigned integer	SensorManufacturerProfileID uniquely defines the manufacturer profile of the spectrum measurement module.

When used

This primitive is used by the IEEE 1900.6 SAP user when it needs to collect the manufacturer profile of the spectrum measurement module.

Effect of receipt

The spectrum measurement module subsequently uses a Get_Sensor_Manufacturer_Profile.response primitive to reflect the results of the request.

5.3.1.4.2 Get_Sensor_Manufacturer_Profile.response

Function

This primitive returns the results of the request to collect the manufacturer profile of the spectrum measurement module.

Semantics of the service primitive

```
Get_Sensor_Manufacturer_Profile.response
                              (
                              SensorManufacturerProfileID ,
                              Status,
                              SensorID.VendorID,
                              SensorID.ProductID,
                              LockStatus
                              )
```

Parameters

Name	Type	Description
SensorManufacturerProfileID	Unsigned integer	SensorManufacturerProfileID uniquely defines the manufacturer profile of the spectrum measurement module.
Status	Enumeration	Status of operation. 0: success 1: unspecified failure 2: rejection 3: authorization failure
SensorID.VendorID	String	The SensorID.VendorID uniquely identifies the manufacturer of the sensor (cf. 6.3.29).
SensorID.ProductID	String	The SensorID.ProductID uniquely identifies the product of a manufacturer (cf. 6.3.29).
LockStatus	Enumeration	The LockStatus indicates whether the entity (i.e., the sender of this primitive) has been locked by a client for exclusive service. 1: The status of the entity is locked. The entity is locked by a client for exclusive service. 0: 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.

When used

The primitive is used in response to the Get_Sensor_Manufacturer_Profile.request primitive.

Effect of receipt

Upon receipt, the IEEE 1900.6 SAP user collects the manufacturer profile of the spectrum measurement module.

5.3.1.4.3 Get_Sensor_ Power_Profile.request

Function

This primitive is used by the IEEE 1900.6 SAP user to obtain the power profile of the spectrum measurement module.

Semantics of the service primitive

```
Get_Sensor_Power_Profile.request
                                    (
                                    SensorPowerProfileID,
                                    )
```

Parameters

Name	Type	Description
SensorPowerProfileID	Unsigned integer	SensorPowerProfileID uniquely defines the power profile of the spectrum measurement module.

When used

This primitive is used by the IEEE 1900.6 SAP user when it needs to collect the power profile of the spectrum measurement module.

Effect of receipt

The spectrum measurement module subsequently uses a Get_Sensor_Power_Profile.response primitive to reflect the results of the request.

5.3.1.4.4 Get_Sensor_Power_Profile.response

Function

This primitive returns the results of the request to collect the power profile of the spectrum measurement module.

Semantics of the service primitive

```
Get_Sensor_Power_Profile.response
                                    (
                                    SensorPowerProfileID,
                                    Status,
                                    BatteryStatus,
                                    Datasheet.PowerConsumption,
                                    LockStatus
                                    )
```

Parameters

Name	Type	Description
SensorPowerProfileID	Unsigned integer	SensorPowerProfileID uniquely defines the power profile of the spectrum measurement module.
Status	Enumeration	Status of operation. 0: success 1: unspecified failure 2: rejection 3: authorization failure
BatteryStatus	Unsigned integer	The BatteryStatus parameter indicates the remaining battery power in percentage (cf. 6.3.31).
Datasheet.PowerConsumption	String	The Datasheet.PowerConsumption parameter indicates the power consumption profile of the spectrum measurement module (cf. 6.3.32).
LockStatus	Enumeration	The LockStatus indicates whether the entity (i.e., the sender of this primitive) has been locked by a client for exclusive service. 1: The status of the entity is locked. The entity is locked by a client for exclusive service. 0: The status of the entity is unlocked. The entity is has been unlocked by its client, or has been unlocked by a self-generated timeout event.

When used

The primitive is used in response to the Get_Sensor_Power_Profile.request primitive.

Effect of receipt

Upon receipt, the IEEE 1900.6 SAP user collects the power profile of the spectrum measurement module.

5.3.1.4.5 Get_Sensor_Measurement_Profile.request

Function

This primitive is used by the IEEE 1900.6 SAP user to obtain the measurement profile of the spectrum measurement module.

Semantics of the service primitive

```
Get_Sensor_Measurement_Profile.request
                                      (
                                      SensorMeasurementProfileID,
                                      )
```

Parameters

Name	Type	Description
SensorMeasurementProfileID	Unsigned integer	SensorMeasurementProfileID uniquely defines the profile of spectrum measurement that has been carried out by the spectrum measurement module.

When used

This primitive is used by the IEEE 1900.6 SAP user when it needs to collect the measurement profile of the spectrum measurement module.

Effect of receipt

The spectrum measurement module subsequently uses a Get_Sensor_Measurement_Profile.response to reflect the results of the request.

5.3.1.4.6 Get_Sensor_Measurement_Profile.response

Function

This primitive returns the results of the request to collect the measurement profile of the spectrum measurement module.

Semantics of the service primitive

```
Get_Sensor_Measurement_Profile.response
                                        (
                                        SensorMeasurementProfileID,
                                        Status,
                                        ConfidenceLevel,
                                        ReportMode,
                                        SensingMode,
                                        ReportRate,
                                        TimeStamp,
                                        Scan.LowerThreshold,
                                        MeasuBandwidth,
                                        LockStatus
                                        )
```

Parameters

Name	Type	Description
SensorMeasurementProfileID	Unsigned integer	SensorMeasurementProfileID uniquely defines the profile of spectrum measurement that has been carried out by the spectrum measurement module.
Status	Enumeration	Status of operation. 0: success 1: unspecified failure 2: rejection 3: authorization failure
ConfidenceLevel	Structured	ConfidenceLevel indicates the degree of certainty of estimated value (cf. 6.3.33).
ReportMode	Enumeration	The ReportMode indicates either hard information or soft information (cf. 6.3.15).
SensingMode	Enumeration	The SensingMode parameter indicates what sensing method is used to carry out the spectrum measurement (cf. 6.3.28).
ReportRate	Structured	The ReportRate parameter is issued by the IEEE 1900.6 client to specify how often the measurement result should be reported or updated (e.g., every 2 s or every 100 ms) (cf. 6.3.14).
TimeStamp	Structured	This parameter indicates the time that the spectrum measurement has been carried out (cf. 6.3.8).
Scan.LowerThreshold	NoisePower	The Scan.LowerThreshold parameter indicates the noise threshold used in the spectrum experiment. This value is set to satisfy certain detection requirement (cf. 6.3.24).
MeasuBandwidth	Array(Frequency)	The MeasuBandwidth parameter is the same as the bandwidth used by the control parameter, but when the value is returned by the sensor, it indicates the measurement bandwidth that is actually used by the sensor (cf. 6.3.34).
LockStatus	Enumeration	The LockStatus indicates whether the entity (i.e., the sender of this primitive) has been locked by a client for exclusive service. 1: The status of the entity is locked. The entity is locked by a client for exclusive service. 0: 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.

When used

The primitive is used in response to the Get_Sensor_Measurement_Profile.request primitive.

Effect of receipt

Upon receipt, the IEEE 1900.6 SAP user collects the measurement profile of the spectrum measurement module.

5.3.1.4.7 Get_Measurement_Location_Information.request

Function

This primitive is used by the IEEE 1900.6 SAP user to obtain the measurement location information of the spectrum measurement module.

Semantics of the service primitive

```
Get_Measurement_Location_Information.request
                                (
                                MeasurementLocationTransID,
                                )
```

Parameters

Name	Type	Description
MeasurementLocationTransID	Unsigned integer	MeasurementLocationTransID uniquely defines one transaction of requesting the location where the spectrum measurement has been carried out by the spectrum measurement module.

When used

This primitive is used by the IEEE 1900.6 SAP user when it needs to collect the information related to the location where the spectrum measurement has been carried out by the spectrum measurement module.

Effect of receipt

The spectrum measurement module subsequently uses a Get_Measurement_Location_Information.response primitive to reflect the results of the request.

5.3.1.4.8 Get_Measurement_Location_Information.response

Function

This primitive returns the results of the request to collect the measurement location information of the spectrum measurement module.

Semantics of the service primitive

```
Get_Measurement_Location_Information.response
                                (
                                MeasurementLocationTransID,
                                Status,
                                AbsSensorLocation,
                                RelSensorLocation,
                                LockStatus
                                )
```

Parameters

Name	Type	Description
MeasurementLocationTransID	Unsigned integer	MeasurementLocationTransID uniquely defines one transaction of requesting the location where the spectrum measurement has been carried out by the spectrum measurement module.
Status	Enumeration	Status of operation. 0: success 1: unspecified failure 2: rejection 3: authorization failure
AbsSensorLocation	RGeolocation	The AbsSensorLocation parameter is the absolute location that the experiment has carried out (cf. 6.3.25).
RelSensorLocation	RGeolocation	The RelSensorLocation parameter is the relative location with respect to a known reference location that the experiment has been carried out (cf. 6.3.26).
LockStatus	Enumeration	The LockStatus indicates whether the entity (i.e., the sender of this primitive) has been locked by a client for exclusive service. 1: The status of the entity is locked. The entity is locked by a client for exclusive service. 0: 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.

When used

The primitive is used in response to the Get_Measurement_Location_Information.request primitive.

Effect of receipt

Upon receipt, the IEEE 1900.6 SAP user collects the measurement location information of the spectrum measurement module.

5.3.1.4.9 Get_Signal_Measurement_Value.request

Function

This primitive is used by the IEEE 1900.6 SAP user to collect signal measurement values of the spectrum measurement module.

Semantics of the service primitive

```
Get_Signal_Measurement_Value.request
                                (
                                SignalMeasurementID
                                )
```

Parameters

Name	Type	Description
SignalMeasurementID	Unsigned integer	SignalMeasurementID uniquely defines the set of measurement values of a signal obtained by the spectrum measurement module.

When used

This primitive is used by the IEEE 1900.6 SAP user when it needs to collect the signal measurement values of the spectrum measurement module.

Effect of receipt

The spectrum measurement module subsequently uses a Get_Signal_Measurement_Value.response primitive to reflect the results of the request.

5.3.1.4.10 Get_Signal_Measurement_Value.response

Function

This primitive returns the results of the request to collect the signal measurement values of the spectrum measurement module.

Semantics of the service primitive

```
Get_Signal_Measurement_Value.response
                                       (
                                       SignalMeasurementID,
                                       Status,
                                       SignalDesc,
                                       LockStatus
                                       )
```

Parameters

Name	Type	Description
SignalMeasurementID	Unsigned integer	SignalMeasurementID uniquely defines the set of measurement values of the signal obtained by the spectrum measurement module.
Status	Enumeration	Status of operation. 0: success 1: unspecified failure 2: rejection 3: authorization failure
SignalDesc	Structured	The signal description parameter represents a set of information elements that describe the behavior of the signal (e.g., traffic pattern type and modulation scheme (cf. 6.3.41).
LockStatus	Enumeration	The LockStatus indicates whether the entity (i.e., the sender of this primitive) has been locked by a client for exclusive service. 1: The status of the entity is locked. The entity is locked by a client for exclusive service. 0: 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.

When used

The primitive is used in response to the Get_Signal_Measurement_Value.request primitive.

Effect of receipt

Upon receipt, the IEEE 1900.6 SAP user collects the measurement values of signals by the spectrum measurement module.

5.3.1.4.11 Get_Channel_Measurement_Value.request

Function

This primitive is used by the IEEE 1900.6 SAP user to collect channel measurement values of the spectrum measurement module.

Semantics of the service primitive

```
Get_Channel_Measurement_Value.request
                                        (
                                        ChannelMeasurementID
                                        )
```

Parameters

Name	Type	Description
ChannelMeasurementID	Unsigned integer	ChannelMeasurementID uniquely defines the set of measurement values of the channel obtained by the spectrum measurement.

When used

This primitive is used by the IEEE 1900.6 SAP user when it needs to collect the channel measurement values of the spectrum measurement module.

Effect of receipt

The spectrum measurement module subsequently uses a Get_Channel_Measurement_Value.response primitive to reflect the results of the request.

5.3.1.4.12 Get_Channel_Measurement_Value.response

Function

This primitive returns the results of the request to collect the channel measurement values of the spectrum measurement module.

Semantics of the service primitive

```
Get_Channel_Measurement_Value.response
                                        (
                                        ChannelMeasurementID,
                                        Status,
```

Bandwidth,
NoisePower,
SignalLevel,
LockStatus
)

Parameters

Name	Type	Description
ChannelMeasurementID	Unsigned integer	ChannelMeasurementID uniquely defines the set of measurement values of the channel obtained by the spectrum measurement module.
Status	Enumeration	Status of operation. 0: success 1: unspecified failure 2: rejection 3: authorization failure
Bandwidth	Array(Frequency)	The bandwidth parameter describes the estimated bandwidth based on spectrum measurement (cf. 6.3.11).
NoisePower	NoisePower	The NoisePower parameter describes the noise power over the measured channel bandwidth (cf. 6.3.35).
SignalLevel	Structured	The SignalLevel parameter designates the amplitude of measured signal power within the measured bandwidth (cf. 6.3.36).
LockStatus	Enumeration	The LockStatus indicates whether the entity (i.e., the sender of this primitive) has been locked by a client for exclusive service. 1: The status of the entity is locked. The entity is locked by a client for exclusive service. 0: 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.

When used

The primitive is used in response to the Get_Channel_Measurement_Value.request primitive.

Effect of receipt

Upon receipt, the IEEE 1900.6 SAP user collects the measurement values of the spectrum measurement module.

5.3.1.4.13 Get_RAT_ID_Value.request

Function

This primitive is used by the IEEE 1900.6 SAP user to collect RAT ID values of the spectrum measurement module.

Semantics of the service primitive

Get_RAT_ID_Value.request
(
RATIDValueRequestTransID
)

Parameters

Name	Type	Description
RATIDValueRequestTransID	Unsigned integer	RATIDValueRequestTransID uniquely defines one transaction of requesting the set of measurement values of the RAT obtained by the spectrum measurement module.

When used

This primitive is used by the IEEE 1900.6 SAP user when it needs to collect the RAT ID values from the spectrum measurement module.

Effect of receipt

The spectrum measurement module subsequently uses a Get_RAT_ID_Value.response primitive to reflect the results of the request.

5.3.1.4.14 Get_RAT_ID_Value.response

Function

This primitive returns the results of the request to collect the RAT ID values from the spectrum measurement module.

Semantics of the service primitive

```
Get_RAT_ID_Value.response
                                    (
                                    RATIDValueRequestTransID,
                                    Status,
                                    RATID,
                                    LockStatus
                                    )
```

Parameters

Name	Type	Description
RATIDValueRequestTransID	Unsigned integer	RATIDValueRequestTransID uniquely defines one transaction of requesting the set of measurement values of the RAT obtained by the spectrum measurement module.
Status	Enumeration	Status of operation. 0: success 1: unspecified failure 2: rejection 3: authorization failure
RATID	Enumeration	The RATID parameter uniquely identifies the existing standard RATs known by both spectrum sensors and their clients, for example, Global System for Mobile (GSM), etc. (cf. 6.3.42).
LockStatus	Enumeration	The LockStatus indicates whether the entity (i.e., the sender of this primitive) has been locked by a client for exclusive service. 1: The status of the entity is locked. The entity is locked by a client for exclusive service. 0: 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.

When used

The primitive is used in response to the Get_RAT_ID_Value.request primitive.

Effect of receipt

Upon receipt, the IEEE 1900.6 SAP user collects the RAT ID values from the spectrum measurement module.

5.3.1.4.15 Notify

Function

This primitive is used by the measurement module to notify a status change to the IEEE 1900.6 SAP user through the M-SAP.

Semantics of the service primitive

Notify
(
Type,
Status,
Reason
)

Parameters

Name	Type	Description
Type	String	The Type parameter specifies the type of the notification.
Status	Enumeration	Status of operation. 0: success 1: unspecified failure 2: rejection 3: authorization failure
Reason	String	The Reason parameter expresses the reason of the notification.

When used

This primitive is used by the spectrum measurement module to report a status change to the IEEE 1900.6 SAP user.

Effect of receipt

The IEEE 1900.6 SAP user is notified the status change of the spectrum measurement module.

5.3.2 Communication Service Access Point

The C-SAP is used for the exchange of sensing-related information (cf. 6.1) between Sensors and their clients. The client role can be taken by a Sensor, CE, or DA. It abstracts services of the communication by providing a set of generic primitives or method and mapping these primitives to transport protocols.

The IEEE 1900.6 SAP users obtain the following services from the C-SAP:

- Sensing-related information send service
- Sensing-related information receive service
- Information services

5.3.2.1 Sensing-related information send service

The sensing-related information send service provides a set of primitives or method through which the IEEE 1900.6 SAP users send sensing-related information to another IEEE 1900.6 SAP user utilizing their local C-SAP. Because both SAP users are collocated with their respective IEEE 1900.6 service (see the definition of IEEE 1900.6 service in 3.1), this is the general method to carry out the exchange of sensing-related information between IEEE 1900.6 client and server (see the definitions of IEEE 1900.6 client and IEEE 1900.6 server in 3.1), or between remote IEEE 1900.6 logical entities, respectively (see the definition of IEEE 1900.6 logical entity in 3.1).

The sensing-related information send service primitives are defined as part of the C-SAP and are described in the following table.

Primitives	Description
Sensing_Related_Information_Send	This primitive is used by the IEEE 1900.6 SAP user to send sensing-related information to another IEEE 1900.6 SAP user.

5.3.2.1.1 Sensing_Related_Information_Send.request

Function

This primitive is used by the IEEE 1900.6 SAP user to send sensing-related information through its C-SAP to another IEEE 1900.6 SAP user that has C-SAP.

Semantics of the service primitive

```
Sensing_Related_Information_Send.request
        (
        InfoSource,
        InfoDestination,
        Route,
        ReportMode,
        SecLevel,
        ReportRate,
        SensingRelatedInformation
        )
```

Parameters

Name	Type	Description
InfoSource	String	InfoSource gives an ID that defines one IEEE 1900.6 logical entity as the source of sensing-related information (cf. 6.3.22 and 6.3.30).
InfoDestination	String	InfoDestination gives an ID that uniquely defines one IEEE 1900.6 logical entity as the destination of sensing-related information (cf. 6.3.22 and 6.3.30).
Route	Array(String)	Route gives a list of IDs of sensors and clients through which the sensing-related information flows from remote sensors to the client (cf. 6.3.17, 6.3.22, and 6.3.30).
ReportMode	Enumeration	The ReportMode parameter indicates either hard information or soft information (cf. 6.3.15).
SecLevel	Unsigned integer	The Seclevel parameter indicates the level of security depending on the available levels of security for authentication and data verification (cf. 6.3.20).
ReportRate	Structured	The ReportRate parameter specifies how often the measurement result should be reported (e.g., every 2 s or every 100 ms) (cf. 6.3.14).
SensingRelatedInformation	Structured	SensingRelatedInformation describes the sensing-related information that is sent through the C-SAP. Primitives/commands can also be sent as a payload via this C-SAP (cf. Clause 6).

When used

This primitive is used by the IEEE 1900.6 SAP user when it needs to send sensing-related information to another IEEE 1900.6 SAP user through the C-SAP.

Effect of receipt

The communication subsystem subsequently uses a Sensing_Related_Information_Send.response primitive to reflect the results of the request.

5.3.2.1.2 Sensing_Related_Information_Send.response

Function

This primitive returns the results of the request to send sensing-related information through the C-SAP.

Semantics of the service primitive

```
Sensing_Related_Information_Send.response
                        (
                        InfoSource,
                        InfoDestination,
                        Route,
                        ReportMode,
                        SecLevel,
                        ReportRate,
                        Status,
                        Timeout,
                        LockStatus
                        )
```

Parameters

Name	Type	Description
InfoSource	String	InfoSource gives an ID that uniquely defines one IEEE 1900.6 logical entity as the source of sensing-related information (cf. 6.3.22 and 6.3.30).
InfoDestination	String	InfoDestination gives an ID that uniquely defines one IEEE 1900.6 logical entity as the destination of sensing-related information (cf. 6.3.22 and 6.3.30).
Route	Array(String)	Route gives a list of IDs of sensors and clients through which the sensing-related information flows from remote sensors to the client (cf. 6.3.17, 6.3.22, and 6.3.30).
ReportMode	Enumeration	The ReportMode parameter indicates either hard information or soft information (cf. 6.3.15).
SecLevel	Unsigned integer	The Seclevel parameter indicates the level of security depending on the available levels of security for authentication and data verification (cf. 6.3.20).
ReportRate	Structured	The ReportRate parameter specifies how often the measurement result should be reported (e.g., every 2 s or every 100 ms) (cf. 6.3.14).
Status	Enumeration	Status of operation. 0: success 1: unspecified failure 2: rejection 3: authorization failure
Timeout	Unsigned fixed-point	The Timeout value in second indicates the timeout of the link.
LockStatus	Enumeration	The LockStatus indicates whether the entity (i.e., the sender of this primitive) has been locked by a client for exclusive service. 1: The status of the entity is locked. The entity is locked by a client for exclusive service. 0: 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.

When used

The primitive is used in response to the Sensing_Related_Information_Send.request primitive.

Effect of receipt

Upon receipt, the IEEE 1900.6 SAP user obtains the results of sending sensing-related information by the communication subsystem.

5.3.2.2 Sensing-related information receive service

The sensing-related information receive service provides a set of primitives or method through which the IEEE 1900.6 SAP user receives sensing-related information from another IEEE 1900.6 SAP user through the C-SAP.

The primitives for receiving sensing-related information are defined as a part of the C-SAP and are described in the following table.

Primitives	Description
Sensing_Related_Information_Receive	This primitive is used by the IEEE 1900.6 SAP user to receive sensing-related information from another IEEE 1900.6 SAP user.

5.3.2.2.1 Sensing_Related_Information_Receive.request

Function

This primitive is used by the IEEE 1900.6 SAP user to receive sensing-related information through its C-SAP.

Semantics of the service primitive

```
Sensing_Related_Information_Receive.request
                                        (
                                        InfoSource,
                                        InfoDestination,
                                        Route,
                                        ReportMode,
                                        SecLevel,
                                        ReportRate
                                        )
```

Parameters

Name	Type	Description
InfoSource	String	InfoSource gives an ID that uniquely defines one IEEE 1900.6 logical entity as the source of sensing-related information (cf. 6.3.22 and 6.3.30).
InfoDestination	String	InfoDestination gives an ID that uniquely defines one IEEE 1900.6 logical entity as the destination of sensing-related information (cf. 6.3.22 and 6.3.30).
Route	Array(String)	Route gives a list of IDs of sensors and clients through which the sensing-related information flows from remote sensors to the client (cf. 6.3.17, 6.3.22, and 6.3.30).
ReportMode	Enumeration	The ReportMode parameter indicates either hard information or soft information (cf. 6.3.15).
SecLevel	Unsigned integer	The Seclevel parameter indicates the level of security depending on the available levels of security for authentication and data verification (cf. 6.3.20).
ReportRate	Structured	The ReportRate parameter specifies how often the measurement result should be reported (e.g., every 2 s or every 100 ms) (cf. 6.3.14).

When used

This primitive is used by the IEEE 1900.6 SAP user when it needs to receive sensing-related information from another IEEE 1900.6 SAP user through a C-SAP.

Effect of receipt

The communication subsystem subsequently uses a Sensing_Related_Information_Receive.response primitive to reflect the results of the request.

5.3.2.2.2 Sensing_Related_Information_Receive.response

Function

This primitive returns the results of the request to send sensing-related information through the C-SAP.

Semantics of the service primitive

```
Sensing_Related_Information_Receive.response
                    (
                    InfoSource,
                    InfoDestination,
                    Route,
                    ReportMode,
                    SecLevel,
                    ReportRate,
                    Status,
                    Timeout,
                    LockStatus
                    )
```

Parameters

Name	Type	Description
InfoSource	String	InfoSource gives an ID that uniquely defines one IEEE 1900.6 logical entity as the source of sensing-related information (cf. 6.3.22 and 6.3.30).
InfoDestination	String	InfoDestination gives an ID that uniquely defines one IEEE 1900.6 logical entity as the destination of sensing-related information (cf. 6.3.22 and 6.3.30).
Route	Array(String)	Route gives a list of IDs of sensors and clients through which the sensing-related information flows from remote sensors to the client (cf. 6.3.17, 6.3.22, and 6.3.30).
ReportMode	Enumeration	The ReportMode parameter indicates either hard information or soft information (cf. 6.3.15).
SecLevel	Unsigned integer	The Seclevel parameter indicates the level of security depending on the available levels of security for authentication and data verification (cf. 6.3.20).
ReportRate	Structured	The ReportRate parameter specifies how often the measurement result should be reported (e.g., every 2 s or every 100 ms) (cf. 6.3.14).
Status	Enumeration	Status of operation. 0: success 1: unspecified failure 2: rejection 3: authorization failure
Timeout	Unsigned fixed-point	The Timeout value in second indicates the timeout of the link.
LockStatus	Enumeration	The LockStatus indicates whether the entity (i.e., the sender of this primitive) has been locked by a client for exclusive service. 1: The status of the entity is locked. The entity is locked by a client for exclusive service. 0: 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.

When used

The primitive is used in response to the Sensing_Related_Information_Receive.request primitive.

Effect of receipt

Upon receipt, the IEEE 1900.6 SAP user who sends the request obtains the results of requesting to receive sensing-related information through the communication subsystem.

5.3.2.3 Information services

The sensing information services provide a set of primitives or method through which the IEEE 1900.6 SAP users obtain information such as IDs and capabilities of the communication subsystem.

The primitives of the information services are defined as a part of the C-SAP and are described in the following table.

Primitives	Description
Get_CommSubsys_Profile	This primitive is used by the IEEE 1900.6 SAP user to obtain the information related to the communication subsystem.
Notify	This primitive is used by the communication subsystem to notify a status change to the IEEE 1900.6 SAP user.

5.3.2.3.1 Get_CommSubsys_Profile.request

Function

This primitive is used by the IEEE 1900.6 SAP user to obtain the information related to the capabilities of a communication subsystem.

Semantics of the service primitive

Get_CommSubsys_Profile.request ()
No parameters are used in this primitive.

When used

This primitive is used by the IEEE 1900.6 SAP user when it needs to obtain the information related to the capabilities of the communication subsystem.

Effect of receipt

The communication subsystem subsequently uses a Get_CommSubsys_Profile.response primitive to reflect the results of the request.

5.3.2.3.2 Get_CommSubsys_Profile.response

Function

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

Semantics of the service primitive

```
Get_CommSubsys_Profile.response
                                (
                                Status,
                                Comm_Subsys_ID,
                                Comm_Subsys_Capability,
                                LockStatus
                                )
```

Parameters

Name	Type	Description
Status	Enumeration	Status of operation. 0: success 1: unspecified failure 2: rejection 3: authorization failure
Comm_Subsys_ID	Unsigned integer	Comm_Subsys_ID uniquely defines one compliant communication subsystem.
Comm_Subsys_Capability	String	Comm_Subsys_Capability describes the capability of the communication subsystem.
LockStatus	Enumeration	The LockStatus indicates whether the entity (i.e., the sender of this primitive) has been locked by a client for exclusive service. 1: The status of the entity is locked. The entity is locked by a client for exclusive service. 0: 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.

When used

The primitive is used in response to the Get_CommSubsys_Profile.request primitive.

Effect of receipt

Upon receipt, the IEEE 1900.6 SAP user sending the request obtains the profile of the communication subsystem.

5.3.2.3.3 Notify

Function

This primitive is used by the communication subsystem to notify a status change to the IEEE 1900.6 SAP user through the C-SAP.

Semantics of the service primitive

```
Notify
        (
        Type,
        Status,
        Reason
        )
```

Parameters

Name	Type	Description
Type	String	The Type parameter specifies the type of the notification.
Status	Enumeration	Status of operation. 0: success 1: unspecified failure 2: rejection 3: authorization failure
Reason	String	The Reason parameter expresses the reason of the notification.

When used

This primitive is used by the communication subsystem to report a status change to an IEEE SAP user.

Effect of receipt

The IEEE 1900.6 SAP user obtains the status of the communication subsystem.

5.3.3 Application service access point

The A-SAP is used by the control/application (cf. 5.1) to interact with IEEE 1900.6 service. It provides a set of primitives or method for IEEE 1900.6 SAP users to control spectrum sensing and obtain sensing-related information from spectrum sensing.

The IEEE 1900.6 SAP users obtain the following services from the A-SAP:

— Sensor discovery service
— Sensing-related information access service
— Management and configuration service
— Information services

5.3.3.1 Sensor discovery service

The sensor discovery service provides a set of primitives or method through which the control/application discovers available spectrum measurement modules. The service also provides a set of primitives or method through which the control/application discovers available communication subsystems to provide communication services for sensing-related information exchange with another IEEE 1900.6 logical entity that uses the communication subsystem. The communication subsystem provides services to IEEE 1900.6 logical entities through the C-SAP (refer to 5.1). The following information is discovered in this service:

— Sensor logical ID
— Communication subsystem ID

The primitives of the sensor discovery service are defined as a part of the A-SAP and are defined in the following table. They provide a means to obtain available sensors.

Primitives	Description
Sensor logical ID	
Get_ Sensor_Logical_ID	This primitive is used by the control/application to obtain the list of sensors identified by the sensor logical IDs (cf. 6.3.30).
Communication subsystem ID	
Get_CommSubsys_ID	This primitive is used by the control/application to obtain the list of communication subsystems identified by the communication subsystem IDs.

5.3.3.1.1 Get_ Sensor_Logical_ID.request

Function

This primitive is used by the control/application to obtain a list of available sensors.

Semantics of the service primitive

Get_Sensor_Logical_ID.request ()

No parameters are used in this primitive.

When used

This primitive is used by the control/application when it needs to discover available sensors.

Effect of receipt

Upon receipt, the IEEE 1900.6 service provider uses a Get_Sensor_Logical_ID.reponse primitive to reflect the results of the request sent by the control/application.

5.3.3.1.2 Get_Sensor_Logical_ID.response

Function

This primitive returns the results of the request to obtain the list of available sensors.

Semantics of the service primitive

```
Get_ Sensor_Logical_ID.response
                    (
                    Status,
                    ListOfSensors,
                    LockStatus
                    )
```

Parameters

Name	Type	Description
Status	Enumeration	Status of operation. 0: success 1: unspecified failure 2: rejection 3: authorization failure
ListOfSensors	Vector(String)	List of sensors identified by the sensor logical IDs (cf. 6.3.30).
LockStatus	Enumeration	The LockStatus indicates whether the entity (i.e., the sender of this primitive) has been locked by a client for exclusive service. 1: The status of the entity is locked. The entity is locked by a client for exclusive service. 0: 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.

When used

The primitive is used in response to the Get_Sensor_Logical_ID.request primitive.

Effect of receipt

Upon receipt, the control/application obtains the logical IDs of available sensors.

5.3.3.1.3 Get_CommSubsys_ID.request

Function

This primitive is used by the control/application to obtain a list of available communication subsystems.

Semantics of the service primitive

Get_CommSubsys_ID.request ()

No parameters are used in this primitive.

When used

This primitive is used by the control/application when it needs to discover the available communication subsystems.

Effect of receipt

Upon receipt, the IEEE 1900.6 service provider uses a Get_CommSubsys_ID.response primitive to reflect the results of the request.

5.3.3.1.4 Get_CommSubsys_ID.response

Function

This primitive returns the results of the request to obtain the list of available communication subsystems.

Semantics of the service primitive

```
Get_CommSubsys_ID.response
                              (
                              Status,
                              ListOfCommSubsys,
                              LockStatus
                              )
```

Parameters

Name	Type	Description
Status	Enumeration	Status of operation. 0: success 1: unspecified failure 2: rejection 3: authorization failure
ListOfCommSubsys	Vector(Unsigned integer)	List of available communication subsystem IDs.
LockStatus	Enumeration	The LockStatus indicates whether the entity (i.e., the sender of this primitive) has been locked by a client for exclusive service. 1: The status of the entity is locked. The entity is locked by a client for exclusive service. 0: 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.

When used

The primitive is used in response to the Get_CommSubsys_ID.request primitive.

Effect of receipt

Upon receipt, the control/application obtains the IDs of available communication subsystems.

5.3.3.2 Sensing-related information access services

The sensing-related information access services provide a set of primitives or method through which the control/application accesses or issues sensing-related information. By these services, the control/application performs the following:

- Read sensing-related information
- Write sensing-related information

The primitives of sensing-related information services are defined as part of the A-SAP and are defined in the following table.

Primitives	Description
Read sensing-related information	
Read_Sensing_Related_Info	This primitive is used by control/application to read sensing-related information.
Write sensing-related information	
Write_Sensing_Related_Info	This primitive is used by control/application to write sensing-related information.

5.3.3.2.1 Read_Sensing_Related_Info.request

Function

This primitive is used by the control/application to read sensing-related information.

Semantics of the service primitive

Read_Sensing_Related_Info.request
(
SensingInfoList,
ClientLogID,
ClientPriorityFlag,
SensorLogID,
SensorPriority
)

Parameters

Name	Type	Description
SensingInfoList	Vector(Unsigned integer)	SensingInfoList gives a list of parameter IDs of sensing-related information that the control/application wants to read (cf. Table 2).
ClientLogID	Unsigned integer	ClientLogID uniquely specifies an IEEE 1900.6 client (cf. 6.3.22).
ClientPriorityFlag	Unsigned integer	The ClientPriorityFlag parameter indicates a priority level flag. Such priority level flag is assigned based on the application for which the client exploits the sensing-related information (cf. 6.3.18).
SensorLogID	Unsigned integer	Unique logical identification of a sensor within a certain area (cf. 6.3.30).
SensorPriority	Vector(String)	Sensor priority indicates the list of selected sensors according to priority ranking (cf. 6.3.19).

When used

This primitive is used by the control/application when it needs to read sensing-related information from one or more IEEE 1900.6 logical entities.

Effect of receipt

Upon receipt, the IEEE 1900.6 service provider uses a Read_Sensing_Related_Info.response primitive to reflect the results of the request.

5.3.3.2.2 Read_Sensing_Related_Info.response

Function

This primitive returns the results of the request to obtain the sensing-related information issued by the control/application.

Semantics of the service primitive

```
Read_Sensing_Related_Info.response
                                    (
                                    Status,
                                    Information,
                                    LockStatus
                                    )
```

Parameters

Name	Type	Description
Status	Enumeration	Status of operation. 0: success 1: unspecified failure 2: rejection 3: authorization failure
Information	Structured	Information describes the sensing-related information that is read by the control/application (cf. Table 2).
LockStatus	Enumeration	The LockStatus indicates whether the entity (i.e., the sender of this primitive) has been locked by a client for exclusive service. 1: The status of the entity is locked. The entity is locked by a client for exclusive service. 0: 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.

When used

The primitive is used in response to the Read_Sensing_Related_Info.request primitive.

Effect of receipt

Upon receipt, the control/application obtains the desired sensing-related information.

5.3.3.2.3 Write_Sensing_Related_Info.request

Function

This primitive is used by the control/application to write sensing-related information.

Semantics of the service primitive

```
Write_Sensing_Related_Info.request
                                    (
                                    SensingInfoList,
                                    Information,
                                    ClientLogID,
                                    ClientPriorityFlag,
                                    SensorLogID,
                                    SensorPriority
                                    )
```

Parameters

Name	Type	Description
SensingInfoList	Vector(Unsigned integer)	SensingInfoList gives a list of parameter IDs of sensing-related information that the control/application wants to read (cf. Table 2).
Information	Structured	Information gives a list of sensing-related information that the control/application wants to write (cf. Table 2).
ClientLogID	Unsigned integer	ClientLogID uniquely specifies an IEEE 1900.6 client (cf. 6.3.22).
ClientPriorityFlag	Unsigned integer	The ClientPriorityFlag parameter indicates a priority level flag. Such priority level flag is assigned based on the application for which the client exploits the sensing-related information (cf. 6.3.18).
SensorLogID	Unsigned integer	Unique logical identification of a sensor within a certain area (cf. 6.3.30).
SensorPriority	Vector(String)	Sensor priority indicates the list of selected sensors according to priority ranking (cf. 6.3.19).

When used

This primitive is used by the control/application when it needs to write sensing-related information to one or more IEEE 1900.6 logical entities.

Effect of receipt

Upon receipt, the IEEE 1900.6 service provider uses a Write_Sensing_Related_Info.response primitive to reflect the results of the request.

5.3.3.2.4 Write_Sensing_Related_Info.response

Function

This primitive returns the results of the request to write the sensing-related information issued by the control/application.

Semantics of the service primitive

Write_Sensing_Related_Info.response
(
Status,
LockStatus
)

Parameters

Name	Type	Description
Status	Enumeration	Status of operation. 0: success 1: unspecified failure 2: rejection 3: authorization failure
LockStatus	Enumeration	The LockStatus indicates whether the entity (i.e., the sender of this primitive) has been locked by a client for exclusive service. 1: The status of the entity is locked. The entity is locked by a client for exclusive service. 0: 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.

When used

The primitive is used in response to the Write_Sensing_Related_Info.request primitive.

Effect of receipt

Upon receipt, the control/application obtains the results of the request to write sensing-related information to one or more IEEE 1900.6 logical entities.

5.3.3.3 Management and configuration services

The management and configuration services provide a set of primitives or method through which the control/application manages IEEE 1900.6 logical entities and configures communication among IEEE 1900.6 logical entities. Through these services, the control/application performs the following:

- IEEE 1900.6 logical entity management
- Communication configuration

The primitives of management and configuration services are defined as a part of the A-SAP and are listed in the following table.

Primitives	Description
IEEE 1900.6 logical entity management	
Lock	This primitive is used by the control/application to lock IEEE 1900.6 logical entities and communication subsystems to prevent other controls/applications from accessing those resources.
Unlock	This primitive is used by the control/application to unlock IEEE 1900.6 logical entities and communication subsystems so that other controls/applications can access those resources.
BreakLock	This primitive is used by the control/application to break the lock so that other controls/applications can access those resources.
Trigger	This primitive is used by the control/application to trigger a specific action.
Communication configuration	
Comm_Manage	This primitive is used by the control/application to manage communications of IEEE 1900.6 logical entities.

5.3.3.3.1 Lock.request

Function

This primitive is used by the control/application to lock IEEE 1900.6 logical entities or communication subsystems for exclusive use and to prevent other controls/applications from accessing those resources.

Semantics of the service primitive

```
Lock.request
                (
                EntityID,
                CommSubsysID
                )
```

Parameters

Name	Type	Description
EntityID	String	EntityID uniquely identifies an IEEE 1900.6 logical entity. This EntityID can be client logical ID (cf. 6.3.22) or sensor logical ID (cf. 6.3.30).
CommSubsysID	Unsigned integer	CommSubsysID uniquely identifies a communication subsystem.

When used

This primitive is used by the control/application when it needs to lock IEEE 1900.6 logical entities or communication subsystems for exclusive use.

Effect of receipt

Upon receipt, the IEEE 1900.6 service uses a Lock.response to reflect the results of the request.

5.3.3.3.2 Lock.response

Function

This primitive returns the results of the request to lock IEEE 1900.6 logical entities or communication subsystems.

Semantics of the service primitive

```
Lock.response
                (
                Status,
                EntityID,
                CommSubsysID
                )
```

Parameters

Name	Type	Description
Status	Enumeration	Status of operation. 0: success 1: unspecified failure 2: rejection 3: authorization failure
EntityID	String	EntityID uniquely identifies an IEEE 1900.6 logical entity. This EntityID can be client logical ID (cf. 6.3.22) or sensor logical ID (cf. 6.3.30).
CommSubsysID	Unsigned integer	CommSubsysID uniquely identifies a communication subsystem.

When used

The primitive is used in response to the Lock.request primitive.

Effect of receipt

Upon receipt, the control/application obtains the results of the request to lock IEEE 1900.6 logical entities or communication subsystems.

5.3.3.3.2.1.1 Unlock

5.3.3.3.3 Unlock.request

Function

This primitive is used by the control/application to unlock IEEE 1900.6 logical entities or communication subsystems from exclusive use.

Semantics of the service primitive

Unlock.request
(
EntityID,
CommSubsysID
)

Parameters

Name	Type	Description
EntityID	String	EntityID uniquely identifies an IEEE 1900.6 logical entity. This EntityID can be client logical ID (cf. 6.3.22) or sensor logical ID (cf. 6.3.30).
CommSubsysID	Unsigned integer	CommSubsysID uniquely identifies a communication subsystem.

When used

This primitive is used by the control/application when it needs to unlock the IEEE 1900.6 logical entities or communication subsystems from exclusive use.

Effect of receipt

Upon receipt, the IEEE 1900.6 service uses an Unlock.response primitive to reflect the results of the request.

5.3.3.3.4 Unlock.response

Function

This primitive returns the results of the request to unlock IEEE 1900.6 logical entities or communication subsystems.

Semantics of the service primitive

Unlock.response
(
Status,
EntityID,
CommSubsysID
)

Parameters

Name	Type	Description
Status	Enumeration	Status of operation. 0: success 1: unspecified failure 2: rejection 3: authorization failure
EntityID	String	EntityID uniquely identifies an IEEE 1900.6 logical entity. This EntityID can be client logical ID (cf. 6.3.22) or sensor logical ID (cf. 6.3.30).
CommSubsysID	Unsigned integer	CommSubsysID uniquely identify a communication subsystem at an IEEE 1900.6 logical entity.

When used

The primitive is used in response to the Unlock.request primitive.

Effect of receipt

Upon receipt, the control/application can obtain the results of the request to unlock IEEE 1900.6 logical entities or communication subsystems.

5.3.3.3.5 BreakLock.request

Function

This primitive is used by the control/application to break the lock of IEEE 1900.6 logical entities or communication subsystems so that it can access those resources.

Semantics of the service primitive

```
BreakLock.request
                        (
                        EntityID,
                        CommSubsysID
                        )
```

Parameters

Name	Type	Description
EntityID	String	EntityID uniquely identifies an IEEE 1900.6 logical entity. This EntityID can be client logical ID (cf. 6.3.22 or sensor logical ID (cf. 6.3.30).
CommSubsysID	Unsigned integer	CommSubsysID uniquely identifies a communication subsystem.

When used

This primitive is used by the control/application when it needs to override a lock set on IEEE 1900.6 logical entities or communication subsystems so that it can use those resources.

Effect of receipt

Upon receipt, the IEEE 1900.6 service uses a BreakLock.response primitive to reflect the results of the request.

5.3.3.3.6 BreakLock.response

Function

This primitive returns the results of the request to override the lock of IEEE 1900.6 logical entities or communication subsystems.

Semantics of the service primitive

Unlock.response
(
Status,
EntityID,
CommSubsysID
)

Parameters

Name	Type	Description
Status	Enumeration	Status of operation. 0: success 1: unspecified failure 2: rejection 3: authorization failure
EntityID	String	EntityID uniquely identifies an IEEE 1900.6 logical entity. This EntityID can be client logical ID (cf. 6.3.22) or sensor logical ID (cf. 6.3.30).
CommSubsysID	Unsigned integer	CommSubsysID uniquely identifies a communication subsystem.

When used

The primitive is used in response to the BreakLock.request.primitive.

Effect of receipt

Upon receipt, the control/application obtains the results of the request to override the lock set on IEEE 1900.6 logical entities or communication subsystems.

5.3.3.3.7 Trigger.request

Function

This primitive is used by the control/application to trigger an event.

Semantics of the service primitive

Trigger.request
(
EventID,
TriggerTime,
Timeout
)

Parameters

Name	Type	Description
EventID	Unsigned integer	EventID uniquely identifies a particular triggered event.
TriggerTime	Unsigned integer	TriggerTime specifies when to begin operation.
Timeout	Unsigned fixed-point	Timeout in second is the maximum time to wait for a time-out error.

When used

This primitive is used by the control/application when it needs to trigger a particular event.

Effect of receipt

Upon receipt, the IEEE 1900.6 service uses a Trigger.response primitive to reflect the results of the request.

5.3.3.3.8 Trigger.response

Function

This primitive is used by the control/application to return the result of trigger.

Semantics of the service primitive

Trigger.response
(
EventID,
Status,
LockStatus
)

Parameters

Name	Type	Description
EventID	Unsigned integer	EventID uniquely identifies a particular triggered event.
Status	Enumeration	Status of operation. 0: success 1: unspecified failure 2: rejection 3: authorization failure
LockStatus	Enumeration	The LockStatus indicates whether the entity (i.e., the sender of this primitive) has been locked by a client for exclusive service. 1: The status of the entity is locked. The entity is locked by a client for exclusive service. 0: 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.

When used

This primitive is used by IEEE 1900.6 service in response to the Trigger.request primitive.

Effect of receipt

The control/application obtains the result of the trigger request.

5.3.3.3.9 Comm_Management.request

Function

This primitive is used by the control/application to manage communications among IEEE 1900.6 logical entities.

Semantics of the service primitive

```
Comm_Management.request
                          (
                          Status,
                          CommManagementTransID
                          )
```

Parameters

Name	Type	Description
Status	Enumeration	Status of operation. 0: success 1: unspecified failure 2: rejection 3: authorization failure
CommManagementTransID	Unsigned integer	CommManagementTransID uniquely identifies one transaction of a communication management.

When used

This primitive is used by the control/application when it needs to obtain the communication type IEEE 1900.6 logical entities.

Effect of receipt

Upon receipt, the IEEE 1900.6 service uses a Comm_Management.response primitive to reflect the results of the request.

5.3.3.3.10 Comm_Management.response

Function

This primitive returns the results of the communication management request issued by the control/application.

Semantics of the service primitive

Comm_Management.response
(
Status,
CommManagementTransID,
NetworkTopology,
LockStatus
)

Parameters

Name	Type	Description
Status	Enumeration	Status of operation. 0: success 1: unspecified failure 2: rejection 3: authorization failure
CommManagementTransID	Unsigned integer	CommManagementTransID uniquely identifies one transaction of a communication management.
NetworkTopology	String	NetworkTopology uniquely describes the communication type of IEEE 1900.6 logical entities, such as P2P, star, etc.
LockStatus	Enumeration	The LockStatus indicates whether the entity (i.e., the sender of this primitive) has been locked by a client for exclusive service. 1: The status of the entity is locked. The entity is locked by a client for exclusive service. 0: 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.

When used

The primitive is used in response to the Comm_Management.request primitive.

Effect of receipt

Upon receipt, the control/application obtains the results of the request.

5.3.3.4 Information services

The sensing information service provides a set of primitives or method through which the control/application obtains information such as ID and capability of the IEEE 1900.6 clients through the A-SAP.

The primitives of the information services are defined as a part of the A-SAP and are described in the following table.

Primitives	Description
Get_Client_Profile	This primitive is used by the control/application to obtain the information related to IEEE 1900.6 clients.
Notify	This primitive is used by the control/application to report its status.

5.3.3.4.1 Get_ Client_Profile.request

Function

This primitive is used by the control/application to obtain the profile such as ID and capability of IEEE 1900.6 clients.

Semantics of the service primitive

Get_Client_Profile.request ()

No parameters are used in this primitive.

When used

This primitive is used by the control/application when it needs to obtain the profile of the IEEE 1900.6 clients.

Effect of receipt

The IEEE 1900.6 service uses a Get_AppContr_Profile.response primitive to reflect the results of the request.

5.3.3.4.2 Get_Client_Profile.response

Function

This primitive returns the results of the request to get IEEE 1900.6 client profile.

Semantics of the service primitive

Get_ Client_ Profile.response

(
Status,

ClientLogID,
Client_Capability,
LockStatus
)

Parameters

Name	Type	Description
Status	Enumeration	Status of operation. 0: success 1: unspecified failure 2: rejection 3: authorization failure
ClientLogID	String	ClientLogID uniquely defines one IEEE 1900.6 client (cf. 6.3.22).
Client_Capability	String	Client_Capability describes the capability of IEEE 1900.6 clients.
LockStatus	Enumeration	The LockStatus indicates whether the entity (i.e., the sender of this primitive) has been locked by a client for exclusive service. 1: The status of the entity is locked. The entity is locked by a client for exclusive service. 0: 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.

When used

The primitive is used in response to the Get_AppContr_Profile.request primitive.

Effect of receipt

Upon receipt, the control/application obtains the profile of IEEE 1900.6 logical entities.

5.3.3.4.3 Notify

Function

This primitive is used by the control/application to notify a status change to IEEE 1900.6 service.

Semantics of the service primitive

Notify
(
Type,
Status,
Reason
)

Parameters

Name	Type	Description
Type	String	The Type parameter specifies the type of the notification.
Status	Enumeration	Status of operation. 0: success 1: unspecified failure 2: rejection 3: authorization failure
Reason	String	The Reason parameter expresses the reason of the notification.

When used

This primitive is used by the control/application to report a status change to IEEE service.

Effect of receipt

The IEEE 1900.6 service is notified the status change of the control/application.

6. Information description

This clause presents the structure of information exchanged between 1900.6 clients and sensors or, more generic, between providers and users of sensing-related information. First, 6.2 elaborates on the main categories of sensing-related information (i.e., sensing information, sensing control information, sensor information, and requirements derived from regulation). Additionally, the structure of control commands is addressed. Subclause 6.2 elaborates on data types and structures required in formally describing the parameters exchanged. Subclause 6.3 then formally describes these parameters exchanged based on the data type descriptions given by 6.2. Finally, 6.4 provides an object model for all sensing-related parameters and commands exchange between client and sensor.

6.1 Information categories

This subclause addresses the main categories of sensing-related information exchanged between sensors and their clients. Sensing-related information comprises sensing information, sensing control information, control commands, sensor information, and requirements derived from regulation.

6.1.1 Sensing information

Sensing information may include information about frequency band, energy, channel condition, time stamp of sensing, and local detection results, etc. In addition, related meta-information (i.e., the information required for describing, understanding, and evaluating information) is also considered sensing information. Meta-information may be both a priori knowledge or may be obtained from sensors. For example, the update rate of a sensing parameter is considered meta-information because it provides information on the temporal resolution of another sensing parameter. Consequently, relative positioning of sensors is providing the necessary information on the spatial resolution of the sensing information obtained without communicating sensor positions along with every sensing information report.

6.1.2 Sensing control information

Sensing control information may include information about the data to sense, the nature of the data (whether it should be raw data, not yet processed data obtained from a measurement, preprocessed data, relayed data, or fused data), the expected target performance, sensing duration, start/stop time of sensing activity, synchronization control information, power control, priority control, etc.

6.1.3 Control commands

Control commands are exchanged between IEEE 1900.6 logical entities to obtain information about a logical entity or about an ongoing sensing or processing activity. Additionally, control commands are used to set parameters for a sensing activity and to configure the logical entity. In general, two types of control commands are available: (1) standard control commands and (2) proprietary (manufacturer-specific) control commands.

— Standard control commands

  Standard control commands are defined by this document. They may include commands to read the status of a sensor, to request information about sensing capabilities, to request sensing information, to identify the sensor information source, etc.

— Proprietary control commands

  Proprietary control commands can be included for manufacturer or device specific purposes such as sensor calibration or sensor diagnostics. Proprietary control commands are not specified by this standard.

6.1.3.1 Structure of control commands

A control command should follow a predefined structure in such a way that it is understood by any entities implementing IEEE 1900.6 logical interfaces.

Command class	Command function	Command version

Command class	Command function	Command version	Command parameter

Figure 10 —Structure of control commands without (top) and with (bottom) control parameters

As shown in Figure 10, a control command consists of the command class, command function, and command version fields.

The command class is an identifier of a group of command functions. Command groups are either associated with a certain 1900.6 logical entity, are generic for all entities, or are manufacturer specific. The latter group shall use a specific command structure so that commands of this group are only recognized within the scope set by the device manufacturer.

The command function unambiguously defines the purpose and scope of validity of a command as well as the subsequent command structure (i.e., if it requires additional command parameters).

The command version indicates the version of the command and the command structure following the command class, function, and version fields. This field is used to provide backward compatibility for further releases of this standard.

Some control commands might be accompanied by parameters to narrow the required action. For example, starting a sensing activity might require setting start time, start frequency, scanning bandwidth, frequency increment, etc. to specify completely the action requested. Requesting the activity status of a sensor (i.e., if it is busy or idle) may not need any command parameter. Vice versa commands, such as TimeSync (cf. 6.3.23) and Scan (cf. 6.3.24), do require a certain number of parameters.

The structure of the command parameter field is specific for each command and depends on the command class, function, and version. It may consist of zero, one, or more sensing control parameters.

Chaining of control commands (Figure 11) shall be supported for communication efficiency and for implementing atomic control commands with and without command parameters. The command structure given by Figure 10 shall be preserved in the process of chaining commands.

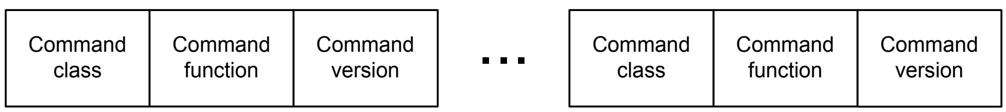


Figure 11 —Control command chaining

6.1.3.2 Control command parameters

In general, information exchange between IEEE 1900.6 logical entities is following a request/response scheme:

- A client requests a sensing parameter and a sensor responds by providing the parameter value.
- A client requests to set a parameter to a new value and a sensor (or DA) responds by providing the set value.

More complex schemes can be derived from this basic behavior as addressed by the description of service primitives in 5.3.

Control command parameters thus consist of (a vector of) zero, one, or more sensing control parameter IDs and, depending on the control command issued, the corresponding sensing control parameter value in a sequence as shown in Figure 12.

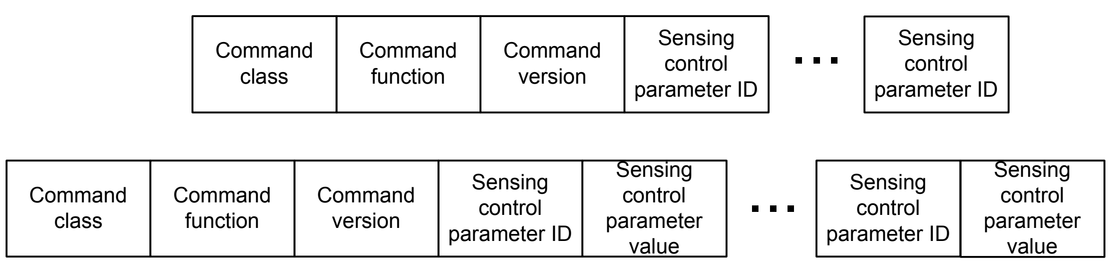


Figure 12 —Control command parameter structure without including parameter values (top) and with parameter values (bottom)

Herein, the sensing control parameter ID is considered a tag value that unambiguously identifies the parameter requested (or provided) and the parameter value. Thus, any sensing parameter and any sensing control parameter need to have their own unique identifiers as defined in the parameter description in 6.3.

This scheme is also valid for the exchange of sensing information. The distinction between sensing parameters and sensing control parameters is not necessary in the scope of information exchange by control commands. For example, setting a sensing parameter can be understood as a control parameter. When the entity is responding by confirming the action and by providing back the updated sensing parameter, the same parameter can be considered a sensing parameter.

6.1.4 Sensor information

Sensor information in this document is related to information about sensors and properties of sensors (i.e., sensor specification, sensor capabilities, or sensor identity). Sensor information, thus, is closely related to meta-information in the context of sensing information. Knowledge about a sensor's properties enables the correct understanding of sensing information obtained from this sensor.

Sensor properties may include information about the types of sensing techniques that can be performed by the sensor, measurement range, data accuracy, calibration information, analog-to-digital (A/D) and digital-to-analog (D/A) resolution, communication channel utilized, sensor's address, sensor's ID (logical or manufacturer ID), battery wear level, sensing cost, etc.

The term "sensor information" has been chosen here to emphasize the focus set to describe the sensor's attributes. The basic principle applies to other IEEE 1900.6 logical entities as well.

6.1.5 Requirements derived from regulation

National and international regulations specify spectrum allocation and spectrum usage in each country. Regulations may specify the operating requirements of devices, and this category of sensing-related information includes definitions of the sensing range, accuracy requirements, granularity, required measurement bandwidth, repeat frequency, and detection sensitivity. Furthermore, it may include information about sensing permission and synchronization. Although types of requirements are mostly identical, the range of the parameters may vary depending on the system, region, or country of deployment. The detailed fields, data types, and parameter ranges to meet the requirements from national regulatory bodies are defined in 6.3.

6.2 Data types

This subclause defines the primitive data types, simple data types, and derived data types used in the formal definition of IEEE 1900.6 parameters defined in 6.3. The physical units used in this definition are based on the International System of Units (SI) and are summarized in Table 1.

Table 1—Units used in the description of types and parameters

Unit	Unit symbol	Value	Note
second	s		SI unit
meter	m		SI unit
Hertz	Hz	1 Hz = 1 / s	SI derived unit
radian	rad	1 rad = 180 ⁄ π	SI derived unit, dimensionless
Watt	W	$kg \cdot m^2/s^3$	SI derived unit
degree of arc	°	1° = π ⁄ 180 rad	dimensionless
power ratio	dBm	$[dBm] = 10 \times \log_{10}([W]/1\ mW)$	dimensionless

6.2.1 Primitive and simple data types

This subclause summarizes primitive and simple data types (i.e., scalars and arrays made up of a single base type).

6.2.1.1 Boolean

A primitive logical data type having one of two values of "true" (1, nonzero) or "false" (0, zero).

Example: Indication that a signal level is above a certain threshold (cf. 6.3.36).

6.2.1.2 Integer

A primitive integral data type representing natural numbers and their negatives. Note that common binary representations limit the number range due to machine word length restrictions. In this standard, the integer length is defined as 32 bits.

Example: Indication of priority level of an IEEE 1900.6 client (cf. 6.3.18).

6.2.1.3 Unsigned integer

A primitive data type representing non-negative integrals.

Example: Priority level of a sensor in relation to a given default priority (cf. 6.3.19).

6.2.1.4 Float

A primitive data type storing real numbers, usually as floating-point numbers. Floating-point number representations as defined in IEEE Std 754™-2008 [B3] can be taken as an example.

Example: Indication of the amplitude of a signal (cf. 6.3.41).

6.2.1.5 String

A simple data type storing a sequence of data values, usually bytes or characters. A string is a special use of a one-dimensional array or vector.

Example: Logical identification of the 1900.6 client (cf. 6.3.22).

6.2.1.6 Vector

A simple data type storing a sequence of data values of a specified type. The notation vector(type) is used to specify the type of vector elements. A vector is a one-dimensional array.

Example: A list of channel numbers (cf. 6.3.13).

6.2.1.7 Array

A simple data type storing a collection of data values of a specified type. The notation array(type) is used to specify the type of array elements.

Example: Indication of different paths for the flow of sensing-related information (cf. 6.3.17).

6.2.2 Complex and derived data types

This subclause summarizes complex and derived data types such as structured types or types that rely on specific interpretation or restriction of the underlying primitive or simple type.

6.2.2.1 Enumeration

An enumeration is a listing of elements of a set in a way that maps to an index set consisting of natural numbers. That is, each element of the set is unambiguously represented by an ordinal.

Example: One out of a well-defined selection of possible configurations of a sensor (cf. 6.3.15).

6.2.2.2 Fixed-point

Fixed-point numbers are rational numbers with a fixed length mantissa and a fixed exponent. In contrast to a floating-point representation utilizing a fixed length but variable exponent, the value range is limited by the mantissa length, but the resolution is constant over the value range. They can be realized by using an integer value in conjunction with an implicit multiplier.

Example: A single relative time value in µs (cf. 6.3.4)

6.2.2.3 Unsigned fixed-point

Unsigned fixed-point numbers represent non-negative fixed-point numbers.

Example: A single frequency value in Hz (cf. 6.3.1).

6.2.2.4 Structured

A complex data type that aggregates a fixed set of labeled elements, possibly of different primitive or simple types, into a single element.

Example: The geographical location of a sensor (cf. 6.3.6).

6.3 Description of sensing-related parameters

Parameter descriptions are given in a tabular form throughout 6.3.1 through 6.3.42. They consist of the parameter name and ID, a short textual description, and a type and size specification, if needed.

— Parameter name and ID

The parameter's name provides a unique identification of the parameter in human readable form, whereas the numerical ID is given to unambiguously identify the parameter in the process of information exchange between IEEE 1900.6 logical entities.

— Parameter type and size

Parameters are of one of the types defined by 6.2 (primitive, simple, complex, and derived types). Some parameters may be further restricted in their value range or magnitude. The size field of the parameter description is supplementary information that is either a fixed value, determined by the number of elements contained (subparameters), or variable if at least one of the elements contained is optional, or is of variable size itself. To avoid implementation-dependent specifications for parameters, the size of a parameter is always given in terms of the underlying type.

For parameters based on array types, the size is given as the number of elements stored in the array or as "variable" if the size of the array is unspecified. Note that variable size arrays demand for an implicit array length value in information exchange.

For structured types, the aggregated size depends on the implementation of the elements enclosed and thus is omitted in the parameter description. The implementation then will decide on the binary representation, encoding, and size in terms of bits or bytes and any tag or length values needed.

Table 2 provides a summary of sensing-related parameters and categorizes these into parameters for sensing, sensing control, sensor information, and requirements derived from regulation (see the definition of regulatory requirements in 3.1).

— Sensing information (see 6.1.1)

Sensing parameters indicate the measurement output at the spectrum sensor and other associated parameters that augment the measurement data. Some parameters may appear as a control parameter when issued by the client to configure the sensor, and they may also be represented as a sensing parameter when they are issued as measurement output by sensor. Bandwidth and time stamp are examples of such parameters.

— Sensing control information (see 6.1.2 and 6.1.3)

Sensing control information is used to optimize the spectrum sensing and the procedure to obtain sensing information. Sensing control parameters are generated by the IEEE 1900.6 client. These parameters shall be used to realize the following two major functions:

- Sensor configuration according to the demands of the application and to the measurement process.
- Topological configuration of multiple sensors according to the demands of information exchange between sensors and between sensors and client.

One or more of the sensing control parameters given by Table 2 can be sent to the sensor for the preceding two purposes at the same time.

— Sensor information (see 6.1.4)

To optimize the measurement request or to configure the sensor, an IEEE 1900.6 client may require information describing a sensor's capabilities. This information may be known to the client if the sensor's capabilities conform to a standard or if the information is available in a local database on the client. In any other case, the client may also request this information directly from the sensor. Depending on the application, selected sensor profile information can be requested.

— Requirements derived from regulation (see 6.1.5)

Regulation may require specification of sensing range, accuracy, granularity, measurement bandwidth, repeat frequency, detection sensitivity, sensing permission, and synchronization information. The values of these parameters may vary according to regulation depending on the region or country of deployment.

Table 2 categorizes the sensing parameters into each of the four sensing-related information categories described earlier. It indicates most of the parameters as allocated to more than one category. This is especially true for simple parameter descriptions because these are sufficiently generic to be used, for example, when exchanging sensing parameters as well as for sensing control.

Table 2—Summary and taxonomy of sensing-related parameters

ID	Parameter name	Subclause	Sensing	Sensing control	Sensor	Regulatory
001	Frequency	6.3.1	Yes	Yes	Yes	Yes
002	Second	6.3.2	Yes	Yes	Yes	Yes
003	Time reference (RSecond)	6.3.3	Yes	Yes	Yes	Yes
004	Microsecond	6.3.4	Yes	Yes	Yes	Yes
005	Angle	6.3.5	Yes	Yes	Yes	Yes
006	Reference geolocation (RGeolocation)	6.3.6	Yes	Yes	Yes	Yes
007	Power	6.3.7	Yes	Yes	Yes	Yes
008	Time stamp (TimeStamp)	6.3.8	Yes	Yes	Yes	Yes
009	Time duration (TimeDuration)	6.3.9	Yes	Yes	Yes	Yes
010	Channel list (ChList)	6.3.10	Yes	Yes	Yes	Yes
…						
101	Bandwidth	6.3.11	Yes	Yes		Yes
102	Total measurement duration (TotMeasuDur)	6.3.12	Yes	Yes		
103	Channel order (ChOrder)	6.3.13	Yes	Yes		
104	Reporting rate (ReportRate)	6.3.14	Yes	Yes		
105	Reporting mode (ReportMode)	6.3.15	Yes	Yes		
106	Performance metric (PerfMetric)	6.3.16	Yes	Yes		
107	Route	6.3.17		Yes		
108	ClientPriorityFlag	6.3.18		Yes		
109	SensorPriority	6.3.19		Yes	Yes	
110	Security level (SecLevel)	6.3.20			Yes	
111	DataKey	6.3.21	Yes	Yes		
112	ClientLogID	6.3.22		Yes		
…						
201	Time synchronization (TimeSync)	6.3.23		Yes		
202	Scan	6.3.24		Yes		
…						
301	Absolute sensor location (AbsSensorLocation)	6.3.25	Yes		Yes	
302	Relative sensor location (RelSensorLocation)	6.3.26	Yes		Yes	
303	Measurement range (MeasuRange)	6.3.27	Yes	Yes	Yes	
304	SensingMode	6.3.28	Yes	Yes	Yes	
305	SensorID	6.3.29		Yes	Yes	
306	Sensor logical ID (SensorLogID)	6.3.30		Yes	Yes	
307	BatteryStatus	6.3.31	Yes		Yes	
308	DataSheet	6.3.32			Yes	
309	ConfidenceLevel	6.3.33	Yes			
…						
401	Measurement bandwidth (MeasuBandwidth)	6.3.34	Yes			
402	NoisePower	6.3.35	Yes			
403	SignalLevel	6.3.36	Yes			
404	ModulationType (ModuType)	6.3.37	Yes			
405	TrafficPattern	6.3.38	Yes			
406	TrafficInformation	6.3.39	Yes			
407	SignalType	6.3.40	Yes			

ID	Parameter name	Subclause	Sensing	Sensing control	Sensor	Regulatory
408	SignalDesc	6.3.41	Yes			
409	RATID	6.3.42	Yes			

6.3.1 Frequency

Frequency denotes a single generic frequency parameter given in Hz. The parameter is realized as an unsigned fixed-point value assuming a resolution of 1 Hz and a maximum range of 0 to 2^{32}–1 Hz (i.e., the equivalent of a 32-bit mantissa). Derived parameters, relaxing or setting constraints to range and resolution, for example, may extend this definition and may modify the range, type, or physical unit fields accordingly.

Name:	**Frequency**	**Phys. unit**	Hz	**Extends:**	—	
ID:	001	**Size:**	1	**Type:**	Unsigned fixed-point	
Desc:	Basic unbounded frequency parameter.					
	Range (min/resolution/max):		0	1 Hz		$(2^{32}–1)$Hz

6.3.2 Second

Second denotes a single generic time parameter given in s. The parameter is realized as a signed integer value assuming a resolution of 1 s and a maximum range of $–2^{31}$ to $+2^{31}–1$ s (i.e., the equivalent of a 32-bit two's complement integer). This parameter can be used to realize time difference values. Derived parameters, relaxing or setting constraints to range and resolution, for example, may extend this definition and may modify the range, type, or physical unit fields accordingly.

Name:	**Second**	**Phys. unit**	s	**Extends:**	—	
ID:	002	**Size:**	1	**Type:**	Signed integer	
Desc:	Basic time value in seconds.					
	Range (min/resolution/max):		$–2^{31}$	1		$2^{31}–1$

6.3.3 Time reference (RSecond)

RSecond denotes a single time parameter given in s. It extends Second (cf. 6.3.2) to realize time difference values with respect to midnight (UTC) of January 1, 1970 not counting leap seconds. The parameter is realized as an unsigned integer assuming a maximum range of 0 to $2^{32}–1$ s (i.e., the equivalent of a 32-bit integer).

Name:	**RSecond**	**Phys. unit**	s	**Extends:**	Second	
ID:	003	**Size:**	1	**Type:**	Unsigned integer	
Desc:	Basic reference time value. Seconds since midnight (UTC) of January 1, 1970 absolute time.					
	Range (min/resolution/max):		0	1		$2^{32}–1$

6.3.4 Microsecond

Microsecond denotes a single generic time parameter given in µs. It can be used in conjunction with Second (cf. 6.3.2) or derived parameters to form a high-resolution time parameter. The parameter is realized as a fixed-point value assuming a resolution of 1 ns and a maximum range of $–10^6 + 10^{-3}$ to $10^6 – 10^{-3}$ µs. Derived parameters, relaxing or setting constraints to range and resolution, for example, may extend this definition and may modify the range, type, or physical unit fields accordingly.

Name:	**Microsecond**	**Phys. unit**	µs	**Extends:**	—	
ID:	004	**Size:**	1	**Type:**	Fixed-point	
Desc:	Basic time value in microseconds. Resolution is nanoseconds, maximum value is 1 s.					
	Range (min/resolution/max):		$–10^6 + 10^{-3}$	10^{-3}		$10^6 – 10^{-3}$

6.3.5 Angle

The Angle parameter is used to specify angular values such as longitude or latitude of a specific geolocation. It can also be used to specify antenna related parameters such as the elevation angle.

Name:	**Angle**	**Phys. Unit**	° (degree)	**Extends:**	—
ID:	005	**Size:**	1	**Type:**	Fixed-point
Desc:	Generic parameter describing an angular value or measurement.				
	Range (min/resolution/max):		–180	1 µ°	180

6.3.6 Reference geolocation (RGeolocation)

The RGeolocation parameter indicates an absolute geolocation based on the WGS 84 reference coordinate system or its successors (see National Imagery and Mapping Agency [B6]). It may serve as a reference location for relative positions to specify, for example, geographical areas.

Name:	**RGeolocation**	**Phys. Unit**	—	**Extends:**	—
ID:	006	**Size:**	3	**Type:**	Structured
Desc:	Basic reference geolocation parameter indicating the absolute geolocation of an IEEE 1900.6 logical entity based on the WGS 84 reference coordinate system or its successors.				
.0	**RGeolocation.elev**	**Type**	Signed integer	See NOTE	
.1	**RGeolocation.lat**	**Type**	Angle		
.2	**RGeolocation.long**	**Type**	Angle		
NOTE—Elevation is measured as altitude in m with respect to sea level.					

6.3.7 Power

Power denotes a single generic parameter given in dBm. It indicates a basic RF power. Derived parameters related to power, relaxing or setting constraints to range and resolution, for example, may extend this definition and may modify the range, type, or physical unit fields accordingly.

Name:	**Power**	**Phys. unit**	dBm	**Extends:**	—
ID:	007	**Size:**	1	**Type:**	Fixed-point
Desc:	Basic RF power value.				
	Range (min/resolution/max):		–160	10^{-1}	70

6.3.8 Time stamp (TimeStamp)

Name:	**TimeStamp**	**Phys. unit**	s, µs	**Extends:**	—
ID:	008	**Size:**	2	**Type:**	Structured
Desc:	Timestamp value. Seconds and microseconds since midnight (UTC) of January 1, 1970.				
.0	**Timestamp.seconds**		**Type:**	RSeconds	
.1	**Timestamp.usec**		**Type:**	Microseconds	

6.3.9 Time duration (TimeDuration)

Name:	**TimeDuration**	**Phys. unit**	s, µs	**Extends:**	—
ID:	009	**Size:**	2	**Type:**	Structured
Desc:	Time distance value in seconds and microseconds.				
.0	**TimeDuration.seconds**		**Type:**	Seconds	
.1	**TimeDuration.usec**		**Type:**	Microseconds	

6.3.10 Channel list (ChList)

The ChList specifies a list of frequency bands by providing a vector of channel numbers along with a corresponding vector of bandwidth parameters.

Name:	**ChList**	**Phys. unit**	—	**Extends:**	—
ID:	010	**Size:**	2	**Type:**	Structured
Desc:	Parameter that describes a set of frequency bands.				
.0	**ChList.channel**		**Type:**	Vector(Unsigned integer)	
.1	**ChList.band**		**Type:**	Vector(Bandwidth)	

6.3.11 Bandwidth

The Bandwidth parameter defines lower and upper frequencies (bandstop and bandstart). The starting frequency can be, for example, 30 MHz for military, 54 MHz for fixed TV White Space, 470 MHz for portable TV White Space, as required for cellular and WiMAX™,[8] and 2400 MHz for Unlicensed bands.[9] To specify fully the bandwidth, the bandstop frequency shall also be specified.

Name:	**Bandwidth**	**Phys. unit**	Hz	**Extends:**	—
ID:	101	**Size:**	2	**Type:**	Array(Frequency)
Desc:	Basic unbounded bandwidth parameter.				
.0	**Bandwidth.bandstart**		**Type:**	Frequency	
.1	**Bandwidth.bandstop**		**Type:**	Frequency	

6.3.12 Total measurement duration (TotMeasuDur)

The TotMeasuDur parameter is issued by the IEEE 1900.6 client to the sensor to specify the total duration of time required to measure the spectrum. The actual unit measurement period will be specified by the sensor based on the quality of sensing. Multiple measurements can be reported during the total measurement duration.

Name:	**TotMeasuDur**	**Phys. Unit**	s, µs	**Extends:**	—
ID:	102	**Size:**	2	**Type:**	Structured
Desc:	Total duration in seconds and microseconds.				
.0	**TotMeasuDur.seconds**		**Type:**	Seconds	
.1	**TotMeasuDur.usec**		**Type:**	Microseconds	

6.3.13 Channel order (ChOrder)

The ChOrder parameter specifies an index set. In order to utilize the information stored in ChOrder, the start and stop frequencies of each channel shall be known to both client and sensor. This can be achieved, for example, by providing a ChList parameter along with this parameter.

Name:	**ChOrder**	**Phys. unit**	—	**Extends:**	—
ID:	103	**Size:**	Variable	**Type:**	Vector(Unsigned integer)
Desc:	Parameter that indicates the sequence of channels to be scanned.				

6.3.14 Reporting rate (ReportRate)

The ReportRate parameter is issued by the IEEE 1900.6 client to specify how often the measurement result should be reported or updated (e.g., every 2 s or every 100 ms).

[8] WiMAX is a trademark in the U.S. Patent & Trademark Office, owned by the WiMAX Forum.
[9] This information is given for the convenience of users of this standard and does not constitute an endorsement by the IEEE of these products. Equivalent products may be used if they can be shown to lead to the same results.

Name:	**ReportRate**	**Phys. unit**	s, µs	**Extends:**	—
ID:	104	**Size:**	2	**Type:**	Structured
Desc:	Time distance value in seconds and microseconds between two consecutive measurement reports.				
.0	**ReportRate.seconds**		**Type:**	Seconds	
.1	**ReportRate.usec**		**Type:**	Microseconds	

6.3.15 Reporting mode (ReportMode)

The ReportMode parameter is issued by the IEEE 1900.6 client to indicate in what form the sensing information should be reported. Two reporting modes are anticipated, namely "hard information" and "soft information." The reporting mode can take a form of one bit hard decision information or soft information in a form of quantized value of signal strength.

Name:	**ReportMode**	**Phys. unit**	—	**Extends:**	—
ID:	105	**Size:**	Variable	**Type:**	Enumeration
Desc:	Reporting mode to indicate either hard information or soft information.				
Enumerator	**ReportMode.hard**	**Value:**	0	See NOTE 1	
Enumerator	**ReportMode.soft**	**Value:**	1	See NOTE 2	
NOTE 1—When hard information reporting mode is selected, sensors provide information that only indicates the spectral occupancy as vacant or occupied. NOTE 2—When soft information reporting mode is selected, sensors provide information that indicates the RF signal level (signal strength) in the band of interest.					

6.3.16 Performance metric (PerfMetric)

The PerfMetric parameter is issued by the IEEE 1900.6 client to the sensors to indicate the quality of sensing. In particular, this value is needed at the sensor side if local decision is carried. Sensors perform measurement configuration to obtain the desired performance metric as specified by the IEEE 1900.6 client.

Name:	**PerfMetric**	**Phys. unit**	%	**Extends:**	—
ID:	106	**Size:**	Variable[a]	**Type:**	Array(Unsigned integer)
Desc:	Parameter that indicates the quality of sensing.				
.0	**PerfMetric.Pd**	**Type:**	Unsigned integer	See NOTE 1	
.1	**PerfMetric.Pfa**	**Type:**	Unsigned integer	See NOTE 2	
NOTE 1—When the PerformanceMetric.pd is specified, sensors perform sensing by setting the rate of detection according to this value. Rate of detection is expressed as a percentage bounded between 0% and 100%. NOTE 2—When the PerformanceMetric.pfa is specified, sensors perform sensing by setting the rate of false alarm according this value. Rate of false alarm is expressed as a percentage bounded between 0% and 100%.					

[a]In some cases, only one of the information elements can be sufficient to describe the performance.

6.3.17 Route

The Route parameter carries information of the path in which the measurement data flows from remote sensors to the client. Routing information may indicate only one sensor to be used as an intermediate node between the remote sensors and the client where this sensor can act as a gateway between the client and multiple sensors. Data fusion at intermediate nodes or gateways can reduce the traffic load, can offload computational demands, and may reduce sensor transmit power toward the IEEE 1900.6 client, potentially reducing communication overhead and extending sensor battery life.

Name:	**Route**	**Phys. Unit**	—	**Extends:**	—	
ID:	107	**Size:**	Variable	**Type:**	Array(String)	
Desc:	Route command indicates the information flow path.					
.0	**Route.path1**		**Type:**	String		See NOTE 1
.1	**Route.path2**		**Type:**	String		See NOTE 2
NOTE 1—Route.path1 is the client address. NOTE 2—Route.path2 is the IEEE 1900.6 logical entity that may act as a gateway. Depending on the path length, additional addresses may be specified.						

6.3.18 ClientPriorityFlag

The ClientPriorityFlag parameter indicates a priority-level flag. Such a priority-level flag is assigned based on the application for which the client exploits the sensing information.

Name:	**ClientPriorityFlag**	**Phys. Unit**	—	**Extends:**	—	
ID:	108	**Size:**	1	**Type:**	Unsigned integer	See NOTE
Desc:	Client priority indicator where integer values represent different priority levels.					
NOTE—It is assumed that priority levels are defined and understood by both the client and the sensor.						

6.3.19 SensorPriority

The SensorPriority parameter indicates which sensors are given priority to report their measurement result. After the client obtains the necessary information about the potential sensing information sources, it will assign priority to each sensor. Only information sources indicated by the sensor priority will be valid to provide sensing information. This parameter is useful when implementing distributed sensing where sensors with an acceptable link quality are allowed to take part in the cooperation. Priority ranking may also be used to implement weighted combining.

Name:	**SensorPriority**	**Phys. Unit**	--	**Extends:**	--
ID:	109	**Size:**	Variable	**Type:**	Vector(String)
Desc:	Sensor priority indicates the list of selected sensors according to priority ranking				
NOTE—When the sensor priority flag is on, it contains a list of sensor's IDs according to a descending priority rank. This information is broadcasted back to the sensors to indicate which sensors are selected for the measurement report.					

6.3.20 Security level (SecLevel)

The Seclevel parameter indicates the level of security depending on the available levels of security for authentication and data verification. Once the security level preference is indicated by the application, the corresponding authentication procedure is carried out by the communication subsystem.

Name:	**SecLevel**	**Phys. Unit**	—	**Extends:**	—
ID:	110	**Size:**	1	**Type:**	Unsigned integer
Desc:	Security level indicator.				
NOTE—Depending on the application, a client may specify a high or low level of security from the available authentication mechanisms supported by both the client and the server.					

6.3.21 DataKey

The DataKey parameter is an information ID that identifies the type of database information stored in the DA. This ID is used by the client to filter the type of information needed from the DA. Database information identifiers (a set of IDs and corresponding database information type) should be specified and known by clients.

Name:	DataKey	Phys. Unit	—	Extends:	—
ID:	111	Size:	1	Type:	Unsigned integer
Desc:	Indicates the type of database information stored in the DA.				

6.3.22 ClientLogID

ClientLogID represents the unique logical identification of the IEEE 1900.6 client. It can be used to identify the control entity during a request for sensing information.

Name:	ClientLogID	Phys. unit	—	Extends:	—
ID:	112	Size:	1	Type:	String
Desc:	Unique logical identification of an IEEE 1900.6 client.				

6.3.23 Time synchronization (TimeSync)

The TimeSync command provides the system to be synchronized. The command described here is with a minimum parameter space. Unnecessary synchronization wastes resources, and some sensors may not have the resource to implement advanced protocols. Insufficient synchronization also leads to poor application performance. This command can be extended by incorporating more information elements depending on the choice of synchronization scheme.

Name:	TimeSync	Phys. Unit	—	Extends:	—
ID:	201	Size:	3	Type:	Structured
Desc:	Sync command for synchronization.				
.0	**TimeSync. ClockAddress (ClientLogID)**		**Type:**	---	
.1	**TimeSync.MaxError**		**Type:**	integer	See NOTE
.2	**TimeSync.RSecond**		**Type:**	RSecond	
NOTE—Max error can be defined as tolerable error where it is given by the ratio of time offset divided by reference time.					

6.3.24 Scan

The Scan command instructs the sensor to scan between a lower and a upper frequency bound with a given measurement bandwidth and temporal resolution. This command can be extended by strategies that are more complex.

Name:	Scan	Phys. Unit	—	Extends:	—
ID:	202	Size:	6	Type:	Structured
Desc:	Basic scan command.				
.0	**Scan.Range**		**Type:**	Bandwidth	
.1	**Scan.Stepbandwidth**		**Type:**	Bandwidth	
.2	**Scan.Resolution**		**Type:**	Frequency	
.3	**Scan.TimePerStep**		**Type:**	TimeDuration	
.4	**Scan.LowerThreshold**		**Type:**	NoisePower	
.5	**Scan.ChannelOrder**		**Type**	ChOrder	See NOTE
NOTE—ChannelOrder specifies the scan order according to a predefined vector of step indices (sequence vector should use a bounded type rather than the generic IntVector type). When this parameter is omitted or set to 0, the scan order will be assumed linear or sequential as the default scan order.					

6.3.25 Absolute sensor location (AbsSensorLocation)

The AbsSensorLocation parameter indicates the absolute location of the sensor.

Name:	AbsSensorLocation	Phys. unit	—	Extends:	—
ID:	301	Size:	1	Type:	RGeolocation
Desc:	The absolute sensor location parameter indicates the absolute location of the sensor.				

6.3.26 Relative sensor location (RelSensorLocation)

The RelSensorLocation parameter is used when the position of the sensor is measured by triangulation with respect to known reference positions.

Name:	**RelSensorLocation**	**Phys. unit**	—	**Extends:**	—
ID:	302	**Size:**	1	**Type:**	RGeolocation
Desc:	Relative sensor location parameter describes the location with respect to a known reference point.				

6.3.27 Measurement range (MeasuRange)

The MeasuRange parameter indicates the range of frequencies that can be measured according to the manufacturer specification.

Name:	**MeasuRange**	**Phys. unit**	—	**Extends:**	—
ID:	303	**Size:**	1	**Type:**	Bandwidth
Desc:	Range of frequency supported by the sensor according to the manufacturer specification.				

6.3.28 SensingMode

The SensingMode parameter indicates the numbers and types of standard sensing methods supported by the sensor. Note that the size of SensingMode is variable, which allows the user to choose between multiple sensing modes. These sensing techniques are indentified in Noguet et al. [B7] and Pucker [B8].

Name:	**SensingMode**	**Phys. unit**	--	**Extends:**	--
ID:	304	**Size:**	Variable	**Type:**	Enumeration
Desc:	List of sensing modes supported by the sensor.				
Enumerator		SensingMode.EnergyDetection		Value	0
Enumerator		SensingMode.MatchedFilter		Value	1
Enumerator		SensingMode.Cyclostationary		Value	2
Reserved					

6.3.29 SensorID

The SensorID parameter describes information about the sensor identity as assigned by the manufacturer. It contains both the vendor identification and product identification.

Name:	**SensorID**	**Phys. unit**	—	**Extends:**	—
ID:	305	**Size:**	2	**Type:**	Array(String)
Desc:	Unique identification of the sensor.				
.0	**SensorID.VendorID**		**Type:**	String	
.1	**SensorID.ProductID**		**Type:**	String	

6.3.30 Sensor logical ID (SensorLogID)

SensorLogID is a parameter indicating the logical ID of a sensor within a certain area.

Name:	**SensorLogID**	**Phys. unit**	—	**Extends:**	—
ID:	306	**Size:**	1	**Type:**	String
Desc:	Unique logical identification of a sensor within a certain area.				

6.3.31 BatteryStatus

When the sensor is battery operated, the BatteryStatus parameter indicates the remaining power.

Name:	**BatteryStatus**	**Phys. unit**	%	**Extends:**	—	
ID:	307	**Size:**	1	**Type:**	Unsigned integer	
Desc:	The remaining power in percentage.					
	Range (min/resolution/max):		0		1	100

6.3.32 DataSheet

The DataSheet parameter indicates standard manufacturer specifications, which are also known by the client device.

NOTE—The manufacturer ultimately defines the content provided by the datasheet parameter. Some of the subparameters thus might be void; that is, it is assumed that most/all of the parameters are optional. The primitive retrieves all subparameters given that the corresponding capability is provided by the sensor. Otherwise, the subparameter is omitted. The list of datasheet parameters given in the parameter description thus is the best case.

Name:	**DataSheet**	**Phys. unit**	—	**Extends:**	—
ID:	308	**Size:**	Variable	**Type:**	Structured
Desc:	List of datasheet items according to manufacturer specification.				
.0	**DataSheet.ADDAResolution**		**Type:**		Unsigned integer
.1	**DataSheet.AmplitudeSensitivity**		**Type:**		Float
.2	**DataSheet.AngleResolution**		**Type:**		Angle
.3	**DataSheet.AntennaBandwidth**		**Type:**		Frequency
.4	**DataSheet.AntennaBeamPointing**		**Type:**		Array(Angle)
.5	**DataSheet.AntennaBeamwidth**		**Type:**		Array(Angle)
.6	**DataSheet.AntennaDirectivityGain**		**Type:**		Float
.7	**DataSheet.AntennaGain**		**Type:**		Float
.8	**DataSheet.AntennaHeight**		**Type:**		Float
.9	**DataSheet.AntennaPolarization**		**Type:**		Enumeration
.10	**DataSheet.CalibrationData**		**Type:**		String
.11	**DataSheet.CalibrationMethod**		**Type:**		Unsigned integer
.12	**DataSheet.ChannelFiltering**		**Type:**		String
.13	**DataSheet.DynamicRange**		**Type:**		Float
.14	**DataSheet.FrequencyResolution**		**Type:**		Frequency
.15	**DataSheet.LocationTimeCapability**		**Type:**		String
.16	**DataSheet.LoggingFunctions**		**Type:**		String
.17	**DataSheet.NoiseFactor**		**Type:**		Float
.18	**DataSheet.PhaseNoise**		**Type:**		Float
.19	**DataSheet.PowerConsumption**		**Type:**		String
.20	**DataSheet.RecordingCapability**		**Type:**		String
.21	**DataSheet.SweepTime**		**Type:**		Time
.22	**Reserve**				

6.3.33 ConfidenceLevel

The ConfidenceLevel parameter indicates the degree of certainty that a statistical estimation of the spectrum measurement (e.g., frequency, amplitude, or phase) is accurate. A new confidence level can be computed whenever the current confidence level needs to be updated, for instance, when the sample size changes.

Name:	**ConfidenceLevel**	**Phys. unit**	—	**Extends:**	—
ID:	309	**Size:**	2	**Type:**	Structured
Desc:	The ConfidenceLevel parameter should be accompanied by the confidence interval in which the true value may be located with respect to the estimated value.				
.0	**ConfidenceLevel.Value**		**Type:**	Float	
.1	**ConfidenceLevel.Interval**		**Type:**	Array(Float)	

6.3.34 Measurement bandwidth (MeasuBandwidth)

The MeasuBandwidth parameter is the same as the bandwidth used by the control parameter, but when the value is returned by the sensor, it indicates the measurement bandwidth that has actually been used by the sensor.

Name:	**MeasuBandwidth**	**Phys. unit**	Hz	**Extends:**	--
ID:	401	**Size:**	2	**Type:**	Array(Frequency)
Desc:	Describes the bandwidth being used by the sensor.				
.0	**MeasuBandwidth.BandStart**		**Type:**	Frequency	
.1	**MeasuBandwidth.BandStop**		**Type:**	Frequency	

6.3.35 NoisePower

Noise power is thermal noise in accordance with the measurement bandwidth.

Name:	NoisePower	**Phys. unit**	dBm	**Extends:**	Power
ID:	402	**Size:**	1	**Type:**	Fixed point
Desc:	Basic RF noise power value.				
	Range (min/resolution/max):		–173	10^{-2}	0

6.3.36 SignalLevel

The SignalLevel parameter designates the amplitude of the measured signal power within a specific bandwidth.

Name:	**SignalLevel**	**Phys. unit**	—	**Extends:**	Power
ID:	403	**Size:**	Variable	**Type:**	Structured
Desc:	SignalLevel indicating the RF power of measured signal.				
.0	**SignalLevel.Soft**		Type	Power	
.1	**SignalLevel.Hard**		Type	Boolean	

6.3.37 Modulation type (ModuType)

The ModuType parameter specifies the type of modulation that the detected signal exhibits.

Name:	**ModuType**	**Phys. unit**	—	**Extends:**	—
ID:	404	**Size:**	1	**Type:**	Enumeration
Desc:	List of known modulation types for RF signal characterization.				
Enumerator	**ModuType.Unknown**		**Value:**	0	
Enumerator	**ModuType.Analog**		**Value:**	1	
Enumerator	**ModuType.Digital**		**Value:**	2	

6.3.38 TrafficPattern

The TrafficPattern parameter classifies the detected signal in temporal domain as periodic burst, random burst, continuous, or unknown. This parameter can be helpful in identifying a signal by comparing its content with known traffic patterns of existing standard air interfaces.

Name:	**TrafficPattern**	**Phys. unit**	—	**Extends:**	—
ID:	405	**Size:**	1	**Type:**	Enumeration
Desc:	List of traffic patterns.				
Enumerator	**TrafficPattern.Unknown**		**Value:**	0	
Enumerator	**TrafficPattern.Continuous**		**Value:**	1	
Enumerator	**TrafficPattern.PeriodicBurst**		**Value:**	2	
Enumerator	**TrafficPattern.Random**		**Value:**	3	

6.3.39 TrafficInformation

The TrafficInformation parameter indicates channel load (e.g., code allocation ratio, time slot allocation ratio, and subcarrier allocation ratio). This parameter is helpful for balancing traffic among channels (e.g., off loading peak traffic into other channels).

Name:	**TrafficInformation**	**Phys. unit**	—	**Extends:**	—
ID:	406	**Size:**	1	**Type:**	Float
Desc:	The traffic information parameter indicates channel load.				

6.3.40 SignalType

The SignalType parameter specifies the detected signal in terms of its RF characteristics when possible. Such characterization can be very useful to perform a coarse sensing first followed by a second level of fine sensing once the signal type is known.

Name:	**SignalType**	**Phys. unit**	--	**Extends:**	--
ID:	407	**Size:**	1	**Type:**	Enumeration
Desc:	List of known signal types for RF signal characterization.				
Enumerator	**SignalType.Unknown**	**Value:**	0	**NOTE**	1
Enumerator	**SignalType.Compound**	**Value:**	1	**NOTE**	2
Enumerator	**SignalType.Noise**	**Value:**	2	**NOTE**	3
Enumerator	**SignalType.Multicarrier**	**Value:**	3	**NOTE**	-—
Enumerator	**SignalType.Singlecarrier**	**Value:**	4	**NOTE**	-—
NOTE 1—*Unknown* is selected if the sensor cannot determine a valid signal type. NOTE 2—*Compound* is selected if the sensor detects a signal too complex to describe by a single parameter (e.g., OFDM with time-variant subcarrier modulation). NOTE 3—*Noise* is selected if the sensor detects structured noise (e.g., correlated) but cannot determine the underlying signal type (e.g., due to low power level).					

6.3.41 SignalDesc

The signal description parameter represents a set of information elements that describe the behavior of the signal such as traffic pattern, type of signal, and modulation scheme.

Name:	**SignalDesc**	**Phys. unit**	--	**Extends:**	--
ID:	408	**Size:**	variable	**Type:**	Structured
Desc:	Detail description of a detected RF signal.				
.0	**SignalDesc.Amplitude**			**Type:**	Float
.1	**SignalDesc.AngleOfArrival**			**Type:**	Angle
.2	**SignalDesc.Bandwidth**			**Type:**	Bandwidth
.3	**SignalDesc.Dutycycle**			**Type:**	Unsigned integer
.4	**SignalDesc.Frequency**			**Type:**	Frequency
.5	**SignalDesc.Modulation**			**Type:**	ModulationType
.6	**SignalDesc.Phase**			**Type:**	Angle
.7	**SignalDesc.Power**			**Type:**	Power
.8	**SignalDesc.TrafficInformation**			**Type:**	TrafficInformation
.9	**SignalDesc.TrafficPattern**			**Type:**	TrafficPattern
.10	**SignalDesc.Type**			**Type:**	SignalType
	Reserve				

6.3.42 RATID

The RATID parameter gives an ID of an existing standard RAT, for example, Global System for Mobile (GSM), etc.

Name:	**RATID**	**Phys. unit**	—	**Extends:**	—
ID:	409	**Size:**	1	**Type:**	Enumerated
Desc:	Unique identification of existing standard RATs known by both spectrum sensors and their clients.				
Enumerator	**RATID.GSM**		**Value:**	0	
Enumerator	**RATID.WiMAX**		**Value:**	1	
Enumerator	**RATID.CDMA2000**		**Value:**	2	
Enumerator	**RATID.WiFi**		**Value:**	3	

6.4 Data representation

This subclause gives the data representation for this standard using Unified Modeling Language (UML). The sensing-related information described in 6.3 is grouped into the following four classes:

— ControlInformation

— SensorInformation

— SensingInformation

— RegulatoryRequirement

6.4.1 ControlInformation Class

As shown in Figure 13, this class describes control parameters for realizing sensing information exchange between sensors and their clients.

— Transport

This class contains control parameters for transport of sensing information. Each instance of ControlInformation class can have 0 or 1 instance of Transport class.

— TransportNetwork

This class contains parameters to control network for the transportation of sensing information. The instance of Transport class can have only one instance of TransportNetwork class.

— TransportMAC

This class contains parameters to control MAC for the transportation of sensing information. The instance of Transport class can have only one instance of TransportMAC class.

— TransportMode

This class contains parameters to control the transportation mode for sensing information exchange, for example, information type such as soft information and hard information. An instance of Transport class can have only one instance of TransportMode class.

— Measurement

This class contains control parameters for spectrum measurement. Each instance of ControlInformation class can have zero or several instances of Measurement class.

— MeasurementObj

This class contains parameters to control the measurement objective, for example, the start frequency and end frequency of a radio frequency to be measured. Any instance of Measurement class can have one or several instances of MeasurementObj class.

— MeasurementProfile

This class contains parameters to control the behavior of measurement, for example, determining the technique that shall be used for spectrum sensing. Any instance of Measurement class can have one or several instances of MeasurementProfile class.

— MeasurementPerformance

This class contains parameters to control the performance of radio frequency measurement, (e.g., the desired probability of false alarm as the output of spectrum sensing). Any instance of Measurement class can have one or several instances of MeasurementPerformance class.

— Application

This class contains control parameters for sensing information request and sensing information provision. Each instance of ControlInformation class can have zero or several instances of Application class.

— SensingInformationRequest

This class contains parameters to set priority levels for the clients to request sensing information. Each instance of Application class has one or multiple instances of SensingInformationRequest class.

— SensingInformationProvision

This class contains parameters to set priority levels for the sensors when providing sensing information to their clients. Each instance of Application class has one or multiple instances of SensingInformationRequest class.

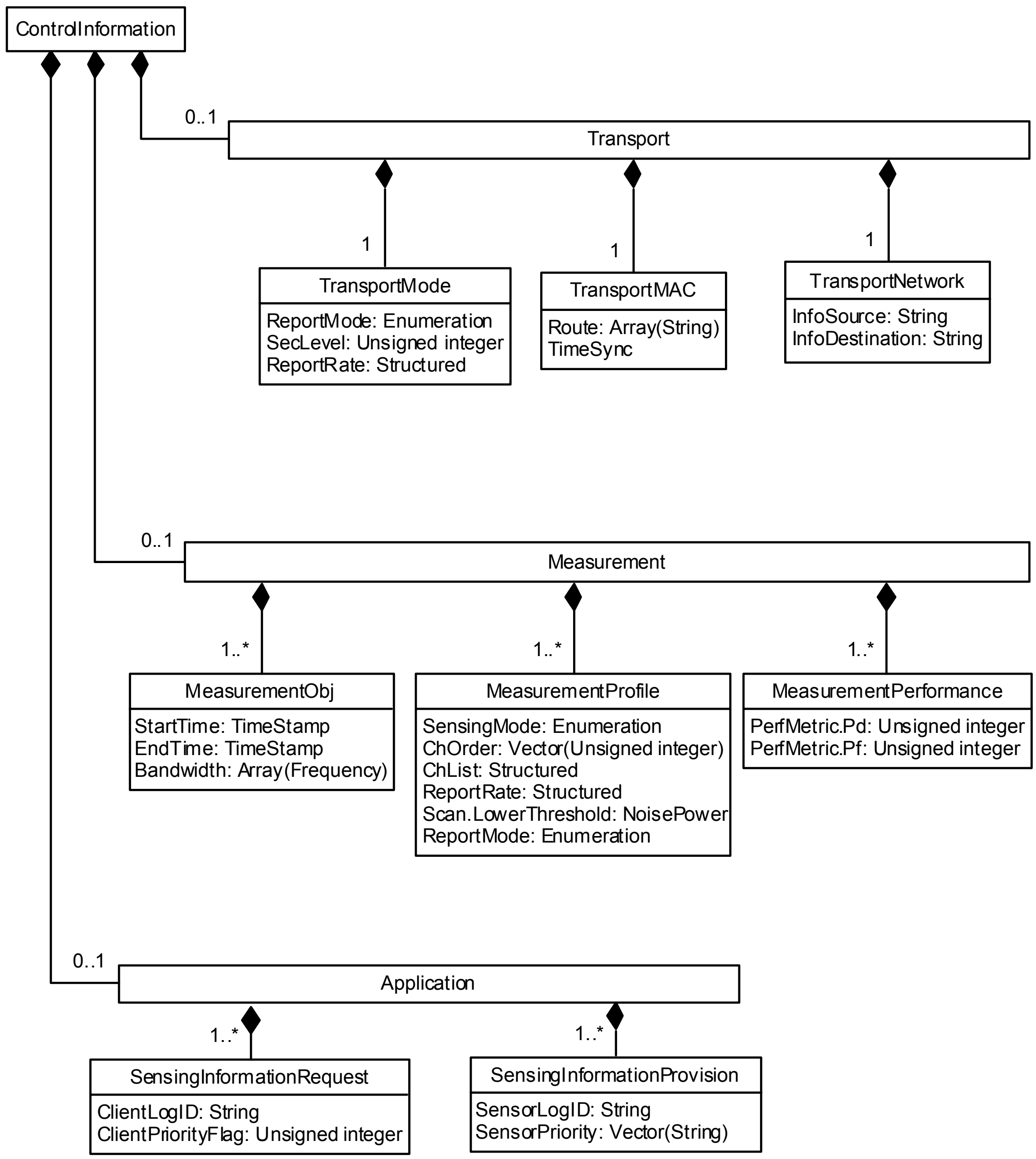


Figure 13—UML class diagram of control information classes

6.4.2 SensorInformation Class

As shown in Figure 14, this class describes parameters related to sensors.

— SensorPHYProfile

  This class contains information about PHY of sensors, for example, phase noise. Each instance of SensorInformation can have only one instance of SensorPHYProfile class.

- Location

  This class contains information about sensors location, for example, absolute location and relative location. Each instance of SensorPHYProfile can have zero or one instance of LocationProfile class.

- AntennaProfile

  This class contains information about sensors' antennas, for example, antenna hight, beam pattern, etc. Each instance of SensorPHYProfile can have zero or multiple instances of AntennaProfile class.

— SensorMACProfile

This class contains information about MAC of sensors, for example, MAC ID. Each instance of SensorInformation can have zero or one instance of SensorMACProfile class.

— SensorProfile

This class contains general information about sensors. Each instance of SensorInformation can have zero or one instance of SensorProfile class.

- ManufacturerInformation

  This class contains manufacturer information of sensors such as manufacturer's ID and product ID. Each instance of SensorProfile class has one instance of ManufacturerInformation class.

- PowerProfile

  This class contains power information of sensors such as battery level and power consumption. Each instance of SensorProfile class has one instance of PowerProfile class.

— SensorCapability

This class contains information about sensors' capability of radio spectrum measurement, such as measurement range, sensitivity, etc. Each instance of SensorInformation can have zero or one instance of SensorCapability class.

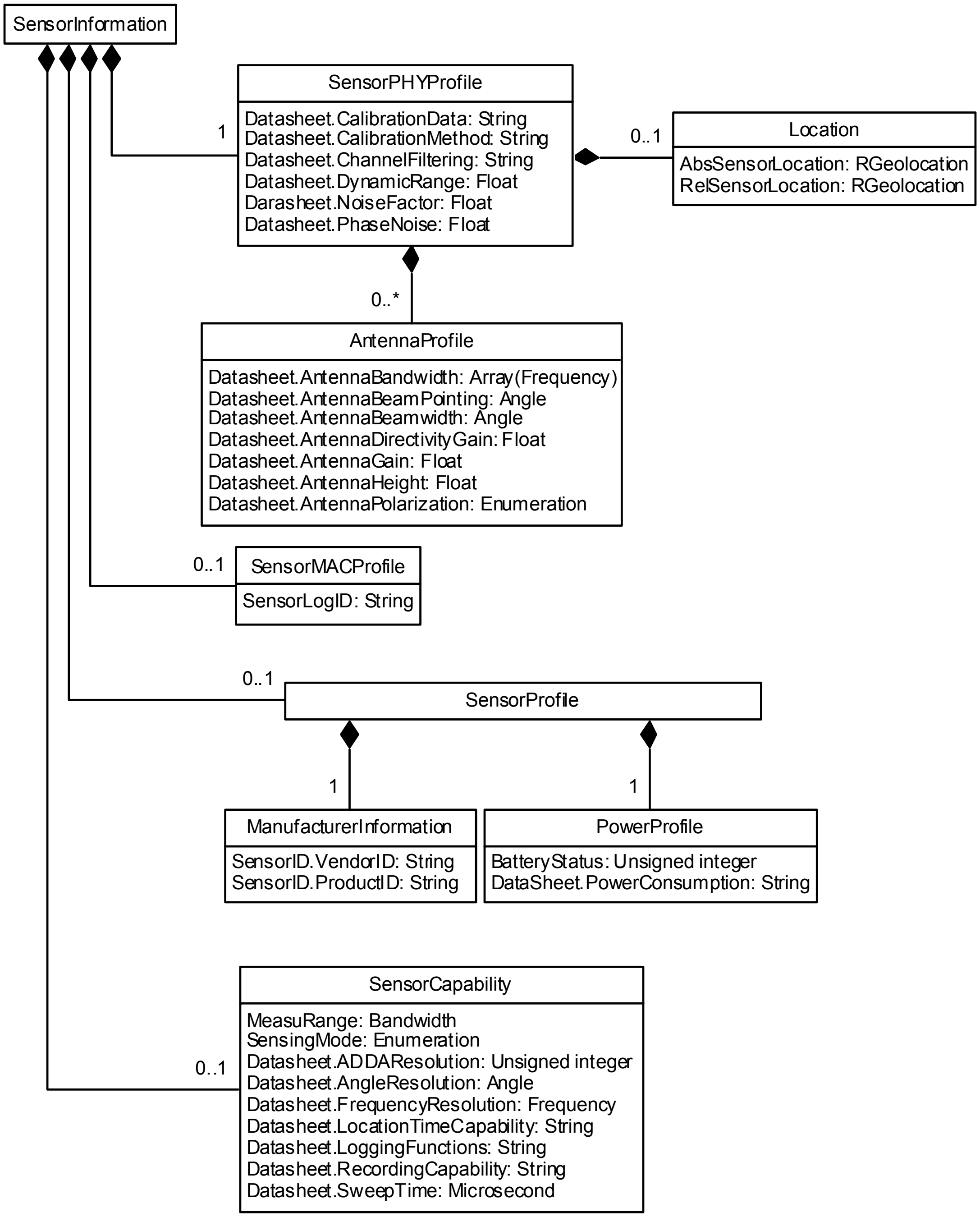


Figure 14—UML class diagram of sensor information classes

6.4.3 SensingInformation Class

As shown in Figure 15, this class describes parameters of sensing results.

- SensingInformationProfile

  This class contains general information about the measurement that is carried out to produce sensing information. For example, the information can be confidence level of the measurement, measurement feature, threshold used in the measurement, etc. One instance of SensingInformation class has only one instance of SensingInformationProfile class.

- LocationInformation

  This class contains information about the location where the measurement has been carried out, for example, absolute location and relative location. Each instance of SensingInformation class has zero or one instance of LocationInformation class.

- Signal

  This class contains information about the measured signals. Each instance of SensingInformation class has zero or multiple instances of Signal class.

  - SignalProfile

    This class contains general information about the measured signals, such as signal level, estimated amplitude, etc. Each instance of Signal class can have only one instance of SignalProfile class.

  - SignalBehavior

    This class contains information describing the behavior of the measured signals such as duty cycle. One instance of Signal class can have zero or one instance of SignalBehavior class.

- Channel

  This class contains information about the measured channel. Each instance of SensingInformation class has zero or multiple instances of Channel class.

  - ChannelProfile

    This class contains general information about the measured channel such as start frequency and stop frequency. Each instance of Channel class has only one instance of ChannelProfile class.

  - ChannelMeasurement

    This class contains information about the measurement results of the channel such as measured bandwidth and noise power. Each instance of Channel class has zero or one instance ChannelMeasurement class.

- RAT

  This class contains information about RAT IDs. Each instance of SensingInformation class has zero or multiple instances of RAT class.

  - RATProfile

    This class contains information about the IDs of RAT. Each instance of RAT class has zero or one instance of RATProfile class.

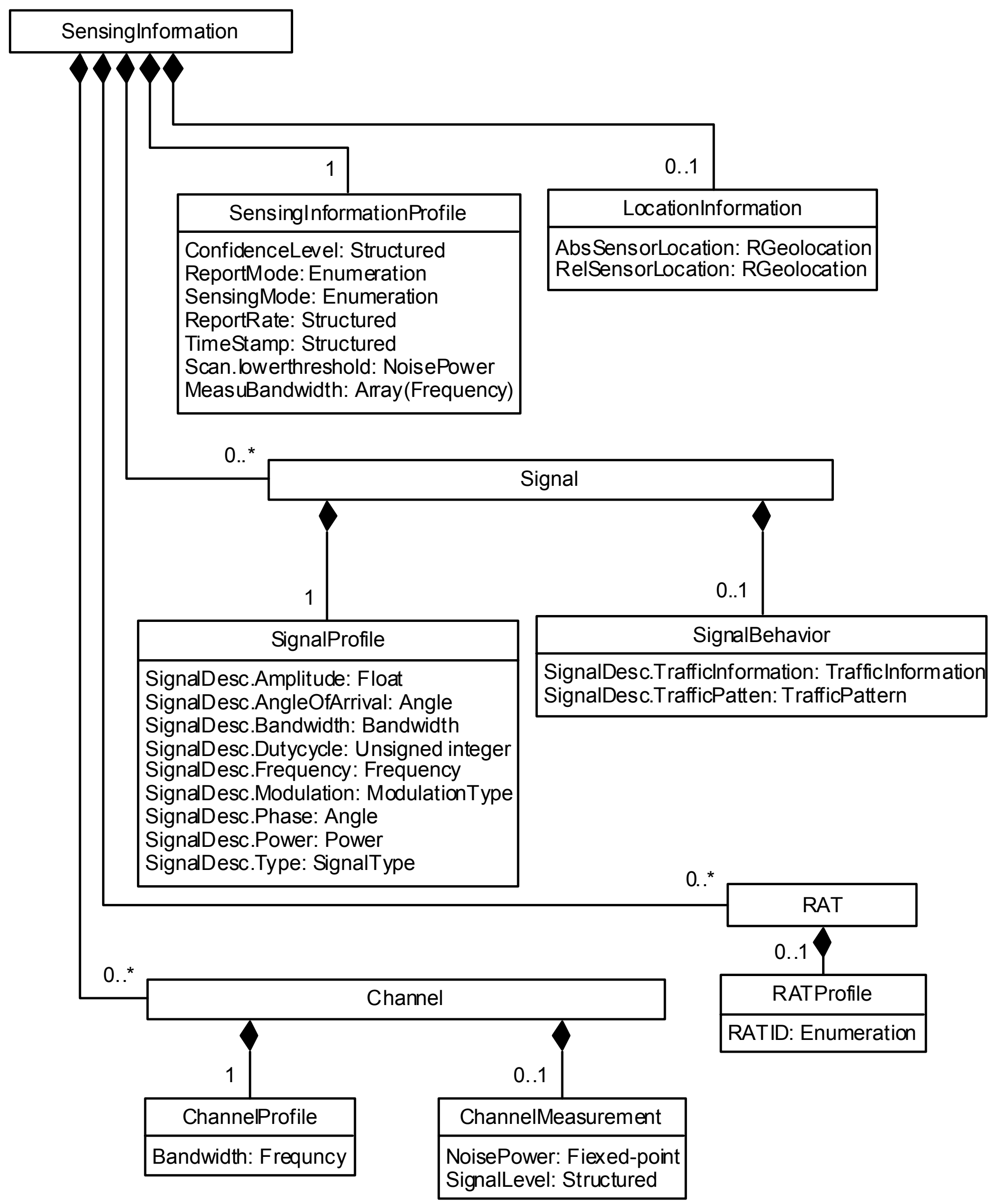


Figure 15—UML class diagram of sensing information classes

6.4.4 RegulatoryRequirement Class

As shown in Figure 16, this class contains requirements derived from regulation according to different primary user signals at different locations. The types of signals are different in terms of regulations.

— PrimaryUserSignals

  This class contains the regulatory requirements of primary user signals such as detection threshold, detection bandwidth, etc. Each instance of RegulatoryRequirement class has zero or multiple instances of PrimaryUserSignals class depending on different regulatory requirements.

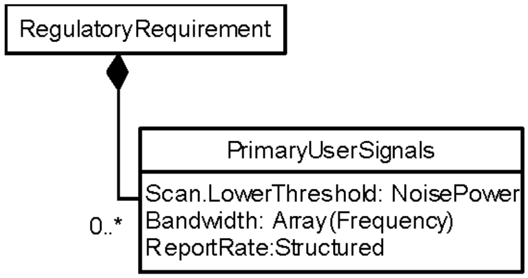


Figure 16—UML class diagram of regulatory requirement classes

7. State diagram and generic procedures

Figure 17 shows a high-level state diagram for an IEEE 1900.6 logical entity. It includes initialization state, idle state, data gathering state, simultaneous communication and data gathering state, and communication state.

7.1 State description

7.1.1 Initialization state

The initialization state includes functions that shall be executed before sensing information exchange, for example, obtaining the control/application ID.

7.1.2 Idle state

This is the state when there is no usage of the logical interface.

7.1.3 Data gathering state

The data gathering state includes functions that shall be executed for obtaining measurement results. It also includes functions that shall be executed for obtaining sensing information for the control/application.

7.1.4 Communication state

The communication state includes functions that shall be executed for transportation of sensing-related information from an IEEE 1900.6 logical entity to another IEEE 1900.6 logical entity using the communication subsystem.

7.1.5 Simultaneous communication and data gathering state

In the simultaneous communication and data gathering state, communication for sensing information exchange and data gathering take place at the same time. For example, the measurement module performs spectrum measurement at frequency f1 and the communication subsystem is transporting sensing information at another frequency f2. The sensor may also receive sensing control information during execution of the measurement.

Note that this state may not be present in simple systems performing only one action at a time (communication or data gathering).

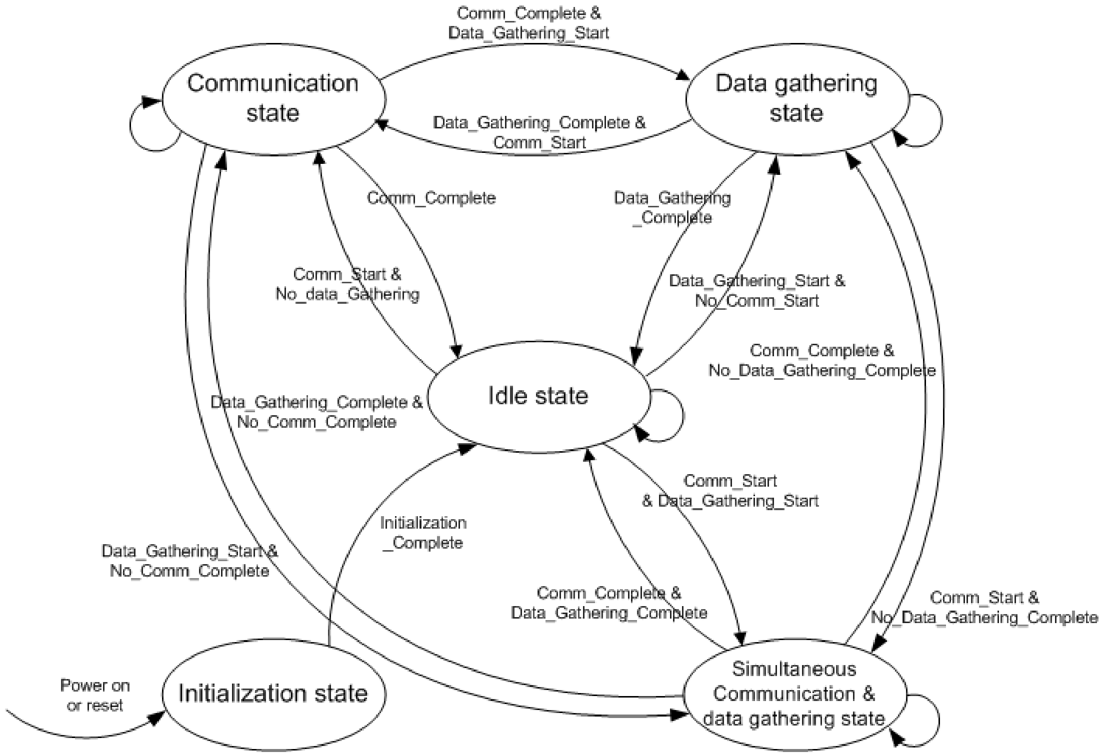


Figure 17—State diagram of IEEE 1900.6 logical entity

7.2 State transition description

7.2.1 Initialization state→Idle state

After power on or reset, IEEE 1900.6 logical entities go to idle state waiting for other commands.

7.2.2 Idle state→Data gathering state

The IEEE 1900.6 logical entity is triggered to collect sensing information without communication request. This excludes the transition from the idle state to the communication state. For example, the control/application initiates spectrum sensing by an internal spectrum measurement module and starts to collect sensing information.

7.2.3 Idle state←Data gathering state

The IEEE 1900.6 logical entity acknowledges that sensing information has been obtained, which indicates the completion of data gathering.

7.2.4 Idle state→Communication state

The IEEE 1900.6 logical entity requests to send available sensing information and commands/primitives remote entity without data gathering request. This excludes the transition from the idle state to the data gathering state.

7.2.5 Idle state←Communication state

The IEEE 1900.6 logical entity acknowledges that the sensing information or commands/primitives payload has been received.

7.2.6 Data gathering state→Communication state

The IEEE 1900.6 logical entity requests to send available sensing information and commands/primitives remote entity upon completion of data gathering.

7.2.7 Data gathering state←Communication state

The IEEE 1900.6 logical entity starts data gathering after confirming that the sensing information or commands/primitives payload have been received.

7.2.8 Idle→Simultaneous communication and data gathering state

The IEEE 1900.6 logical entity starts both data gathering and communication at the same time.

7.2.9 Idle←Simultaneous communication and data gathering state

The IEEE 1900.6 logical entity completes a data gathering task and communication task at the same time.

7.2.10 Simultaneous communication and data gathering state→Communication state

The IEEE 1900.6 logical entity ends the data gathering while maintaining the communication status; i.e., the communication task is ongoing.

7.2.11 Simultaneous communication and data gathering state←Communication state

The IEEE 1900.6 logical entity is requested to initiate a data gathering task while the previously initiated communications task continues.

7.2.12 Simultaneous communication and data gathering state→Data gathering state

The IEEE 1900.6 logical entity completes its communications task while continuing its data gathering task.

7.2.13 Simultaneous communication and data gathering state←Data gathering state

The IEEE 1900.6 logical entity is requested to start the communication for sending information to another entity as payload while maintaining the data gathering status. The IEEE 1900.6 logical entity may also receive configuration commands while executing measurement.

7.3 Generic procedures

7.3.1 Purpose

This subclause provides generic procedures and usage examples for the exchange of sensing-related information between an IEEE 1900.6 logical entity and its client.

An IEEE 1900.6 logical entity shall be a CE, DA, or Sensor as defined in 3.1. Any IEEE 1900.6 logical entity can be a client to one or more of these. Additionally, any other entity not explicitly defined by the standard can act as a client provided that the interface procedures are used in conformance with the standard.

Without loss of generality, for the following discussion, a Sensor is assumed as the primary source of information. CE and DA are assumed to act primarily as clients but may also provide information within specific scenarios.

To express the procedures, the service user and service provider model concept as specified by ITU-T X.210 [B5] is used as shown in Figure 18 (cf. 3.1). The interface between the IEEE 1900.6 logical entity and its client is assumed transparent in this information flow with respect to any kind of transport service that may be needed to mediate between the service user side and service provider side of the interface.

7.3.2 Generic procedure and notations

In the description of the generic procedure, the following terms are used along with the terms known in the document.

IEEE 1900.6 Service Provider: An abstraction of the totality of those entities that provide an IEEE 1900.6 service to the service user. The IEEE 1900.6 service provider includes the IEEE 1900.6 service (as indicated in the reference model in Clause 5) and the associated 1900.6 SAPs that instantiate the logical interface for exchange of sensing-related information between the IEEE 1900.6 servers and clients. (This is a logical mechanism that should be seen different from the IEEE 1900.6 logical entities.)

IEEE 1900.6 Service User: The IEEE 1900.6 logical entities that shall implement or use the IEEE 1900.6 logical interface to exchange sensing information. For example, the IEEE 1900.6 logical entity that shall use the logical interface to obtain sensing-related information plays the client role. While the IEEE 1900.6 logical entity that shall use the logical interface to provide sensing information plays a server role. This means, the client controls the initiation and termination of the sensing information exchange.

The basic communication flow for a synchronous, blocking primitive is described as follows:

— When a client IEEE 1900.6 logical entity needs to communicate with another IEEE 1900.6 logical entity (server) (e.g., to acquire sensing information), it will create a request (request message) to an IEEE 1900.6 service provider.

— In the following message exchange procedure, the IEEE 1900.6 service forwards the request along with its parameters toward the remote IEEE 1900.6 logical entity. The parameters from the IEEE 1900.6 logical interface forms the service data unit (SDU), which shall be communicated between client and server as the payload of a protocol data unit (PDU) used by the protocol layers beneath. Upon reception of an SDU, the service provider generates an indication toward the remote IEEE 1900.6 logical entity (server). The actual procedures and protocols used in the communication subsystem are out of the scope of this standard. Upon receiving the indication, the server performs further processing to consider or reject the indication.

— In response to the indication, the IEEE 1900.6 logical entity (server) generates a response to the IEEE 1900.6 service provider to communicate the results (e.g., the sensing parameters requested), toward the client (response message) via the logical interface and supporting C-SAP. It is also possible for either an error message or an acknowledgement message to be conveyed in the response message.

— Upon successful reception of the response message on the client side, the IEEE 1900.6 service provider generates a confirmation toward the initiating client providing the SDU as originated by the remote IEEE 1900.6 logical entity (server). For this type of synchronous communication flow, the confirmation indicates that the IEEE 1900.6 request has been completed and the client may now check if it obtained valid results.

Note that, in the preceding communication message flow (also as depicted in Figure 18(a) and Figure 18(b)), only the service request from service user A or client is depicted. In this case, the service user B or the server is stated as "acceptor," whereas the service user is stated as "requestor." The requestor and acceptor role changes depending on who is initiating the service request.

— Requestor: A service user from whom a service request originates

— Acceptor: A service user to whom a service request is targeted

Following these roles of the service users, the four primitives [depicted in Figure 18(a) and Figure 18(b)] can be written only by using two basic service-primitives known as submit and deliver, as follows:

— Service-primitive:

  Submit: a basic service primitive issued by service user

  Deliver: a basic service primitive issued by service provider

— Primitives:

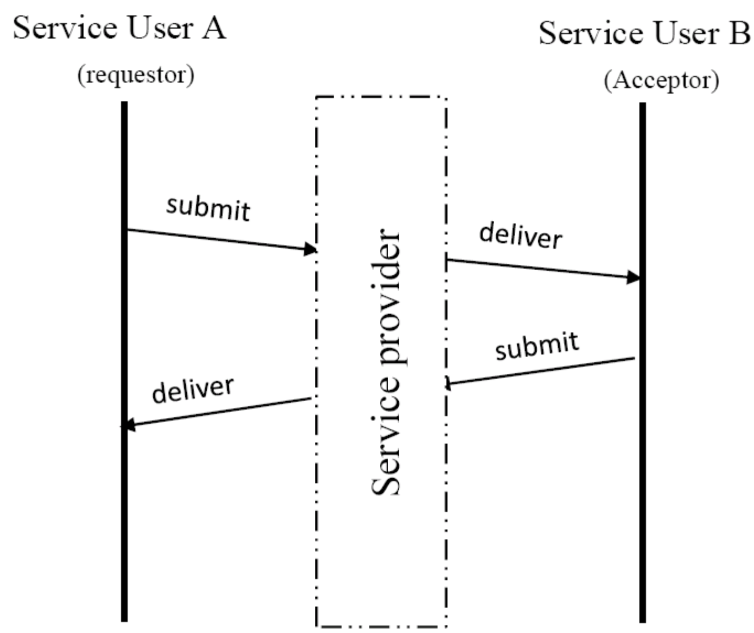
  Request, or requestor.submit

  Indication, or acceptor.deliver

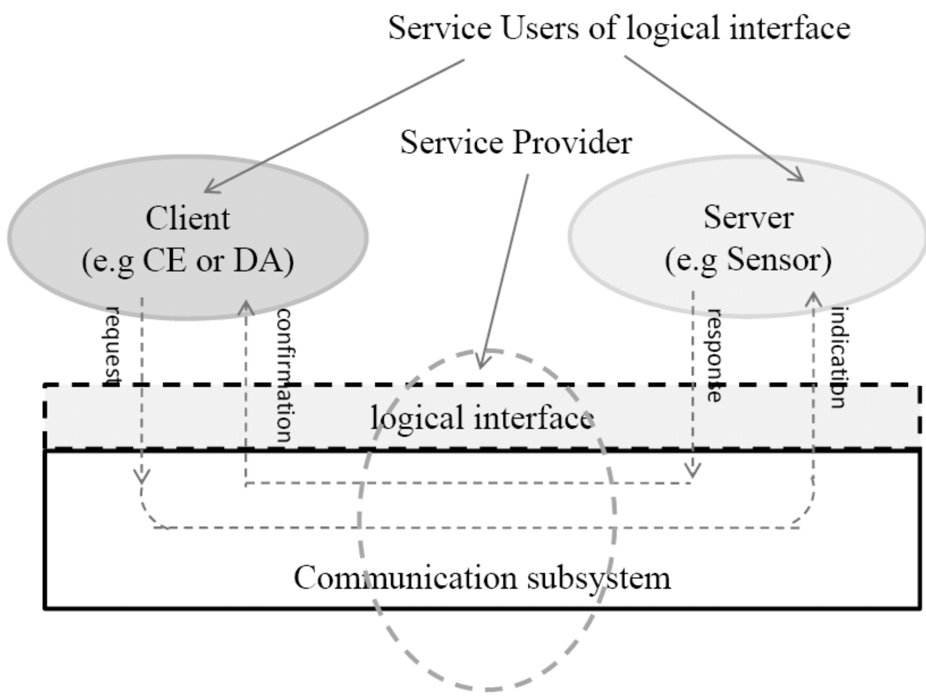
  Response, or acceptor.submit

  Confirm, or requestor.deliver

For the purpose of this standard, indication primitives are equivalent to their request counterparts, and confirmation primitives are equivalent to their response counterparts.



(a)



(b)

Figure 18—(a) Information exchange between service users of logical interface and (b) logical interface on top of communication subsystem acting as service provider for the exchange of information between client and server

7.4 Example procedures for use cases

This subclause presents multistep procedures by using the generic procedure shown in 7.3. These examples are based on different instances of the IEEE 1900.6 logical entities paired for sensing information exchange involved in different scenarios in Clause 4.

7.4.1 CE–Sensor procedure

In this example, as depicted in Figure 19, a sequence of the generic procedure is used to portray a typical information exchange between CE and Sensor. The procedure is initialized by the spectrum sensing demand by the CE. The complete cycle to obtain sensing information can follow the following steps:

a) CE detects a spectrum sensing demand (internal)

b) CE performs spectrum sensing if it has embedded sensor (internal)

c) CE detects control channel (to obtain additional information from externally located sensor)

d) CE identifies spectrum sensors (sending control signal and receiving response)

    1) Request: CE submits a request on the presence of spectrum sensor/s

    2) Indication: Deliver primitive is issued to the sensor so that it responds on its presence

    3) Response: Sensor submits its presence status information

    4) Confirm: Deliver primitive is issued to CE to confirm successful reception of message from sensor

e) CE identifies capability of the sensor

    1) Request: CE submits a request to provide sensor's profile

    2) Indication: Deliver primitive is issued to the sensor so that it responds to the request

    3) Respond: Sensor submits its profile information

    4) Confirm: Deliver primitive is issued to the CE to confirm successful reception of the sensor's message

f) CE requests sensing information

    1) Request: CE submits a request to provide sensing information (by specifying control parameters)

    2) Indication: Deliver primitive is issued to the sensor so that it responds to the request

    3) Respond: Sensor submits sensing information

    4) Confirm: Deliver primitive is issued to CE to confirm successful reception of sensor's message

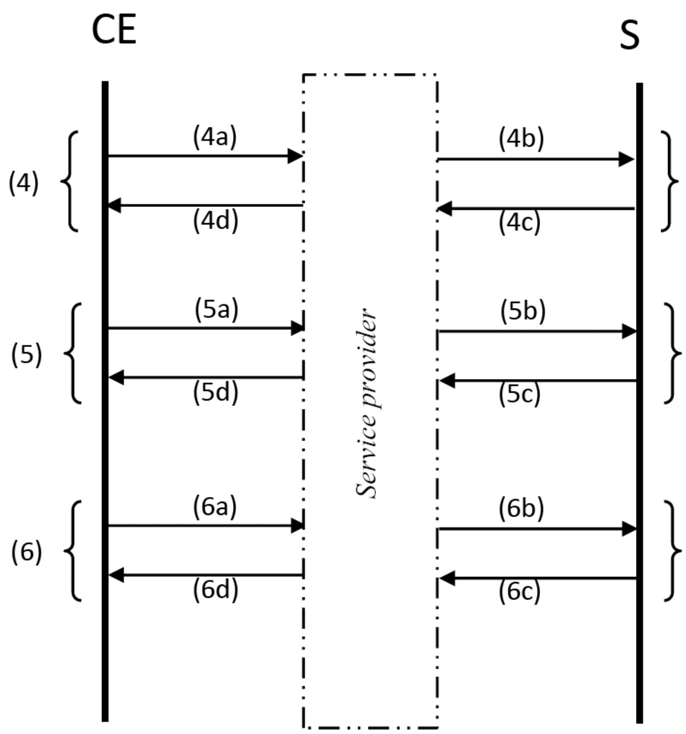


Figure 19—Sensing information exchange procedure between CE and Sensor (S)

7.4.2 DA–Sensor procedure

In this example, as depicted in Figure 20, a sequence of the generic procedure is used to portray a typical information exchange between DE and Sensor. The procedure is initialized by the spectrum sensing demand in the DA. The complete cycle to obtain sensing information might follow the following steps:

a) DA regularly updates sensing and related information (internal)

b) DA detects control channel (to obtain sensing information from externally located sensor/s)

c) DA identifies spectrum sensors (sending control signal and receiving response)

   1) Request: DA submits a request on the presence of spectrum sensor(s)

   2) Indication: Deliver primitive is issued to the sensor so that it responds on its presence

   3) Response: Sensor submits its presence status information

   4) Confirm: Deliver primitive is issued to DA to confirm successful reception of message from sensor

d) DA identifies capability of the sensor

   1) Request: DA submits a request to provide sensor's profile

   2) Indication: Deliver primitive is issued to the sensor so that it responds to the request

   3) Respond: Sensor submits its profile information

   4) Confirm: Deliver primitive is issued to DA to confirm successful reception of the sensor's message

e) DA requests sensing information

  1) Request: DA submits a request to provide sensing information (by specifying control parameters)

  2) Indication: Deliver primitive is issued to the sensor so that it responds to the request

  3) Respond: Sensor submits sensing information

  4) Confirm: Deliver primitive is issued to DA to confirm successful reception of sensor's message

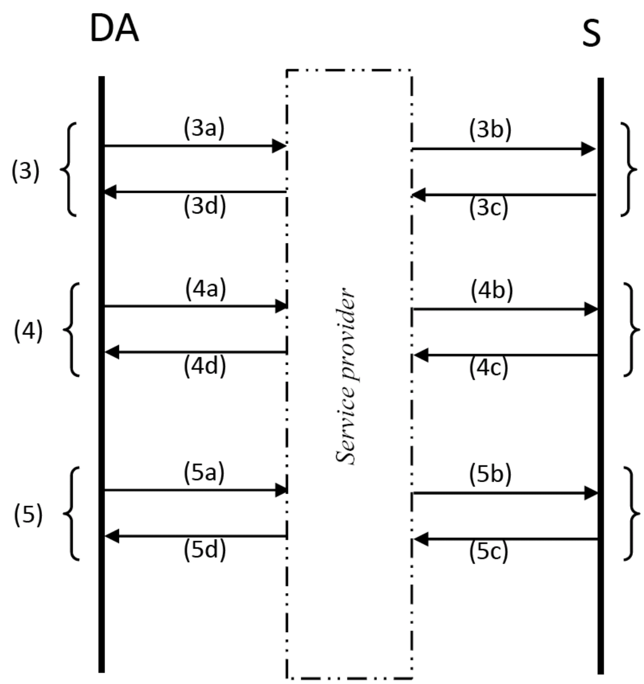


Figure 20—Sensing information exchange procedure between DA and Sensor (S)

7.4.3 CE–DA procedure

In this example, as depicted in Figure 21, a sequence of the generic procedure is used to portray a typical information exchange between CE and DA. The procedure is initialized by the spectrum sensing demand in the CE. The complete cycle to obtain sensing information might follow the following steps:

a) CE detects a spectrum sensing demand (internal)

b) CE performs spectrum sensing if it has embedded sensor (internal)

c) CE detects control channel (to obtain additional information from externally located data source or DA)

d) CE identifies DA (sending control signal and receiving response)

  1) Request: CE submits a request on the presence of DA

  2) Indication: Deliver primitive is issued to the DA so that it responds on its presence

  3) Response: DA submits its presence status information

  4) Confirm: Deliver primitive is issued to CE to confirm successful reception of message from CE

e) CE identifies type (list) of available information on the DA
   1) Request: CE submits a request to provide type of information available on DA
   2) Indication: Deliver primitive is issued to the DA so that it responds to the request
   3) Respond: DA submits type of available information
   4) Confirm: Deliver primitive is issued to CE to confirm successful reception of the sensor's message

f) CE requests sensing information
   1) Request: CE submits a request to provide sensing information by specifying control parameters, (e.g., for filtering).
   2) Indication: Deliver primitive is issued to the DA so that it responds for the request
   3) Respond: DA submits sensing information
   4) Confirm: Deliver primitive is issued to CE to confirm successful reception of the DA's message

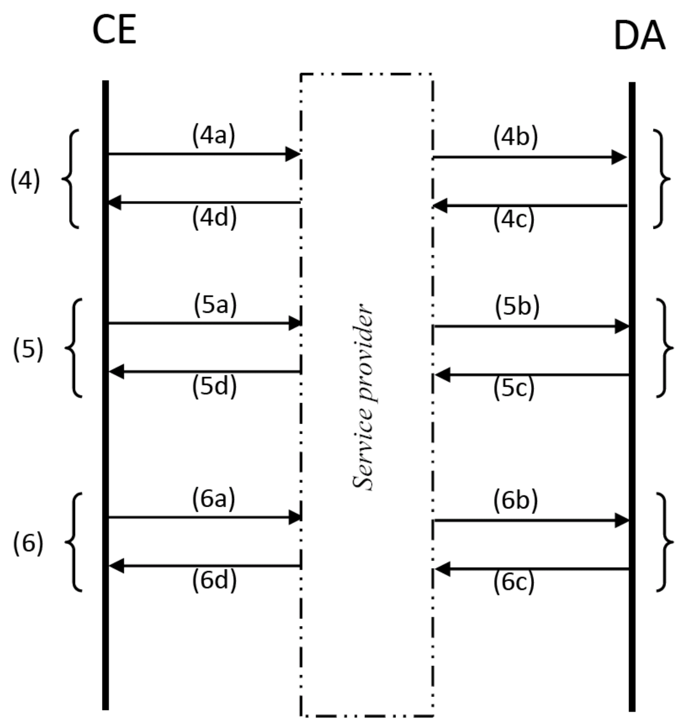


Figure 21—Sensing information exchange procedure between CE and DA

7.4.4 CE–CE procedure

In this example, as depicted in Figure 22, a sequence of the generic procedure is used to portray a typical information exchange between two cooperating CEs. The procedure is initialized by the spectrum sensing demand in the client CE. The complete cycle to obtain sensing information might follow the following steps:

a) CE_A detects a spectrum sensing demand (internal)

b) CE_A performs spectrum sensing if it has embedded sensor (internal)

c) CE_A detects control channel (to obtain additional information from CE_B)

d) CE_A identifies CE_B (sending control signal and receiving response)

   1) Request: CEA submits a request on the presence of CEB
   2) Indication: Deliver primitive is issued to the CEB so that it responds on its presence
   3) Response: CEB submits its presence status information
   4) Confirm: Deliver primitive is issued to CEA to confirm successful reception of message from CEB

e) CE_A identifies sensing capability of CE_B and its appropriateness for distributed sensing

   1) Request: CEA submits a request to provide capability of CEB for distributed sensing
   2) Indication: Deliver primitive is issued to the CEB so that it responds for the request
   3) Respond: CEB submits its capability (availability) for distributed sensing
   4) Confirm: Deliver primitive is issued to CEA to confirm successful reception of sensor's message

f) CE_A requests sensing information

   1) Request: CEA submits a request to provide sensing information (by specifying control parameters…)
   2) Indication: Deliver primitive is issued to the CEB so that it responds for the request
   3) Respond: CEB submits sensing information
   4) Confirm: Deliver primitive is issued to CEA to confirm successful reception of CEB's message

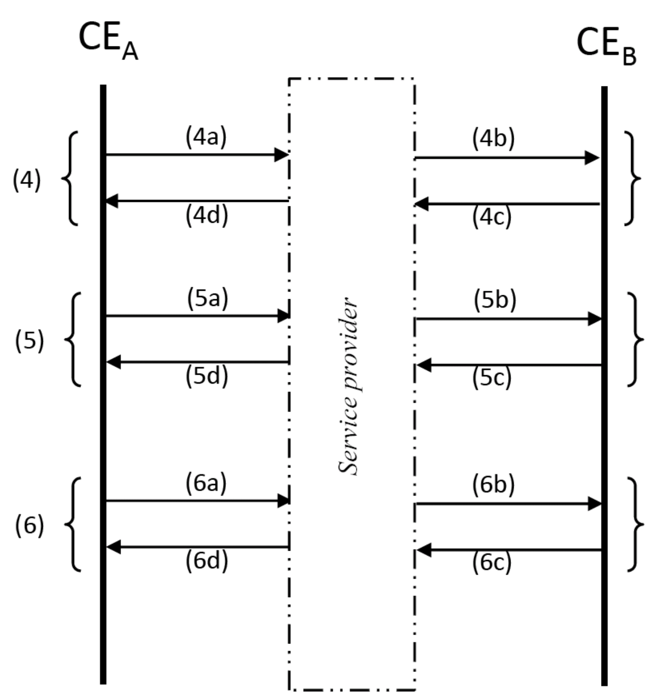


Figure 22 —Sensing information exchange procedure between CE_A and CE_B

7.4.5 Sensor–Sensor procedure

In this example, as depicted in Figure 23, a sequence of the generic procedure is used to portray a typical information exchange between two cooperating Sensors. The procedure is initialized by the client Sensor request for sensing information. The complete cycle to obtain sensing information might follow the following steps:

a) S_A detects a spectrum sensing request (multiple-device spectrum sensing mode)

b) S_A performs spectrum sensing (internal)

c) S_A detects control channel (to obtain sensing information from S_B either for distributed sensing or for simple relaying)

d) S_A identifies S_B (sending control signal and receiving response)

   1) Request: SA submits a request on the presence of SB

   2) Indication: Deliver primitive is issued to the SB so that it responds on its presence

   3) Response: SB submits its presence status information

   4) Confirm: Deliver primitive is issued to SA to confirm successful reception of message from SB

e) S_A requests sensing information

   1) Request: S_A submits a request to provide sensing information (e.g., to relay the sensing information to another client)

   2) Indication: Deliver primitive is issued to the S_B so that it responds to the request

   3) Respond: S_B submits sensing information

   4) Confirm: Deliver primitive is issued to S_A to confirm successful reception of S_B's message

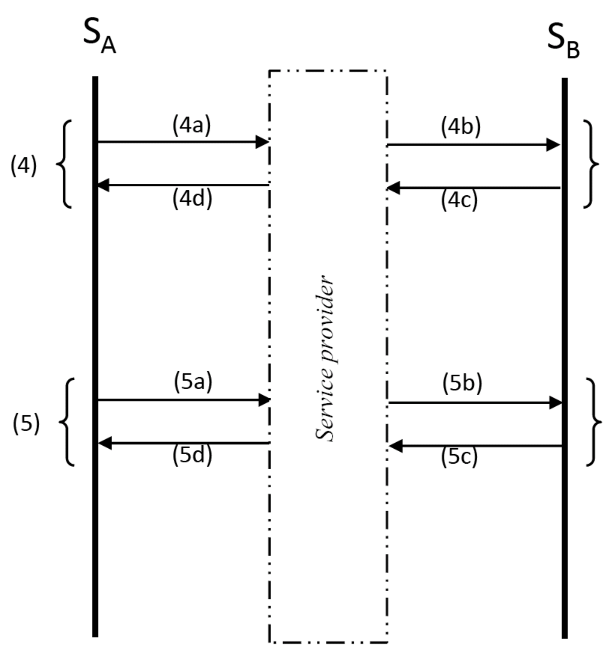


Figure 23—Sensing information exchange procedure between S_A and S_B

Annex A

(informative)

Use cases

CRs should be able to detect independently the presence of primary users through continuous spectrum sensing. The technical characteristics of the primary users will require varying sensing requirements depending on the use case. For example, sensitivity requirements or sensing rates may vary depending on the use cases. Detecting GPS signals, for example, is more difficult than detecting TV signals because the GPS receiver sensitivity is higher than that of TV receivers. Thus, for each use case, a set of sensing requirements consistent with the scope of this standard is identified.

Subclauses A.1 and A.2 define spectrum sensing use cases from different perspectives. A detailed analysis of these use cases with respect to the underlying spectrum usage model and the sensing requirements that can be derived from these use cases is given after each use case description.

A.1 Use cases from the perspective of spectrum usage

Subclauses A.1.1 to A.1.4 examine use cases from the perspective of spectrum usage. The usage models that use spectrum sensing for purposes of opportunistic access to unutilized spectrum are considered. Service type alignment is one of many criteria for categorization of use cases. From the industry point of view, how the spectrum is used (such as who owns the spectrum) in each use cases is a major factor.

CR usage models can be broadly categorized into two models: substitute for primary operation of longer duration (SPOLD) and substitute for primary operation of shorter duration (SPOSD). These describe the relative long-term or short-term usage of spectrum as in the case of the primary system's mode of operation.

A.1.1 SPOLD

In these usage models, CR usage of spectrum is as in the case of the primary system's mode of operation. The spectrum usage is of relatively longer duration. The following subclauses provide examples of each substitute primary operation identified.

A.1.1.1 Emergency services (SPOLD1)

Use case

After a major disaster, much of the existing communications infrastructure is damaged, and emergency personnel from around the country bring their own communications equipment to assist with the response. CR allows all of the emergency personnel to find common usage channels throughout the disaster area in which the CR can operate without causing interference to the surviving legacy communications systems.

Use case analysis

- Spectrum usage model and scenario classification
  - Long-term spectrum usage::1:1
- Sensing requirements
  - Reconfigurable sensing performance.

  - Sensing of multiple channels simultaneously.
  - Only information on signal occupancy can be sufficient (band is free or occupied).
  - Sensing is performed only at initial stage (during or after the catastrophe).
  - Sensing is performed by embedded sensors with in radios (if infrastructure and fixed sensors are destroyed) or temporary deployment of distributed sensors.
- Usage of interface
  - CE–S
- Key sensing information
  - Target detection performance (Pd, Pf, etc.)
  - Sensing method, techniques
  - Sensing duration (channel latency issue)

The key feature of this use case is that the priority level of the emergency system might increase and the emergency system might become primary users. In this situation, the emergency system, even as a secondary system, may have more aggressive usage of spectrum and less protection to primary users. The sensing system should have a reconfiguration capability such that their probability of detection and probability of false alarm can be changed or their sensing techniques can be changed.

To detect spectrum opportunities as quickly as possible, multiple channels should be sensed simultaneously.

Because the target of sensing is to identify occupancy not usage analysis, information on signal occupancy might be enough.

Sensing might be performed at the initial stage when the disaster happened. Once the emergency system has obtained the spectrum, it will take control of the spectrum over a long period.

A.1.1.2 Load sharing to reduce blocking at peak traffic times (SPOLD2)

Use case

Two or more services (e.g., IEEE 802.11 and Cellular) exhibit peak traffic loads at different times and locations. The IEEE 802.11 service may peak during working hours or during the evenings, whereas the Cellular system may exhibit peaks during commute time. Off loading peak traffic into the spectrum of the other service would be possible through the use of CR.

Use case analysis

- Spectrum usage model and scenario classification
  - Long-term spectrum usage::1:1
- Sensing requirements
  - Traffic sensing capability
  - Sensing to identify service channels
  - Sense the most suitable channel first
- Usage of interface
  - CE–S, CE–DA

- Key sensing information
  - Traffic information (resource allocation ratio)
  - Service channels (RATs) ID

To perform load sharing, the network needs to know the traffic of different services.

The CR needs to identify which service channels are available in their environment.

The history of the traffic information might be stored at the DA. Based on this information, the terminals can sense the service channel that is more likely to have low traffic.

A.1.1.3 Sharing of spectrum to reduce blocking at peak traffic times (SPOLD3)

Use case

Two or more services (e.g., WiMAX™[8] and long-term evolution [LTE]) may exhibit peak loads at different times and locations. Offloading peak traffic into the spectrum of the other service would be possible through the use of CR.[9]

Use case analysis

- Spectrum usage model and scenario classification
  - Long-term spectrum usage::1:N
- Sensing requirements
  - Traffic sensing capability (mostly at base stations [BSs])
  - Sensing of white space (mostly at BSs)
  - Sensing to identify service channels
  - Check with DA
  - Sense and vacate immediately if a primary signal is detected
- Usage of interface
  - CE–S, DA–S, S–S
- Key sensing information
  - Traffic information (resource allocation ratio)
  - Service channel ID

Part of the spectrum of a service channel might be shared with another service channel to reduce blocking at peak traffic times. In this case, the traffic information should be known. The white space (unused spectrum) of each service should be identified so that the spectrum can be shared among different services. Note that "white space" is not limited to TV white space but refers to all spectrum opportunities.

A.1.1.4 Self-management of uncoordinated spectrum (SPOLD4)

Use case

In the unlicensed bands, the high density of privately owned networks, together with freedom of movement requirements for certain transitory operations, makes the central coordination of frequency assignments highly impractical. CR provides an effective means of self-coordination to avoid interference with other networks while providing useful throughput.

This is an example of interference mitigation technique on frequency assignment. This can be in the case of public safety, conferences, etc., as the type of user is immaterial. The central factor is the temporary application of the bands.

Use case analysis

- Spectrum usage model and scenario classification
  - Long-term spectrum usage::M:1
- Sensing requirements
  - Sensing of multiple channels (used and unused white space)
  - Sensing to prevent interference
  - Provide sensing information to multiple clients
- Usage of interface
  - CE–S, CE–CE, DA–S
- Key sensing information
  - Client priority (priority flag is set upon coordination of participating engines)
  - Duty cycle (to predict potential interference)

The key feature of self-coordination is sensing to avoid interference. Multiple channels should be sensed to identify unused spectrum and the duty cycle of spectrum usage. Multiple CRs might use the same sensor (for example, a standalone sensor to provide sensing services over an application area).

A.1.1.5 Introduction of new users or services (SPOLD5)

Use case

The near 100% assignment of frequencies in high-density areas makes the introduction of new users or services almost impossible because very few existing licensees are willing to give up what has become a valuable asset. As a result, it may take many months or even years for a new user or service to obtain a frequency assignment in high-density locations. Using CR to find and utilize spectrum that is underutilized at a specific time and location would allow the introduction of new users or services without significant delay. This case is similar to selection of TV whitespaces.

Use case analysis

- Spectrum usage model and scenario classification
  - Long-term spectrum usage::1:N
- Sensing requirements
  - Traffic sensing capability
  - Sensing of white space
  - Sensing to identify service channels
  - Check with DA
  - Sense and vacate immediately if a primary signal is detected
- Usage of interface

  - CE–S, DA–S, S–S
- Key sensing information
  - Traffic information (resource allocation ratio)
  - Service channel ID

The requirements are similar to SPOLD3.

A.1.1.6 Worldwide mobility (SPOLD6)

Use case

Because radio frequency assignments are not harmonized around the world, a traveler may need to change the frequency of operation when moving from country to country. A CR radio can do this automatically, thus avoiding missed calls in the event that the user forgets to select appropriate frequencies upon arrival.

Use case analysis

- Spectrum usage model and scenario classification
  - Long-term spectrum usage::1:1
- Sensing requirements
  - Traffic sensing capability
  - Sense the most suitable channel first
  - Sensing to identify service channels
- Usage of interface
  - CE–S, CE–DA
- Key sensing information
  - Traffic information (resource allocation ratio)
  - Service channel (RATs) ID

The requirements are the same as for SPOLD2.

A.1.2 SPOSD

In these usage models, the CR is operating as a secondary user in an ad hoc manner, for a relatively short duration of time.

A.1.2.1 Tamper-resistant services (SPOSD1)

Use case

In an upscale apartment block, a common server is widely used to provide an Internet connection, as well as to store and manage video and Internet content that is being displayed on multiple monitors throughout each apartment. Each apartment operates its own IEEE 802.11n wireless network at near maximum capacity. All IEEE 802.11n channels are assigned to minimize interference between apartments and between buildings. In addition, building security uses event-triggered video cameras to monitor suspicious activities throughout the building. A CR-based IEEE 802.11n type connection reduces the risk of interruption of the security link due to the other IEEE 802.11n usage or by deliberate jamming. In this use case, the main intent is not for the priority access but to prevent the malicious access.

Use case analysis

- Spectrum usage model and scenario classification
  - Short-term spectrum usage::M:1
- Sensing requirements
  - Sensing of multiple channels simultaneously
  - Traffic sensing capability
  - Provide sensing information to multiple clients (negotiation to maintain normal operation of security devices).
- Usage of interface
  - CE–S, S–S
- Key sensing information
  - Client priority (Priority flag is set upon coordination of participating engines)
  - Duty cycle (to predict potential interference)
  - Traffic information (resource allocation ratio)

This is limited to WLAN. It is similar to both SPOLD2 (load sharing) and SPOLD4 (self-management).

A.1.2.2 Ad hoc licensee service—virtual spectrum coordination model (SPOSD2)

Use case

A licensed ad hoc secondary user is operating inside the primary service area. Other secondary users may be inside or outside of the primary service area. The licensed ad hoc secondary node takes the role of a virtual access point (AP)/BS when other secondary users want to connect to the IP network and acts as an interference controller for the primary service. The secondary system may not always follow the primary system's configuration (i.e., using different RATs). The ad hoc licensee node can be defined by either a virtual AP/BS controlled by the primary AP/BS or a self-controlled virtual AP/BS.

Use case analysis

- Spectrum usage model and scenario classification
  - Short-term spectrum usage::1:N
- Sensing requirements
  - Sense and vacate immediately when a primary user signal is detected
  - Sensing to identify service channels
  - Sensing information sharing with primary resource controller for validation of measurement results
  - Sensing to prevent interference
  - Check with DA
  - Sensing the most suitable channel list

  - Synchronization among sensors or radios
  - Sensing of multiple channels (used and unused white space)
  - Sensing of RAT
- Usage of interface
  - CE–S, S–S, DA–CE
- Key sensing information
  - Service channel (RATs) ID
  - Power level of primary signals
  - Policy information to enable underlay operation
  - Sensing method, techniques
  - Sensing duration (channel latency issue)
  - Detection threshold
  - Information to be sent to DA should be as complete as possible
  - Client ID

Two spectrum coordination methods are defined in this case. One possibility is to define it as a centric secondary spectrum coordination method via a mobile node, and a target spectrum for secondary spectrum utilization is selected by itself. The other is to define it as a spectrum sharing case controlled by a wired-network-node.

A.1.2.3 Ad hoc licensee service—underlay spectrum coordination model (SPOSD3)

Use case

This ad hoc networking model is composed of an ad hoc connection in the primary service area. Two kinds of secondary spectrum usage models may be assumed: first, a model where the ad hoc connection uses an underlay spectrum and where the received power of the primary signal at the ad hoc networking area is larger than the transmitted power of the secondary signal, and second, a model where the ad hoc connection uses a specific underlay spectrum and where the received power of the primary signal at the ad hoc networking area is smaller than the transmitted power of the secondary signal. These cases should assume perfect interference control for primary users, and the secondary spectrum coordination can be a centralized or a decentralized operation.

Use case analysis

- Spectrum usage model and scenario classification
  - Short-term spectrum usage::1:N
- Sensing requirements
  - Sense and vacate immediately if a primary signal is detected
  - Sensing to identify service channels
  - Sensing to identify primary system feature
  - Sensing to prevent interference
  - Sensing of RAT

- Usage of interface
  - CE–S, S–S
- Key sensing information
  - Service channel (RATs) ID
  - Sensing method, techniques
  - Sensing duration (channel latency issue)
  - Detection threshold
  - Client ID
  - Policy information to enable underlay operation

The requirements are similar to those of spectrum sharing (SPOLD3) but in short-term fashion. It is an underlay system. The power level of primary signals can be used to indicate the presence or absence of primary signals so that the secondary user can vacate immediately when a primary user signal is detected.

A.1.2.4 Self-management of uncoordinated spectrum (SPOSD4)

Use case

In the unlicensed bands, and to a lesser extent in some licensed bands, the high density of privately owned networks together with freedom-of-movement requirements for certain transitory operations makes central coordination of frequency assignments highly impractical. CR provides an effective means of self-coordination so as to avoid interference with other networks while providing useful throughput.

Use case analysis

- Spectrum usage model and scenario classification
  - Short-term spectrum usage::M:1
- Sensing requirements
  - Sensing of multiple channels (used and unused white space)
  - Sensing to prevent interference
  - Provide sensing information to multiple clients
- Usage of interface
  - CE–S, CE–CE, DA–S
- Key sensing information
  - Client priority (Priority flag is set upon coordination of participating engines)
  - Duty cycle (to predict potential interference)

This use case is similar to SPOLD4.

A.1.3 Service enhanced models (SEM)

These CR usage models look into assisting CR technology in terms of service enhancements. The first four models look into applying CR technology to enhance the communications capabilities of public safety responders. These use cases are in line with the identifications of SDR Forum relating to the Scenario of

7/7 bombing of the London underground (see SDR Forum [B9]). The fifth model is related to assisting CR technology for the investigation of policy violation.

A.1.3.1 Network extension (SEM1)

Use case

For coverage extension, CR capabilities can be used to reconfigure radios automatically to include a repeater capability to extend network coverage to areas where radios are otherwise cut off from their infrastructure, particularly during initial response to an incident prior to additional communications resources being deployed.

Use case analysis

- Spectrum usage model and scenario classification
  - Short-term spectrum usage::1:N
- Sensing requirements
  - Sense and vacate immediately if a primary signal is detected
  - Sensing to identify service channels
- Usage of interface
  - CE–S, S–S, DA–CE
- Key sensing information
  - Service channel (RATs) ID
  - Power level of primary signals
  - Policy information to enable underlay operation

The requirements are the same as for SPOSD2.

A.1.3.2 Dynamic access additional spectrum (SEM2)

Use case

At several points in the London bombing scenario, there were communications difficulties because of the sheer volume of calls on the voice communications networks. Dynamic spectrum access, or the ability for CRs to identify unused or underutilized spectrum, can be a solution in this scenario and provide a means for expanding capacity when needed. In this case, the objective is to create an additional capacity for emergency personnel to communicate.

Use case analysis

- A combination of load sharing (SPOLD2) and spectrum sharing (SPOLD3) use cases

A.1.3.3 Temporary reconfiguration (SEM3)

Use case

In Temporarily Reconfigure First Responder Communication Device Priorities use case, CRs (referring to cell phones) might be able to be temporarily reconfigured with higher priorities based on the circumstances of the emergency responder.

Use case analysis

- Sensing is not required

A.1.3.4 Interface to non-first responders (SEM4)

Use case

CRs can allow non-first responders communication access to first responders in specific situations in which the non-first responders are actively participating in the response while avoiding impact on mission-critical public safety networks.

Use case analysis

- Spectrum usage model and scenario classification
  - Short-term spectrum usage::M:1
- Sensing requirements
  - Reconfigurable protection to public safety network
  - Sensing of multiple channels simultaneously
  - Provide sensing information to multiple clients
- Usage of interface
  - CE–S, CE–CE
- Key sensing information
  - Client ID
  - Detection threshold
  - Sensing model
  - Sensing duration

No distinct difference between the first responder and the non-first responder. No unique sensing requirements exist.

A.1.3.5 Policy investigation (SEM5)

Use case

This use case focuses on the usage of spectrum sensing to assist investigation of policy violations in CR usage. Spectrum sensing functionalities can be deployed for policy verifications as in the case of rouge radio identification.

Use case analysis

- Spectrum usage model and scenario classification
  - Short-term spectrum usage::1:M
- Sensing requirements
  - As complete as possible
  - On regular basis

- Usage of interface
  - DA–S, S–S
- Key sensing information
  - Information to be sent to DA should be as complete as possible (RAT or modulation, geolocation, time stamp, operator ID, signal level, center frequency, bandwidth, Sensing methods, etc.).

The DA plays an important role in this use case. Because the goal of this use case is to analyze spectrum usage for policy investigation, information on spectrum usage should be as complete as possible for usage analysis. Furthermore, information should be collected regularly.

A.1.3.6 An ad hoc network model composed of multimode virtual AP/BS in several primary service areas (SEM6)

Use case

This model is an extension model of SPOSD2. There is a virtual AP/BS in the overlapping area of at least two kinds of different primary services. The licensed ad hoc node may select a link that has a better performance or lower traffic from the connectable primary services when other secondary users want to connect to an IP network. Furthermore, the node should be able to control interference to primary services. Other secondary users may be inside or outside of these primary service areas. The other assumptions are the same as the SPOSD2.

Use case analysis

- Spectrum usage model and scenario classification
  - Short-term spectrum usage::1:N
  - Sensing requirements
  - Traffic sensing capability
  - Sensing identify service channels
  - Sensing of multiple RATs
  - Sensing information sharing with primary resource controller for validation of measurement results
  - Sensing to prevent interference
  - Check with DA
  - Sensing the most suitable channel list
  - Synchronization among sensors or radios
  - Sense and vacate immediately if a primary signal is detected
  - Sensing of multiple channels (used and unused white space)
- Usage of interface
  - CE–S, S–S
- Key sensing information
  - Traffic information (resource allocation ratio)
  - Service channel (RATs) ID

- Sensing method, techniques
- Sensing duration (channel latency issue)
- Detection threshold
- Information to be sent to DA should be as complete as possible
- Client ID

The end effect is load sharing, but the use case looks into a different implementation scenario.

A.1.4 IEEE 1900.4-enabled dynamic spectrum access use cases

IEEE Std 1900.4™-2009 considers a heterogeneous wireless environment shown in Figure A.1. Such a heterogeneous wireless environment may include multiple operators, multiple radio access networks (RANs), multiple RATs, or multiple terminals.

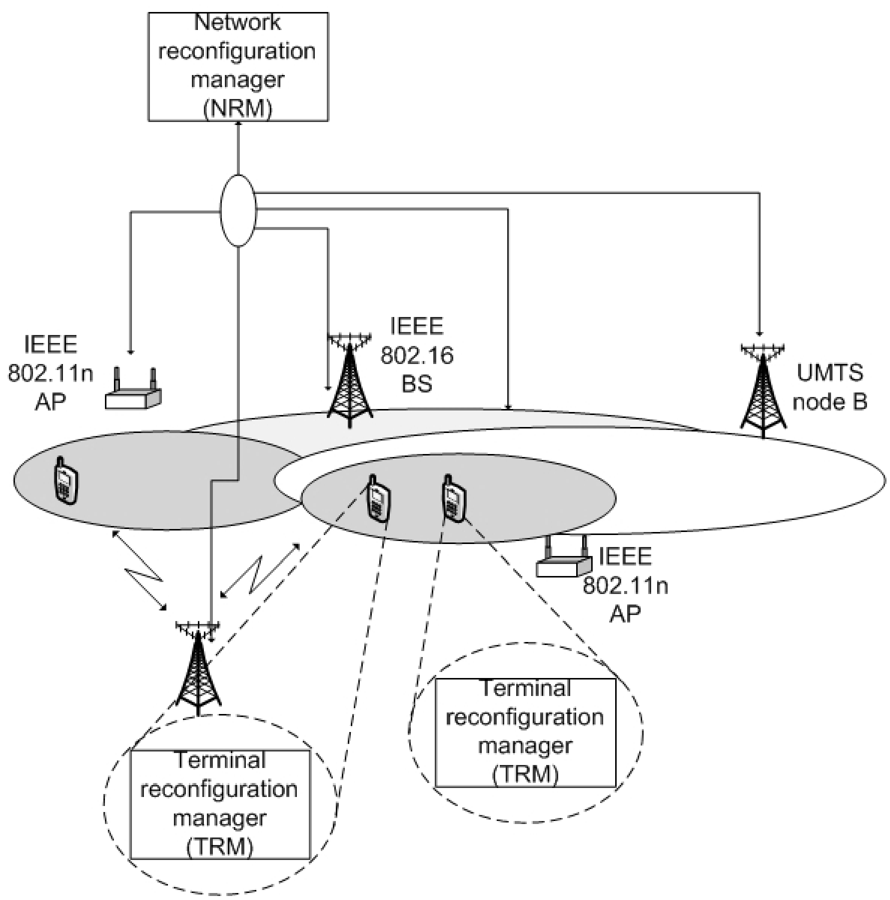


Figure A.1—Heterogeneous wireless environment considered in IEEE Std 1900.4-2009

Within Figure A.1:

Network Reconfiguration Manager (NRM) is the entity that manages the composite wireless network and terminals in terms of network-terminal distributed optimization of spectrum usage. This management is done within the regulatory framework and in a manner consistent with available context information.

Terminal Reconfiguration Manager (TRM) is the entity that manages the terminal in terms of network-terminal distributed optimization of spectrum usage. This management is done within the framework of radio resource selection policies conveyed by the NRM and in a manner consistent with the user's preferences and the available context information.

For this heterogeneous wireless environment, IEEE Std 1900.4-2009 defines three use cases representing different types of dynamic spectrum access (see Figure A.2):

— Dynamic spectrum assignment
— Dynamic spectrum sharing
— Distributed radio resource usage optimization

NOTE—NRM and TRM may require sensing-related information for resource management.

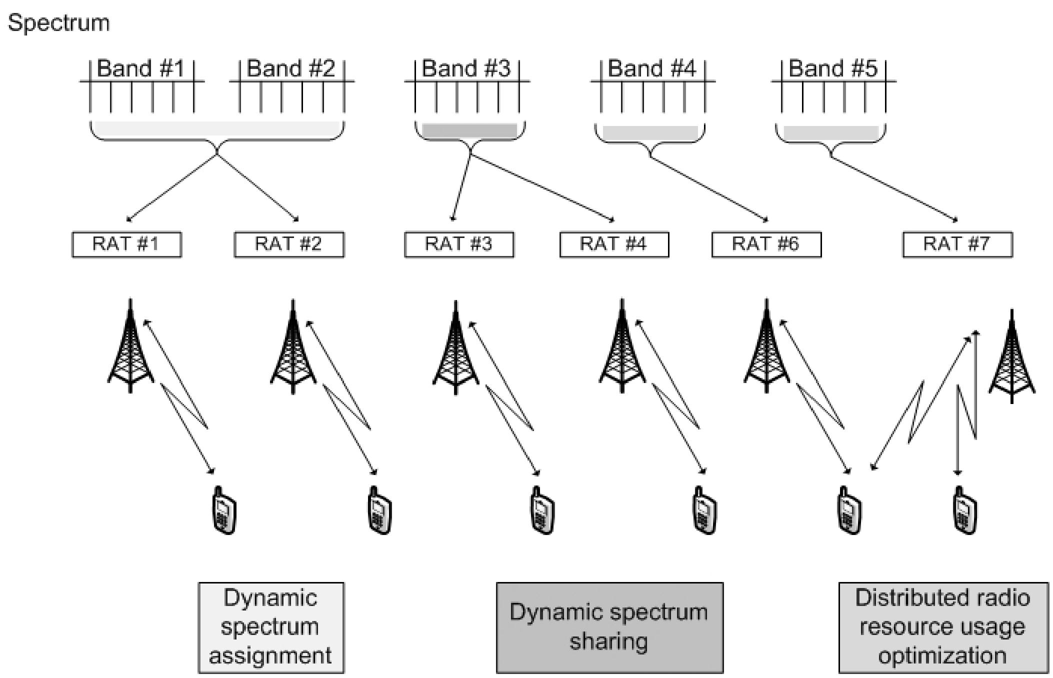


Figure A.2— IEEE 1900.4-2009 use cases

A.1.4.1 Dynamic spectrum assignment (DS Assignment)

Use case

In the dynamic spectrum assignment (DS Assignment) use case, frequency bands are dynamically assigned to the RANs in order to optimize radio resource usage and improve quality of service (see Figure A.2).

A short summary of the DS Assignment use case is as follows:

— The NRM analyzes available context information and dynamically makes spectrum assignment decisions.
— After the new spectrum assignment decisions have been made, the NRM requests corresponding reconfiguration of its RANs.
— After the RANs reconfiguration, terminals may need to reconfigure correspondingly.

Decision making for DS Assignment requires context information, which could include spectrum sensing information.

Use case analysis

- Spectrum usage model and scenario classification
  - Long-term spectrum usage::1:N
- Sensing requirements
  - Traffic sensing capability
  - Sensing of service channel
  - Sensing of multiple channels
- Usage of interface
  - CE–S, CE–DA, S–S
- Key sensing information
- Resource allocation ratio
- Service channel

This is a general use case of CR.

A.1.4.2 Dynamic spectrum sharing (DSS)

Use case

In the DSS use case, frequency bands assigned to RANs are fixed. However, a particular frequency band can be shared by several RANs to optimize radio resource usage and improve quality of service (see Figure A.2).

A short summary of the dynamic spectrum assignment use case is as follows:

- NRM analyzes available context information and dynamically makes spectrum access decisions.
- After these decisions, NRM requests corresponding reconfiguration of its RANs.
- NRM dynamically generates radio resource selection policies and sends them to its TRMs. These radio resource selection policies will guide these TRMs in their spectrum access decisions.
- TRMs analyze these radio resource selection policies and the available context information and dynamically make spectrum access decisions. These spectrum access decisions are made within the framework of the radio resource selection policies.
- After these decisions, each TRM requests corresponding reconfiguration of its terminal.

The DSS use case can also include the case of primary/secondary spectrum access.

Decision making for DSS requires context information, which could include spectrum sensing information.

Use case analysis

- Spectrum usage model and scenario classification
  - Long-term spectrum usage::M:1
- Sensing requirements
  - Traffic sensing capability

  - Sensing of white space
  - Sensing of service channel
  - Check with DA
- Usage of interface
  - CE–S, CE–DA, S–S
- Key sensing information
  - Traffic information (resource allocation ratio)
  - Service channel ID

This use case is similar to SPOLD3 in terms of sensing requirements.

A.1.4.3 Distributed radio resource usage optimization (DRRUO)

Use case

In the distributed radio resource usage optimization (DRRUO) use case, frequency bands assigned to RANs are fixed. Also, reconfiguration of RANs is not considered in this use case. Instead, reconfigurable terminals with or without multihoming capability are considered (see Figure A.2).

A short summary of the DRRUO use case is as follows:

- NRM analyzes available context information, dynamically generates radio resource selection policies, and sends them to its TRMs. These radio resource selection policies will guide these TRMs in their reconfiguration decisions.
- TRMs analyze radio resource selection policies and available context information and dynamically make decisions on reconfiguration of their terminals to improve radio resource usage and quality of service.
- Following these decisions, TRMs request corresponding reconfiguration of their terminals.

Decision making for DRRUO requires context information, which may include spectrum sensing information.

Use case analysis

- Spectrum usage model and scenario classification
  - Long-term spectrum usage::1:N
- Sensing requirements
  - Traffic sensing capability
  - Sensing to identify service channels
  - Sensing of multiple channels
- Usage of interface
  - CE–S, CE–DA, S–S
- Key sensing information
  - Resource allocation ratio
  - Service channel IDs

The key feature of this use case is that the terminals can perform spectrum usage optimization. Therefore, multiple sensors can be used to sense as many channels as possible.

A.2 Perspective of spectrum sensing

In A.1, the use cases are given from the perspective of spectrum usage. This annex introduces use cases from the perspective of how the sensing information is used. Each use case is analyzed, and the requirements are extracted.

A.2.1 Distributed sensing models (DSMs)

Multiple sensing devices (sensors) that are located at different physical positions in a service area can fulfill the sensing functionality of a radio system either independently or in a coordinated manner. Note that this is different from distributed processing. These usage models look into the CR technology in which the sensing information is obtained from spatially distributed sensors.

A.2.1.1 CR system with distributed sensors (DSM1)

Use case

The CR system analyzes spectrum usage and makes spectrum access decision based on the sensing-related information provided by sensors that are located at different physical positions (i.e., distributed sensing [cf. 3.1]). This type of sensing usage can be found in applications such as using CR to increase the capacity of communication for commercial purpose in hot-spot areas such as a university campus or a central business district area in a city. This type of sensing usage can also be found in SEM2, but the purpose and priority levels are different.

A CR terminal (such as a cognitive mobile phone) can initiate the sensing function of the sensing network once it enters the service area of the sensor network. If the CR terminal has an embedded sensing functional block, it can use the sensing information from the sensor network to enhance its sensing capability. This would ultimately help DSA decision making.

In another example, multiple CR nodes form a radio network, such as a CR-based ad hoc/multihop network for building up an efficient communication. Some of the sensors in individual nodes can form a virtual sensor network. Furthermore, these nodes can use the sensing information from this virtual sensor networks to analyze spectrum usage and make the decision of spectrum access.

Distributed sensing can be further classified into peer-to-peer sensing, cooperative sensing, collaborative sensing, or selective sensing based on the way the sensing-related information is conveyed from sensors to the CE. Here, sensing channel denotes the frequency channel on which the sensors perform spectrum measurement, and sensor channel denotes the frequency channel through which distributed sensors exchange sensing-related information. Note that the sensing channel and sensor channel may be of different frequencies, types, and media.

Peer-to-peer sensing

This is a special case where closely located CRs can negotiate and exchange sensing-related information to improve the sensing quality at each CR terminal. This case does not require multiple access schemes, whereas in collaborative and cooperative schemes, multiple access schemes should be provided for exchanging sensing-related information among multiple sensors in the sensor channel. Initial negotiation between the peer radios to exchange the sensing-related information can be carried out through handshaking.

Cooperative sensing

In this case, multiple sensors report sensing-related information independently to a CE. Such local processing can be performed in hybrid fashion. When multiple local processed data are received, the CE still needs to perform data fusion operations. This scenario is a combination of the examples shown in Figure A.3. The CE collects and combines the information from the sensor network to obtain a resultant output. The different sensors may provide information on different frequency bands, or the same frequency but collect data from a different location. The resultant output is an aggregate of information based on the information from individual sensors, as well as the underlying combining algorithm or data fusion.

Collaborative sensing

In this case, multiple sensors exchange sensing-related information to perform local processing before providing the sensing-related information as well as sensing results to the CE. To achieve this, the sensors should have a common goal and should provide similar sensing information.

Selective sensing

In this case, one sensor is selected to perform sensing. The sensing information is sent to a CR. Selection of sensors can be performed on the basis of a priori information regarding to the quality of the sensor channels, sensors' previous performance, energy level, etc. One of the benefits of this scheme is that only one sensor is active during sensing period and the others are in standby mode. This reduces the energy consumption of these sensors.

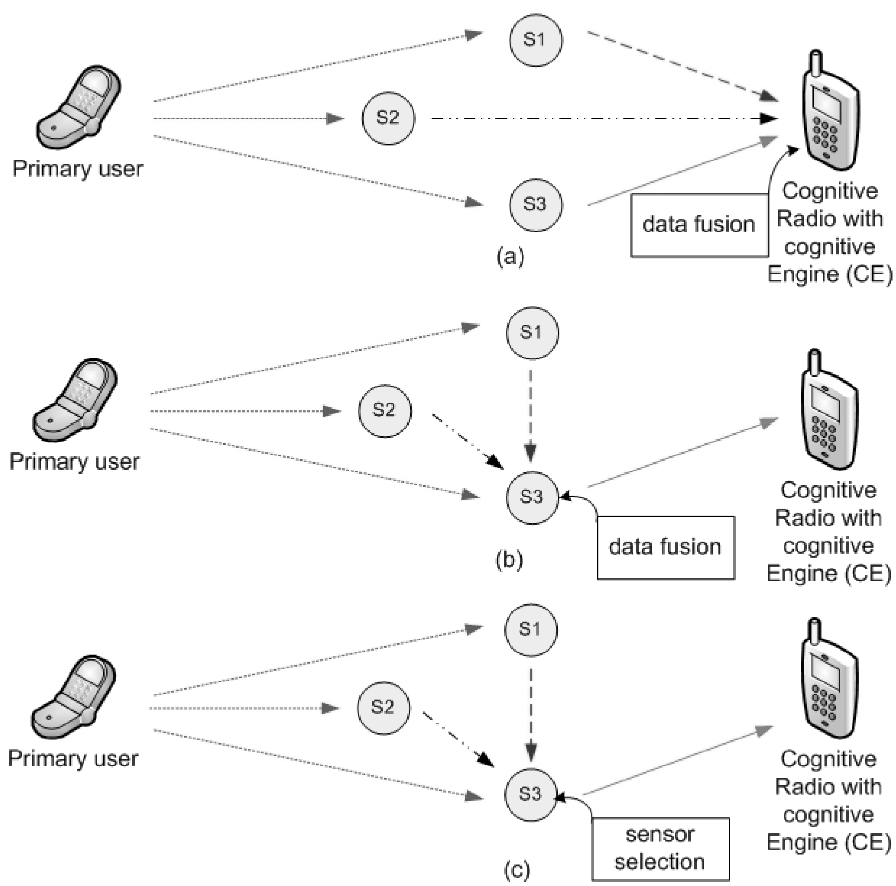


Figure A.3—Examples of distributed sensing with three spectrum sensors: (a) cooperative sensing, (b) collaborative sensing, and (c) selective sensing

Use case analysis

- Spectrum usage model and scenario classification
  - Short-term spectrum usage::1:N
- Sensing requirements
  - Sensing of multiple channels simultaneously
  - Synchronization among sensors or radios
- Usage of interface
  - CE–S, S–S
- Key sensing information
  - Sensor information (ID, geolocation, battery, specification, etc.)

This is a unique case for distributed sensing (cooperative sensing, collaborative sensing, and selective sensing). Because multiple sensors exchange sensing information, their IDs should be known. Synchronization among sensors or radios is needed for sensing information exchange. Note that this use case is applicable at both the network side and the terminal.

A.2.1.2 Peer-to-peer CRs (DSM2)

Use case

In this scenario, at least one terminal has a built-in spectrum sensor. When such terminals are located in close range and wish to establish a communication link, each terminal can use its own sensor (if it has a sensor) or the sensor in the other terminal can perform sensing and decide spectrum usage. Note that in the latter case, the sensor is distributed to a location that is different from that of the terminal. In the situation when both terminals have built-in sensors, the sensors can exchange sensing-related information for better exploitation of spectrum opportunity. Furthermore, the peer radios can also exploit any relevant sensing information from other nonpeer radios in their vicinity to enhance their sensing information.

This kind of use of sensing scenario can be utilized in the following types of applications.

Power efficient communication

In most of the existing communication systems, mobile terminals communicate with other mobile terminals through a BS with other mobile terminals. There are many situations where two mobile terminals are located very close to each other and communication via BSs becomes power inefficient. In this situation, the power of the mobile terminal can be efficiently used by setting up peer-to-peer communication. Otherwise, a large amount of power is wasted for transmitting signals to BSs.

Short-range data transfer

This application includes short-range data transfer (e.g., a cordless projector using the ISM band). Spectrum sensing can be used to identify a frequency band appropriate for the data transfer.

Cognitive wireless network

In a cognitive wireless network, multiple CR terminals form a wireless ad hoc network or mesh network with multiple peer-to-peer links. Each terminal in the peer-to-peer link detects the behavior of the spectrum occupancy in multiple channels and provides efficient communication links and paths. Peers also can exploit multiple channels at a time through different interfaces to form multichannel mesh networks. The sensing-related information from a sensor at one terminal can be shared by multiple terminals in a wireless network.

As an example, consider a simple cognitive relay network consisting of nodes A, B, and C. A relay can be set up on the basis of dynamic spectrum access between any two CR terminals, such as the service model SEM1. Data are then relayed from node A to node B and then to node C. The link between A and B can be considered as a peer-to-peer communication, which can be set up relying on the sensing results at node A and node B. The same goes to nodes B and C. Note that the example given here is to show where distributed sensing can be used for peer-to-peer communication. It is not related to the routing protocol.

Use case analysis

- Spectrum usage model and scenario classification
  - Short-term spectrum usage::M:1
- Sensing requirements
  - Sensing of multiple channels simultaneously
  - Sense to avoid interference
  - Synchronization among radios
- Usage of interface
  - CE–S, CE–CE
- Key sensing information
  - Client ID
  - Duty cycle

This is a use case for the cooperation among radios where the CE–CE interface is needed. Furthermore, the CE is equally distributed between two radios. CRs should sense to avoid interference with each other. Multiple CEs can access the same sensors. Synchronization among sensors or radios is needed for sensing information exchange.

A.2.2 Sensing enhanced models (SeEM)

These usage models look into applications where enhanced sensing function can be provided to the CR terminals.

A.2.2.1 Faulty sensing prevention (SeEM1)

Use case

Some sensors located in a blocked area might fail to detect the presence of the primary users. The CR terminals relying on these sensors might make the wrong decision of spectrum access and thus interfere with the primary users. Using enhanced sensing, the cognitive terminal can be provided with the most accurate sensing information to make a spectrum access decision. In this sense, the application prevents faulty sensing information by switching off the faulty sensor.

The reasons for a sensor producing faulty sensing information are many. For example, it might be due to the hidden terminal problem, such as the application of CR technologies in a highly scattering environment, including an office or urban area. Buildings in an urban area or office partitions, desk, closets, etc., prevent the line-of-sight propagation channel between primary users and sensors.

Use case analysis

- Spectrum usage model and scenario classification
  - Short-term spectrum usage::1:N
- Sensing requirements
  - Provide sensing information with performance measure
- Usage of interface
  - CE–S, S–S
- Key sensing information
  - Sensor ID
  - Sensing methods
  - Sensing performance information (e.g., confidence level)

This is a use case where sensors should provide their sensing information with Sensor ID, information regarding sensing methods, as well as sensing confidence level so that those sensors that providing wrong sensing information can be identified.

A.2.2.2 Power constrained sensing (SeEM2)

Use case

There are many applications where a sensor(s) is battery powered. It is desirable to increase the service life of these sensors to account for longer duration spectrum sensing. The most efficient way to reduce power consumption is only to turn on a sensor when necessary. This way of using sensing could be applied to applications described in use case such SPOSD1.

Use case analysis

- Spectrum usage model and scenario classification
  - Short-term spectrum usage::1:1
- Sensing requirements
  - Provide sensing information with power consumption
  - Information exchange method should be chosen for low power consumption
- Usage of interface
  - CE–S
- Key sensing information
  - Sensor ID
  - Sensor power consumption
  - Reporting mode (soft or hard information)

Sensors should provide sensing information with limited power consumption. Furthermore, different ways of exchanging sensing information have different power requirements. For example, sending soft information takes a relatively longer time and more power than sending hard information. In power constrained sensing, power-efficient ways of exchanging sensing information should be chosen.

A.2.2.3 Priority based sensing information provision (SeEM3)

Use case

Multiple clients may try to access the sensing-related information from a resource (e.g., Sensor and DA) at the same time. Therefore, competition might occur among these clients, and a priority scheme may be introduced. The client that has a higher priority level can access the sensing-related information first. This way of using sensing information reduces system overhead for transporting sensing-related information to multiple clients.

Use case analysis

- Spectrum usage model and scenario classification
  - Short-term spectrum usage::M:1
- Sensing requirements
  - Provide sensing information to multiple clients
- Usage of interface
  - CE–S, CE–CE
- Key sensing information
  - Client ID
  - Client Priority

The priority levels of sensors' clients should be identified when there are multiple clients. This priority characteristic can also be applied at sensor side. The use case is applicable to all M:1 and 1:M scenarios.

A.2.3 Sensing models for spectrum coordination

This subclause identifies sensing models for spectrum coordination systems. First, two spectrum coordination models are defined. One is the network node spectrum coordination model (NCM), where the final decision-making function of the secondary spectrum usage is in a single or multiple network nodes. The other is the mobile node spectrum coordination model (MCM), where the final decision-making function of the secondary spectrum usage is in a single or multiple terminal nodes. Second, on the basis of the spectrum coordination models, several sensing models are defined as shown in Figure A.4 through Figure A.6. Figure A.4 shows the sensing models of NCM, and Figure A.5 shows the sensing models for MCM. Furthermore, Figure A.6 shows extension models of NCM and MCM.

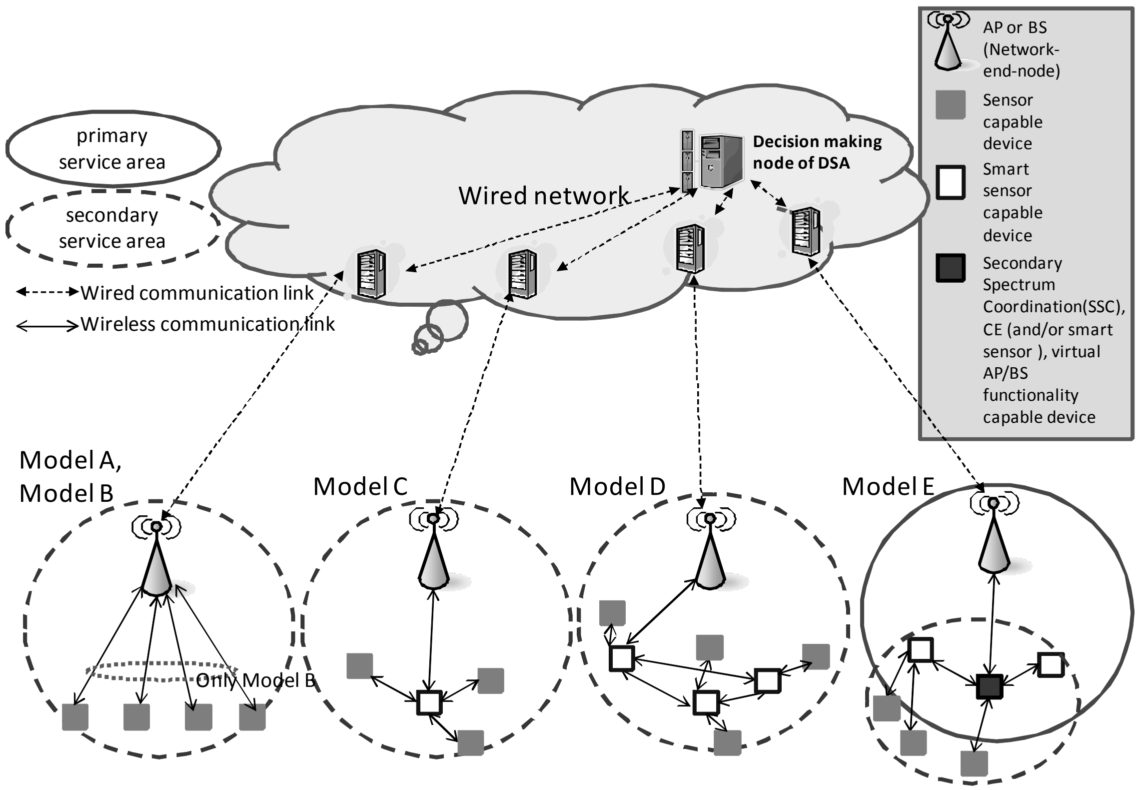


Figure A.4—Sensing models of NCM

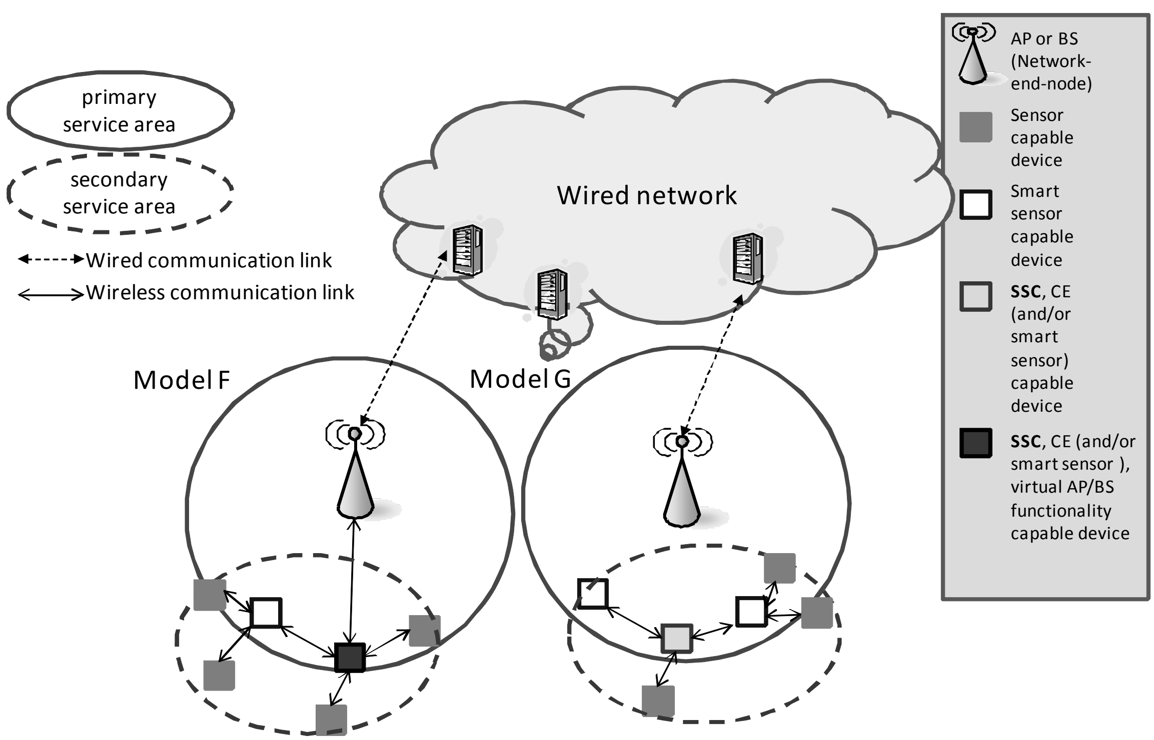


Figure A.5—Sensing models of MCM

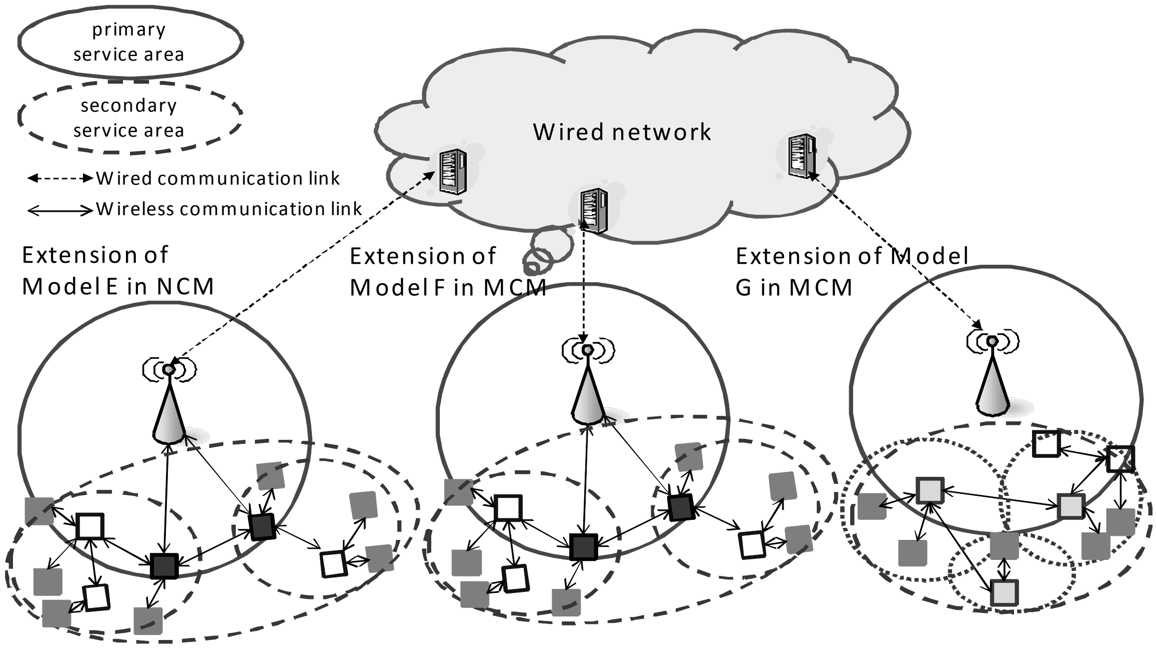


Figure A.6—Extension of model E in NCM and model F and G in MCM

A.2.3.1 Network node spectrum coordination model

Sensing models based on NCM are introduced in Figure A.4. The virtual AP/BS acts as a repeater for the secondary users when they want to connect to the IP network. The smart sensor can combine sensing-related information from multiple sensors and can relay the information to others. The Secondary Spectrum Coordination can make the decision for dynamic spectrum usage, or it can make the decision for spectrum usage through the permission of a spectrum coordinator node in a network.

A.2.3.1.1 Sensing model A: A sensing model for an interface definition in a node (NCM-A)

NCM-A defines an internal interface between the internal functional blocks in each node, which may come from different manufacturers.

The following points need to be considered:

- Sensing/Control information exchange between the smart sensor and the sensors
- Sensing/Control information exchange between the CE and the smart sensor
- Sensing/Control information exchange between the CE and the sensors

A.2.3.1.2 Sensing model B: A sensing model between the CE in a network-node and the sensors (NCM-B)

Use case

This is a distributed sensing model in a network centralized coordination system such as IEEE P802.22 [B2]. Each sensor informs its measurement results to the coordinator.

The following points need to be considered:

- Sensing/Control information exchange between the network-node and the sensors
- Specific points that need to be considered for this model may be summarized as follows:
  - The client can easily download the information about the spectrum usage environment based on geographical information via a wired link.
  - The volume and the number of elements of the sensing information that needs to be shared between the network-end-node and the sensors may be the smallest among all the models.
  - If the client needs just the received power of the primary signal, the information may be hard information (one bit) produced based on a regulation determined level.

Use case analysis

- Spectrum usage model and scenario classification
  - Long-term spectrum usage::1:N
- Sensing requirements
  - Sensing of multiple channels
  - Sensing of white space
  - Check with DA
- Usage of interface
  - CE–S, S–S, DA–CE
- Key sensing information
  - Sensor information (to distinguish sensor *vs.* smart sensor)
  - Sensing method, techniques
  - Sensing duration (channel latency issue)
  - Detection threshold
  - Reporting mode
  - Client ID

These use cases employ distributed sensing. The CE is located at the network side.

A.2.3.1.3 Sensing model C: A sensing model between the smart sensor and the sensors (NCM-C)

Use case

In this model, each sensor transmits the measurement results to a smart sensor. The smart sensor then combines the results and sends the merged sensing information to the coordinator.

The following points need to be considered:

- Sensing/Control information exchange between the smart sensor and the sensors

- Sensing/Control information exchange between the smart sensor and the CE in wired network/node
- Specific points that need to be considered for this model may be summarized as follows:
  - The volume and the number of elements of the sensing information that needs to be shared between the smart sensor and the sensors may be the smallest among all the models.
  - The control information between the smart sensor and the sensors should be well considered.
  - The control information between the smart sensor and the CE in wired-network/fixed-node should be well considered.
  - The other features may be the same as those of sensing model B.

Use case analysis

- Spectrum usage model and scenario classification
  - Long-term spectrum usage::1:N
- Sensing requirements
  - Sensing of multiple channels
  - Sensing of white space
  - Check with DA
- Usage of interface
  - CE–S, S–S, DA–CE
- Key sensing information
  - Sensor information (to distinguish sensor *vs.* smart sensor)
  - Sensing method, techniques
  - Sensing duration (channel latency issue)
  - Detection threshold
  - Reporting mode
  - Client ID

These use cases employ distributed sensing. The CE is located at the network side.

A.2.3.1.4 Sensing model D: A sensing model between smart sensors (NCM-D)

Use case

This model has smart sensors that collect the sensing information and combine it into a merged result. This merged result is sent to the coordinator.

The following points need to be considered:

- Sensing/Control information exchange between the smart sensors
- Specific points that need to be considered for this model may be summarized as follows:

- The volume and the number of elements of the sensing information that needs to be shared between the smart sensors may be the smallest compared with the other models described.
- The control information between the smart sensors should be well considered.
- The other features may be the same as those of the sensing model B.

Use case analysis

- Spectrum usage model and scenario classification
  - Long-term spectrum usage::1:N
- Sensing requirements
  - Sensing of multiple channels
  - Sensing of white space
  - Check with DA
- Usage of interface
  - CE–S, S–S, DA–CE
- Key sensing information
  - Sensor information (to distinguish sensor vs. smart sensor)
  - Sensing method, techniques
  - Sensing duration (channel latency issue)
  - Detection threshold
  - Reporting mode
  - Client ID

These use cases employ distributed sensing. The CE is located at the network side.

A.2.3.1.5 Sensing model E: A sensing model of a secondary spectrum coordination with a primary AP/BS connection (NCM-E)

Use case

This model has a CE capable device in the primary service area, where the device is a secondary spectrum coordinator authorized by a spectrum coordinator node in a wired network. The device collects sensing information from several other devices and combines it into a merged result. This merged result is sent to the spectrum coordinator node. The device also acts as a virtual AP/BS when a secondary user wants to connect to the IP network.

The following points need to be considered:

- Sensing/Control information exchange between the CE and the smart sensors.
- Sensing/Control information exchange between the CE and the sensors.
- Specific points that need to be considered for this model may be summarized as follows:

- The client can easily download the information about the spectrum usage environment based on geographical information via a primary wireless link.
- The volume and the number of elements of the sensing information that needs to be shared between the smart sensor and the CE of the secondary spectrum coordinator may be relatively smaller than other models.
- The volume and the number of elements of the sensing information that needs to be shared between the CE of the secondary spectrum coordinator and the sensors may be relatively smaller than other models.
- The control information between the CE of the secondary spectrum coordinator and the sensors should be well considered.
- The control information between the CE of the secondary spectrum coordinator and the smart sensor should be also well considered.
- The other features may be the same as those of the sensing model D.

Use case analysis

- Spectrum usage model and scenario classification
  - Short-term spectrum usage::1:N
- Sensing requirements
  - Sensing of multiple channels
  - Sensing information integration
  - Sensing information sharing control
  - Sensing information sharing with primary resource controller for the validation of measurement results
  - Distributed device control
  - Check with DA
  - Sensing the most suitable channel list
  - Provide sensing information to multiple clients
- Usage of interface
  - CE–S, S–S, DA–CE
- Key sensing information
  - Sensor information (to distinguish sensor vs. smart sensor)
  - Target detection performance
  - Sensing method, techniques
  - Sensing duration
  - Client priority
  - Duty cycle
  - Detection threshold
  - Information to be sent to DA should be as complete as possible
  - Reporting mode
  - Client ID

These use cases employ distributed sensing. Furthermore, CEs are located at a wired-network-node and a mobile, respectively.

A.2.3.2 Mobile node spectrum coordination model

Figure A.5 introduces the sensing models based on MCM.

A.2.3.2.1 Sensing model F: A sensing model of a secondary spectrum coordination with a primary AP/BS connection (MCM-F)

Use case

This model has the CE-capable device acting as a permanent secondary spectrum coordinator operating in the primary service area. The difference between this model and model E is that in this model, the spectrum coordinator has a permanent spectrum usage right. The device collects and combines sensing information from several sensors. This merged result is transmitted to the coordinator. The device also acts as a virtual AP/BS when a secondary user wants to connect to the IP network.

The following points need to be considered:

- Sensing/Control information exchange between the smart sensor and the sensors with a primary AP/BS connection
- Sensing/Control information exchange between the CE and the sensors with a primary AP/BS connection
- Sensing/Control information exchange between the CE and the smart sensor with a primary AP/BS connection
- Specific points that need to be considered for this model may be summarized as follows:
  - The secondary spectrum coordinator can download the information about the spectrum usage environment based on geographical information via a primary wireless link, but the download may be difficult depending on the condition of the primary link.
  - The volume and number of elements of sensing information that needs to be shared between the smart sensor and the CE of the secondary spectrum coordinator may be relatively larger than other models.
  - The volume and the number of elements of sensing information to be shared between the CE of the secondary spectrum coordinator and the sensors may be relatively larger than other models.
  - The volume and the number of elements of sensing information to be shared between the smart sensor and the sensors may be relatively larger than other models.
  - The control information between the CE of the secondary spectrum coordinator and the sensors should be well considered.
  - The control information between the CE of the secondary spectrum coordinator and the smart sensor should be well considered.
  - The control information between the sensors and the smart sensor should be well considered.

Use case analysis

- Spectrum usage model and scenario classification
  - Long-term spectrum usage::1:N
- Sensing requirements
  - Sensing of multiple channels
  - Sensing information integration
  - Sensing information sharing control
  - Sensing information sharing with primary resource controller for validation of measurement results
  - Distributed device control
  - Sensing to prevent interference
  - Check with DA
  - Sensing the most suitable channel list
  - Synchronization among sensors or radios
  - Sensing to identify service channels
  - Sense and vacate immediately if a primary signal is detected
  - Provide sensing information to multiple clients
  - Sensing of multiple channels
  - Sensing of RAT
- Usage of interface
  - CE–S, S–S, CE–DA
- Key sensing information
  - Sensor information (to distinguish sensor *vs.* smart sensor)
  - Target detection performance
  - Sensing method, techniques
  - Sensing duration
  - Client priority
  - Duty cycle
  - Detection threshold
  - Information to be sent to DA should be as complete as possible
  - Reporting mode
  - Client ID

The spectrum coordinator has permanent spectrum usage. Distributed sensing is applied with CE located at the terminal side.

A.2.3.2.2 Sensing model G: A sensing model of a secondary spectrum coordination without a primary AP/BS connection (MCM-G)

Use case

In this model, the CE capable device, acting as a permanent secondary spectrum coordinator, is operating in the primary service area. The difference between this model and model F is that in this scenario, the secondary spectrum coordinator does not have the capability to act as a virtual AP/BS. The device collects and combines sensing information from several sensors into combined sensing information. This merged result is sent to the secondary spectrum coordinator.

The following points need to be considered:

- Sensing/Control information exchange between the smart sensor and the sensors
- Sensing/Control information exchange between the CE and the sensors
- Sensing/Control information exchange between the CE and the smart sensor
- Specific points that need to be considered for this model may be summarized as follows:
  - The secondary spectrum coordinator cannot download the information about the spectrum usage environment based on geographical information.
  - The volume and the number of elements of sensing information that needs to be shared between the smart sensor and the CE of the secondary spectrum coordinator may be the largest among all models.
  - The volume and number of elements of sensing information that needs to be shared between the CE of the secondary spectrum coordinator and the sensors may be the largest among all models.
  - The volume and number of elements of sensing information that needs to be shared between the smart sensor and the sensors may be the largest amongst all models.
  - The control information between the CE of the secondary spectrum coordinator and the sensors should be well considered.
  - The control information between the CE of the secondary spectrum coordinator and the smart sensor should be well considered.
  - The control information between the sensors and the smart sensor should be well considered.

Use case analysis

- Spectrum usage model and scenario classification
  - Long-term spectrum usage::1:N
- Sensing requirements
  - Sensing of multiple channels
  - Sensing to identify primary system feature
  - Sensing information integration
  - Sensing information sharing control
  - Distributed device control
  - Sensing to prevent interference

    - Synchronization among sensors or radios
    - Sensing to identify service channels
    - Sense and vacate immediately if a primary signal is detected
    - Provide sensing information to multiple clients
    - Sensing of multiple channels
    - Sensing of RAT
- Usage of interface
    - CE–S, S–S
- Key sensing information
    - Sensor information (to distinguish sensor *vs.* smart sensor)
    - Target detection performance
    - Sensing method, techniques
    - Sensing duration
    - Client priority
    - Duty cycle
    - Detection threshold
    - Information to be sent to DA should be as complete as possible
    - Reporting mode
    - Client ID
    - Policy information to enable underlay operation

Spectrum coordinator has permanent spectrum usage. Distributed sensing is applied with CE being located at the terminal side.

A.2.3.3 Extension models

The sensing models shown in Figure A.6 are described in this subclause and are based on extensions of NCM model E and extensions of MCM models F and G.

A.2.3.3.1 Extension model of sensing model E (NCM-E-extension)

Use case

In this model, the CE capable devices are located in or near the primary service area. The devices function as secondary spectrum coordinators authorized by a spectrum coordinator node in a wired network. The device also acts as a virtual AP/BS when a secondary user wants to connect to the IP network. This model may be essential for an ad hoc secondary spectrum usage case in NCM because the secondary service can extend the communication range more than the single network cooperated by secondary spectrum coordinators.

The following points need to be considered:

- Sensing/Control information exchange between CEs

- Sensing/Control information exchange between smart sensors
- Specific points that need to be considered for this model may be summarized as follows:
  - The clients can easily download the information about the spectrum usage environment based on geographical information via a primary wireless link.
  - The volume and the number of elements of the sensing information that needs to be shared between the smart sensors may be relatively smaller than other models.
  - The volume and the number of elements of the sensing information that needs to be shared between the CEs of the secondary spectrum coordinators may be relatively smaller than other models.
  - The control information between the CEs of the secondary spectrum coordinator should be well considered.
  - The control information between the smart sensors should be well considered.
  - The other features may be the same as those of the sensing model E.

Use case analysis

- Spectrum usage model and scenario classification
  - Short-term spectrum usage::M:1
- Sensing requirements
  - Providing sensing information to multiple clients
  - Sensing information integration
  - Sensing information sharing control
  - Sensing information sharing with primary resource controller for validation of measurement results
  - Distributed device control
  - Check with DA
  - Sensing the most suitable channel list
  - Provide sensing information to multiple clients
- Usage of interface
  - CE–S, CE–CE, CE–DA
- Key sensing information
  - Client ID
  - Client priority
  - Target detection performance
  - Sensing method, techniques
  - Sensing duration
  - Client priority
  - Duty cycle
  - Detection threshold

- Information to be sent to DA should be as complete as possible
- Sensor information
- Reporting mode

This use case extends NCM-E by using multiple CEs. Thus, CE–CE interface is required.

A.2.3.3.2 Extension model of sensing model F (MCM-F-extension)

Use case

In this model, the CE capable devices are located in or near the primary service area. The devices function as permanent secondary spectrum coordinators. The difference between this case and the extension model of sensing model E is that in this case the secondary spectrum coordinator has a permanent spectrum usage right. The device also acts as a virtual AP/BS when a secondary user wants to connect to the IP network. This model may be essential for an ad hoc secondary spectrum usage case in MCM because the secondary service coverage area is extended by allowing multiple individual networks that are operated by different secondary spectrum coordinators.

The following points need to be considered:

- Sensing/Control information exchange between CEs
- Sensing/Control information exchange between smart sensors
- Specific points that need to be considered for this model may be summarized as follows:
    - The secondary spectrum coordinators can download the information about the spectrum usage environment based on geographical information via a primary wireless link, but the download may be difficult depending on the condition of the primary link.
    - The volume and number of elements of sensing information that needs to be shared between the smart sensors may be relatively larger than other models.
    - The volume and the number of elements of sensing information to be shared between the CEs of the secondary spectrum coordinators may be relatively larger than other models.
    - The volume and the number of elements of sensing information to be shared between the smart sensor and the sensors may be relatively larger than other models.
    - The control information between the CEs of the secondary spectrum coordinator should be well considered.
    - The control information between the smart sensors should be well considered.
    - The other features may be the same as those for the sensing model F.

Use case analysis

- Spectrum usage model and scenario classification
    - Long-term spectrum usage::M:1
- Sensing requirements
- Providing sensing information to multiple clients
- Sensing to identify primary system feature
- Sensing information integration

- Sensing information sharing control
- Sensing information sharing with primary resource controller for validation of measurement results
- Distributed device control
- Sensing to prevent interference
- Check with DA
- Sensing the most suitable channel list
- Synchronization among sensors or radios
- Sensing to identify service channels
- Sense and vacate immediately if a primary signal is detected
- Provide sensing information to multiple clients
- Sensing of multiple channels
- Sensing of RAT
- Usage of interface
  - CE–S, CE–CE, CE–DA
- Key sensing information
  - Client ID
  - Client priority
  - Target detection performance
  - Sensing method, techniques
  - Sensing duration
  - Client priority
  - Duty cycle
  - Detection threshold
  - Information to be sent to DA should be as complete as possible
  - Sensor information
  - Reporting mode

MCM-F is extended by using multiple CEs.

A.2.3.3.3 Extension model of sensing model G (MCM-G-extension)

Use case

This model has the CE capable devices located in or near the primary service area. The devices function as permanent secondary spectrum coordinators. The difference from the extension model of sensing model F is that in this case, the secondary spectrum coordinator does not have the capability to act as a virtual AP/BS. This model may be essential for an ad hoc secondary spectrum usage case in MCM because the secondary service coverage area is extended by allowing multiple individual networks that are operated by different secondary spectrum coordinators.

The following points need to be considered:

- Sensing/Control information exchange between CEs
- Sensing/Control information exchange between smart sensors
- Specific points that need to be considered for this model may be summarized as follows:
  - The client cannot download the information about the spectrum usage environment based on geographical information.
  - Therefore, the volume and the number of elements of sensing information that needs to be shared between the smart sensor and the CE of the secondary spectrum coordinator may be the largest among all models.
  - Also, the volume and the number of elements of sensing information that needs to be shared between the CE of the secondary spectrum coordinator and the sensors may be the largest among all models.
  - Also, the volume and the number of elements of sensing information to be shared between the smart sensor and the sensors may be the largest among all models.
  - The control information between the CEs of the secondary spectrum coordinator should be well considered.
  - The control information between the smart sensors should be well considered.
  - The other features may be the same as those of the sensing model G.

Use case analysis

- Spectrum usage model and scenario classification
  - Long-term spectrum usage::M:1
- Sensing requirements
  - Providing sensing information to multiple clients
  - Sensing to identify primary system feature
  - Sensing information integration
  - Sensing information sharing control
  - Distributed device control
  - Sensing to prevent interference
  - Synchronization among sensors or radios
  - Sensing to identify service channels
  - Sense and vacate immediately if a primary signal is detected
  - Provide sensing information to multiple clients
  - Sensing of multiple channels
  - Sensing of RAT
- Usage of interface
  - CE–S, CE–CE

- Key sensing information
  - Client ID
  - Client priority
  - Target detection performance
  - Sensing method, techniques
  - Sensing duration
  - Client priority
  - Duty cycle
  - Detection threshold
  - Information to be sent to DA should be as complete as possible
  - Sensor information
  - Reporting mode
  - Policy information to provide underlay operation

MCM-G is extended by using multiple CEs.

Annex B

(informative)

Use case classification

Table B.1—Use case classification

Spectrum usage models	Scenarios of system model	Use cases and application examples
Long-term spectrum usage	1:1	Emergency services (SPOLD1) Load Sharing to reduce blocking at peak traffic times (SPOLD2)
	1:N	Sharing of spectrum to reduce blocking at peak traffic times (SPOLD3) NCM-B,NCM-C ,NCM-D, NCM-E, MCM-F, MCM-G IEEE 1900.4 Dynamic spectrum assignment (IEEE 1900.4-DSS)
	M:1	Self management of uncoordinated spectrum (SPOLD4) MCM-F-extension MCM-G-extension IEEE 1900.4 Dynamic spectrum assignment (IEEE 1900.4-DSS)
Short-term spectrum usage	1:1	Power constrained sensing (SeEM2)
	1:N	Policy Investigation (SEM5) Distributed sensing (DSM1) Faulty sensing prevention (SeEM1) NCM-E IEEE 1900.4 Distributed radio resource usage optimization (1900.4-DRRUO) IEEE 1900.4 Dynamic spectrum sharing (IEEE 1900.4-DSS)
	M:1	Peer-to-peer Communication (DSM2) Priority based sensing (SeEM3) NCM-E-Extension IEEE 1900.4 Distributed radio resource usage optimization (1990.4-DRRUO) IEEE 1900.4 Dynamic spectrum sharing (IEEE 1900.4-DSS)

Table B.2—Mapping between use cases and sensing requirements

Reconfigurable sensing performance	SPOLD1, SEM4
Sensing of multiple channels simultaneously	SPOLD1, SPOLD4, SPOSD1, SPOSD2,SPOSD4, SEM4, SEM6, IEEE 1900.4-DS Assignment, IEEE 1900.4-DRRUO, DSM1, DSM2, NCM-B, NCM-C, NCM-D, NCM-E, MCM-F(-extension), MCM-G(-extension)
Only information on signal occupancy can be sufficient (band is free or occupied)	SPOLD1
Sensing is performed only at initial stage (during or after the catastrophe)	SPOLD1
Sensing is performed by embedded sensors with in radios (if infrastructure and fixed sensors are destroyed) or temporary deployment of distributed sensors	SPOLD1
Sensing of RAT(s)	SPOSD2,SPOSD3, SEM6
Traffic sensing capability	SPOLD2, SPOLD3, SPOLD5, SPOLD6, SPOSD1, SEM6, IEEE 1900.4-DS Assignment, IEEE 1900.4-DSS, IEEE 1900.4-DRRUO
Sensing to identify service channels	SPOLD2, SPOLD3, SPOLD5, SPOLD6, SPOSD2, SPOSD3, SEM1, SEM6, IEEE 1900.4-DS Assignment, IEEE 1900.4-DSS, IEEE 1900.4-DRRUO, MCM-F(-extension)
Sense the most suitable channel first	SPOLD2, SPOLD6, SPOSD2, SEM6, NCM-E(-extension), MCM-F(-extension)
Sensing of white space	SPOLD3, SPOLD5, IEEE 1900.4-DSS, NCM-B, NCM-C, NCM-D
Check with DA	SPOLD3, SPOLD5, SPOSD2, SEM6, IEEE 1900.4-DSS, NCM-B, NCM-C, NCM-D, NCM-E(-extension), MCM-F(-extension),
Sense and vacate immediately if a primary signal is detected	SPOLD3, SPOLD5, SPOSD2, SPOSD3, SEM1, SEM6, MCM-F(-extension) , MCM-F(-extension), MCM-F(-extension)
Sensing to prevent interference	SPOLD4, SPOLD4, SPOSD2, SPOSD3, SEM6, DSM2, MCM-F(-extension), MCM-F(-extension)
Provide sensing information to multiple clients	SPOLD4, SPOSD1, SPOSD4, SEM4, SeEM3, NCM-E-extension, MCM-F-extension, MCM-G-extension
As complete as possible	SEM5
On regular basis	SEM5
Synchronization among sensors or radios	SPOSD2, SEM6, DSM1, DSM2, MCM-F(-extension), MCM-F(-extension)
Provide sensing information with performance measure	SeEM1,
Provide sensing information with power consumption	SeEM2
Information exchange method should be chosen for low power consumption	SeEM2
Sensing to identify primary system feature	SPOSD3, MCM-F(-extension), MCM-G(-extension)
Sensing information integration	NCM-E-extension, MCM-F-extension, MCM-G-extension, DSM1, DSM2
Sensing information sharing control	NCM-E-extension, MCM-F-extension, MCM-G-extension, DSM1, DSM2, SPOSD3
Sensing information sharing with primary resource controller for validation of measurement results	SPOSD2, SEM6, NCM-E(-extension), MCM-F(-extension)
Distributed device control	NCM C, D, E(-extension), MCM-F(-extension), MCM-G(+extension), DSM1, DSM2

Table B.3—Mapping between use cases and key sensing information

Target detection performance (Pd, Pf, etc.)	SPOLD1
Sensing method, techniques	SPOLD1, SPOSD2, SPOSD3, SEM6, NCM-B, NCM-C, NCM-D, NCM-E(-extension) , MCM-F(-extension),
Sensing duration (channel latency issue)	SPOLD1, SPOSD2, SPOSD3, SEM6, NCM-B, NCM-C, NCM-D, NCM-E(-extension) , MCM-F(-extension), MCM-F(-extension)
Traffic information (resource allocation ratio)	SPOLD2, SPOLD3, SPOLD5, SPOLD6, SPOSD1, SEM6, IEEE 1900.4-DS Assignment, IEEE 1900.4-DSS, IEEE 1900.4-DRRUO
Service channels (RATs) ID	SPOLD2, SPOLD3, SPOLD5, SPOLD6, SPOSD2, SPOSD3, SEM1, SEM6, IEEE 1900.4-DS Assignment, IEEE 1900.4-DSS, IEEE 1900.4-DRRUO
Client priority, ID	SPOLD4, SPOSD1, SPOSD4, SEM4, DSM2, SeEM3, NCM-E-extension, MCM-F-extension, MCM-G-extension
Duty cycle (to predict potential interference)	SPOLD4, SPOSD1, SPOSD4, DSM2, NCM-E(-extension) , MCM-F(-extension) , MCM-F(-extension)
Power level of primary signals	SPOSD2, SEM1
Policy information	SPOSD2, SEM1
Sensing mode, duration	SEM4
Information to be sent to DA should be as complete as possible (RAT or modulation, geolocation, timestamp, operator ID, signal level, center frequency, bandwidth, Sensing methods, etc.)	SEM5, SPOSD2, SEM6, NCM-E(-extension) , MCM-F(-extension),
Sensor information (ID, geolocation, battery, specification, etc.)	DSM1, DSM2, SeEM2, NCM-B, NCM-C, NCM-D, NCM-E, MCM-F, MCM-G
Sensing performance (confidence level)	SeEM1
Sensor power consumption	SeEM2
Reporting mode (soft or hard information)	SeEM2, NCM-B, NCM-C, NCM-D, NCM-E(-extension) , MCM-F(-extension) , MCM-G(-extension)
primary system feature	SPOSD3, MCM-F(-extension), MCM-G(-extension)
Sensing information integration	NCM-E-extension, MCM-F-extension, MCM-G-extension, DSM1, DSM2
Sensing information sharing control	NCM-E-extension, MCM-F-extension, MCM-G-extension, DSM1, DSM2, SPOSD3
Sensing information sharing with primary resource controller for validation of measurement results	SPOSD2, SEM6, NCM-E(-extension), MCM-F(-extension)
Distributed device control	NCM C, D, E(-extension), MCM-F(-extension), MCM-G(+extension), DSM1, DSM2

Table B.4—Spectrum sensing system model and its relation to IEEE 1900.6 interfaces

System models	Interfaces involved	Control information	Sensing-related information
1:1 scenario Single device spectrum sensing model (reference model)	CE/DA–S interface	Command + sensing parameters	Sensing + sensor information
1:N scenario Distributed sensors spectrum sensing model	CE/DA–S interface	Command + sensing parameters	Sensing + sensor information
	S–S interface	Command + sensing parameters	Sensing + sensor information
M:1 scenario Multiple CEs	CE–CE/DA interface	Command + sensing parameters	Sensing + sensor information
	CE/DA–S interface	Command + sensing parameters	Sensing + sensor information

Table B.5—IEEE 1900.6 interfaces and sensing-related information

Sensing-related information	CE–CE/DA network	CE–CE/DA terminal	CE/DA–S single sensor	CE/DA–S distributed sensors	S–S distributed sensors
Sensing control information	Sensing information request	Sensing information request	—Sensing information request —Measurement data request —Sensing duration —Termination .desired Pd or Pf —Types of sensing information — Priority control — Sensing information security level —Database information request — Client identification	—Sensing information request —Measurement data request —Sensing duration —Termination .desired Pd or Pf —Types of sensing information — Routing — Sensor gateway selection — Sensing information security level — Client identification — Location information	—Sensing information request —Measurement data request —Sensing duration —Termination —types of sensing information — Routing — Sensor gateway selection — Sensing information security level — Client identification
Sensor information	Not required	Not required	—Sensor logical ID —Battery status —Noise power —Sensor capability —Manufacturer ID	—Sensor logical ID —Battery status Noise power —Sensor capability —Manufacturer ID	—Sensor logical ID —Battery status —Noise power —Manufacturer ID —Types of sensing

			—Types of sensing technique —Measurement range —A/D, D/A resolution —Calibration data	—Types of sensing technique —Measurement range —A/D, D/A resolution —Calibration data	technique —Measurement range —A/D, D/A resolution —Calibration data
Sensing information	—Band occupancy —Frequency band —Confidence level	—Band occupancy —Frequency band —Confidence level	—Band occupancy —Signal energy —Confidence level —Frequency band —Bandwidth —Center frequency —Cyclic frequency — Location information —Local threshold	—Band occupancy —Signal energy —Confidence level —Frequency band —Bandwidth —Center frequency —Cyclic frequency — Location information —Local threshold	—Band occupancy —Signal energy —Confidence level —Frequency band —Bandwidth —Cyclic frequency — Location information —Local threshold

Table B.6—IEEE 1900.6 interfaces and locations of sensors

Sensing-related information	CE–CE/DA network	CE–CE/DA terminal	CE/DA–S single sensor	CE/DA–S distributed sensors	S–S distributed sensors
Sensor location	—Distributed sensors —Smart - sensors —Sensing capable device	—Distributed sensors —Smart - sensors —Sensing capable device	—Base station —AP —CR terminal	—Base station —AP —CR Terminal —Non-CR devices (personal digital assistant [PDA], smart phone, laptop, etc.) —Distributed standalone fixed sensors —Smart sensors	—Distributed standalone fixed sensors —Smart sensors

Annex C

(informative)

Implementation of distributed sensing

C.1.1 General description

Sensors and their clients can exchange sensing-related information through approved radio channels. The clients can be sensors and CE. The approved radio for sensing-related information exchange can be WiFi®,[10] ZigBee™,[11] UWB, etc.[12] This sensing-related information exchange is involved in distributed sensing models (DSMs) as described in Annex A.

C.1.2 Assumptions/requirement

For a given terminal or network node to communicate with multiple spectrum sensors, multiple access support is assumed on the client side. Furthermore, sensors are assumed to be equipped with onboard transceivers. In the case of distributed sensors spectrum sensing, sensors may need to execute directly sensing algorithms locally; hence, they need to be equipped with digital signal processing chip to perform sensing. Sensing algorithms may require estimating some parameters adaptively at certain interval such as noise power. Therefore, a memory chip is needed to store the information temporarily. Figure C.1 shows and example of the OSI model for sensing-related information exchange between a sensor and a client.

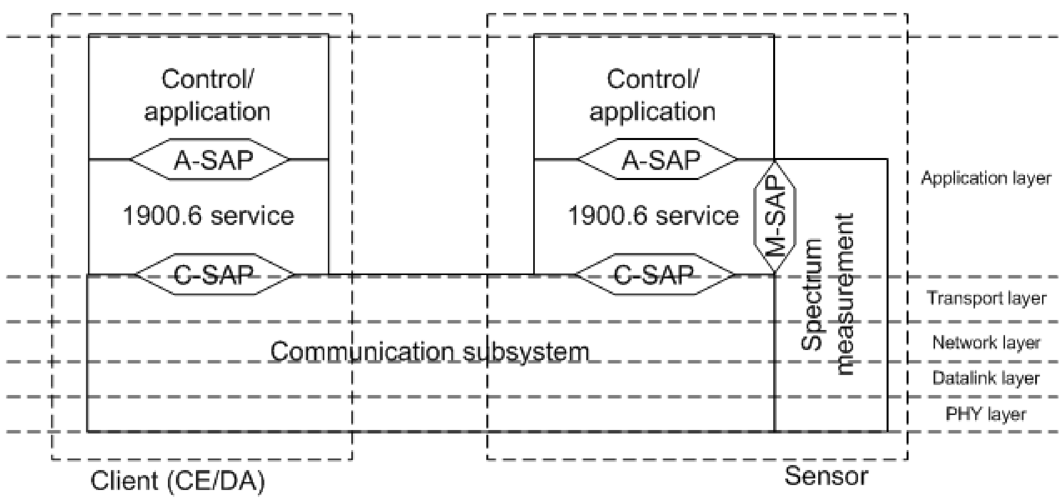


Figure C.1—Example OSI model for sensing-related information exchange between sensor and client

Figure C.2 shows the architecture of a sensor that can exchange sensing-related information. The figure shows where the IEEE 1900.6 interface is applied in the sensing-related information exchange.

[10] WiFi is a registered trademark in the U.S. Patent & Trademark Office, owned by the WiFi Alliance.
[11] ZigBee is a trademark in the U.S. Patent & Trademark Office, owned by the ZigBee Alliance.
[12] This information is given for the convenience of users of this standard and does not constitute an endorsement by the IEEE of these products. Equivalent products may be used if they can be shown to lead to the same results.

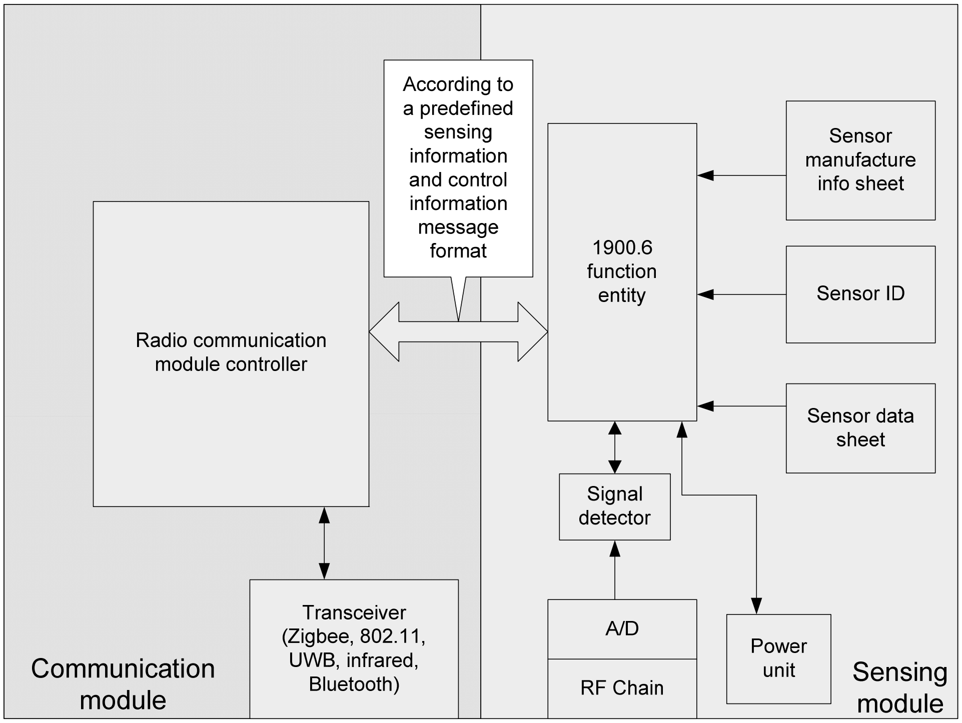


Figure C.2—Architecture of a sensor that can exchange sensing-related information through approved radio channels

C.1.3 Examples

C.1.3.1 Embedded/plugged sensor type I (spectrum sensor collocated with CE)

Embedded or plugged sensor type I referrers to a sensor that is physically wired to the CE capable device with a communication application programming interface. This is a typical example of where the IEEE 1900.6 logical interface is applicable. Interoperability between devices coming from different vendors can be provided by implementing a functional entity at both the client and the sensors that provide the IEEE 1900.6 logical interface as depicted in Figure C.3 and Figure C.4.

C.1.3.2 Embedded/plugged sensor type II (spectrum sensor not collocated with CE)

Embedded or plugged sensor type II referrers a sensor that is physically connected to a non-CE capable device. A non-CE capable device is any electronic device that has memory, computational resource for spectrum sensing application, and transceiver to communicate with other devices. Such devices can include but are not limited to legacy terminal such as cell phone, PDA, notebook, personal computer, AP, etc. The entire combined set of the spectrum sensing module and the client can be viewed as a smart sensing capable device. For example, smart sensing algorithms can be implemented on the client side for distributed sensing. One interesting aspect of the smart sensor capable device is that an existing office infrastructure such as laptops and APs can be exploited to deploy distributed spectrum sensing. Interoperability between devices coming from different vendors can be provided by implementing a functional entity at both the client and sensors that provide the IEEE 1900.6 logical interface. Figure C.4 depicts the block diagram of the smart sensor capable device.

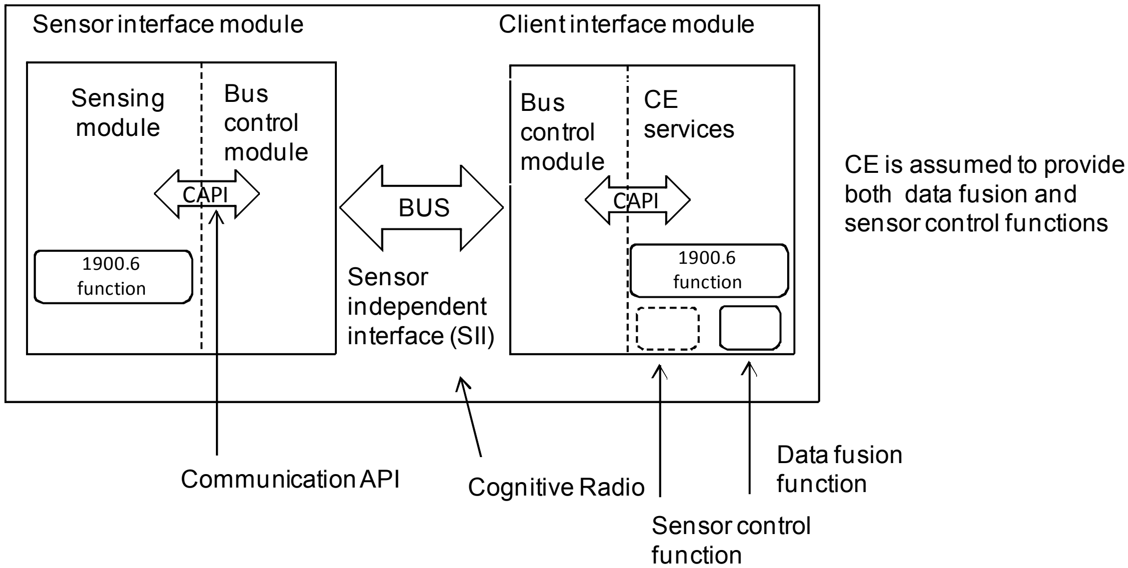


Figure C.3—Block diagram showing a wired spectrum sensor to a CE capable device

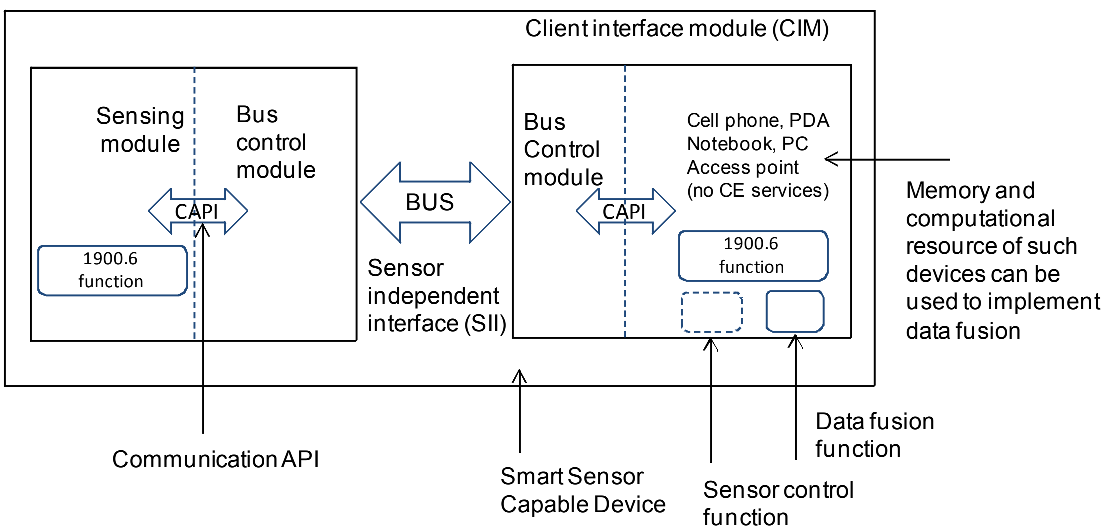


Figure C.4—Block diagram showing a spectrum sensor wired to a non-CE capable device

C.1.3.3 Distributed standalone sensor type I (information exchange between CE or DA and a sensor)

In distributed standalone sensor type I, sensors provide their sensing information to CE/DA through a wireless medium. Sensors in such an implementation example do not require any smart functionality, such as data fusion or control over other sensors. Simple spectrum sensors with wireless transceiver capability can be used to realize distributed sensing such as cooperative sensing, collaborative sensing, and selective sensing. Data fusion or sensor control is performed at the client side. Interoperability between devices coming from different vendors can be provided by implementing a functional entity at both the client and the sensors that provide the IEEE 1900.6 logical interface. Figure C.5 depicts a block diagram of a client.

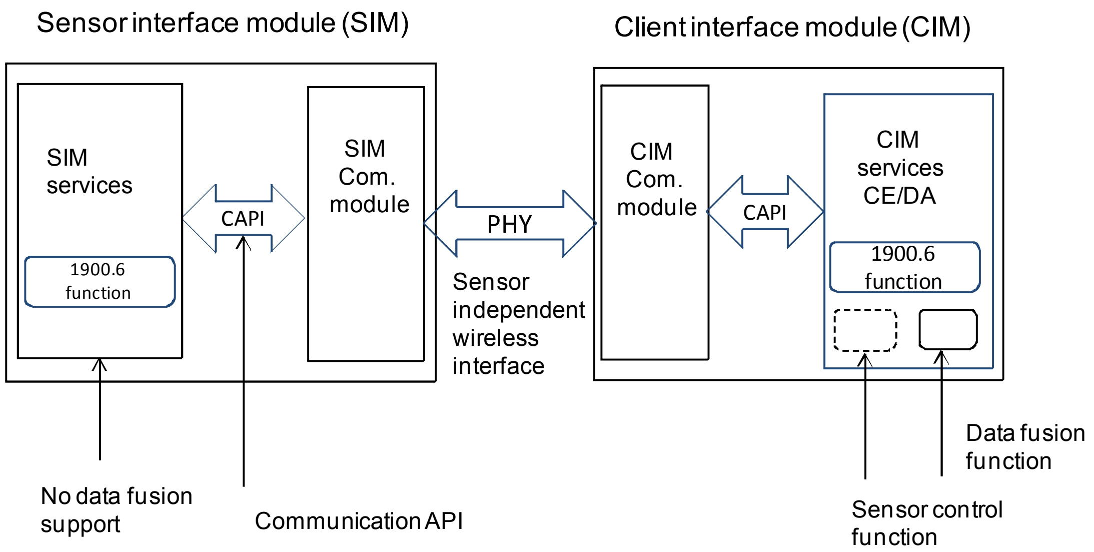


Figure C.5—Block diagram showing a spectrum sensor connected in a wireless medium to a client that is either a CE or DA

C.1.3.4 Distributed standalone sensor type II (information exchange between CE or DA and smart sensor)

Distributed standalone sensor type II, as depicted in Figure C.6, is an example of using the IEEE 1900.6 logical interface to exchange sensing control and sensing information between a CE/DA and a smart sensor.. Interoperability between devices coming from different vendors can be provided by implementing a functional entity at both the client and the sensors that provides the IEEE 1900.6 logical interface. The smart sensor can provide an aggregate of sensing information obtained from other sensors to the client to enhance sensing quality.

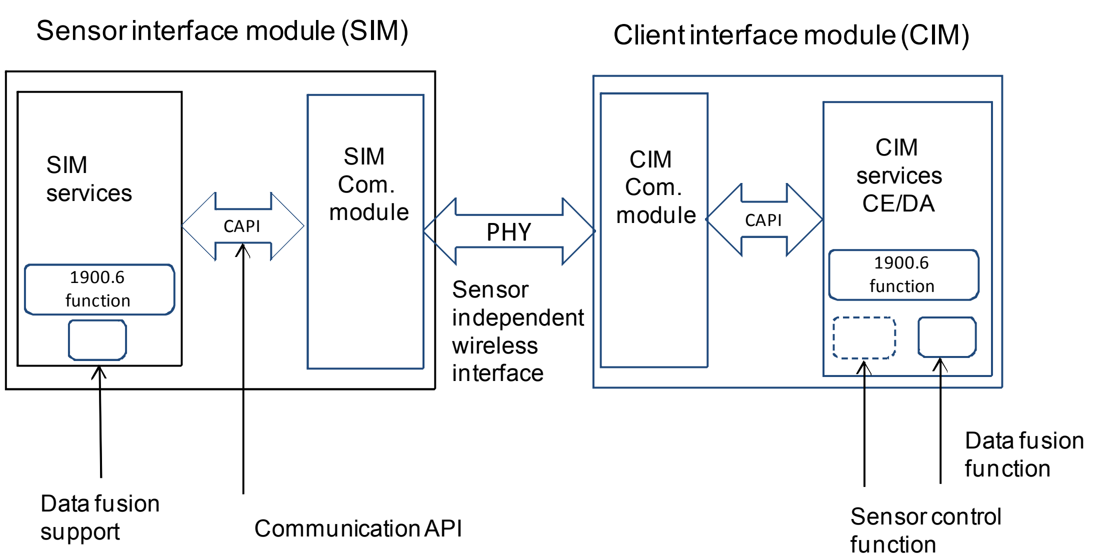


Figure C.6—Block diagram showing a smart spectrum sensor connected in a wireless medium to a client that is either a CE or DA

C.1.3.5 Distributed standalone sensor type III (information exchange between sensor and smart sensor or between smart sensors)

Distributed standalone sensor type III, as depicted in Figure C.7, is an example of using the IEEE 1900.6 logical interface to exchange sensing control and sensing information between a client spectrum sensor and another spectrum sensor. The client sensor is a sensor with application for either data fusion or relaying. It obtains sensing information from other sensors, merges it with its own sensing information, and forwards it to a CE. The smart sensor can be an integrated sensor manufactured to provide the preceding functionality or a smart sensor capable device introduced in Figure C.4. This particular implementation of the IEEE 1900.6 logical interface assists distributed spectrum sensing where sensors share their sensing information to make optimum local decisions before forwarding the final result to the CE. Furthermore, it can assist the relaying function of sensing-related information.

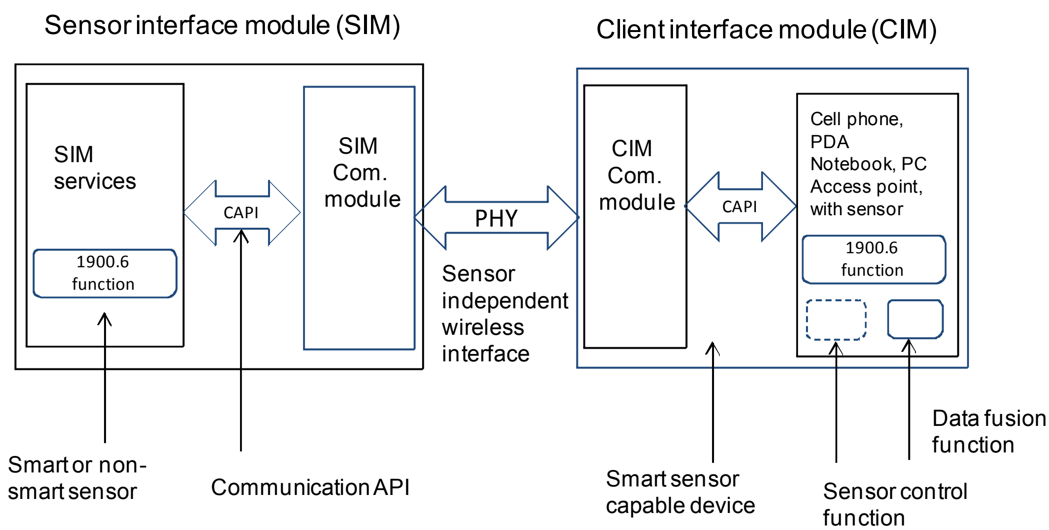


Figure C.7—Block diagram showing a spectrum sensor connected in a wireless medium to a client spectrum sensor

Annex D

(informative)

IEEE 1900.6 DA: Scope and usage

D.1 Introduction

The DA is a logical entity in which sensing-related information obtained from spectrum sensors or other sources, as well as regulatory and policy information, are processed and stored systematically. The DA processing capability is limited to storing, retrieving, data format conversion, and querying (fundamental data processing). Analyzing sensing-related information for decision making is done by the CE.

Annex E clarifies the scope and possible usage of the DA by describing the information to be managed and to be provided by the DA, minimum functionality of the DA, as well as possible locations of the DA. The scope and usage of DA are shown in Figure D.1.

Note that the DA may contain more information and may have interfaces to other entities that are not discussed in this standard. Also, the details of data processing algorithms inside the DA as well as administrative issues for the DA are not included.

D.2 Scope and usage of the DA

The DA can be a part of a CR system; it stores sensing-related information obtained from multiple sources such as spectrum sensors, CEs, other DAs, or other external data repositories. The DA also stores regulatory and policy information obtained from regulatory repositories. An example of a regulatory repository is the TV bands database defined in FCC ET Docket No. 08-260 [B1]. Information stored in the DA can be provided to other DAs or CEs. This information can be provided in the original form (as it is) or after processing by some predefined algorithms. The IEEE 1900.6 logical interface can be used to convey sensing-related information and processed information. However, regulatory information exchange may need another interface.

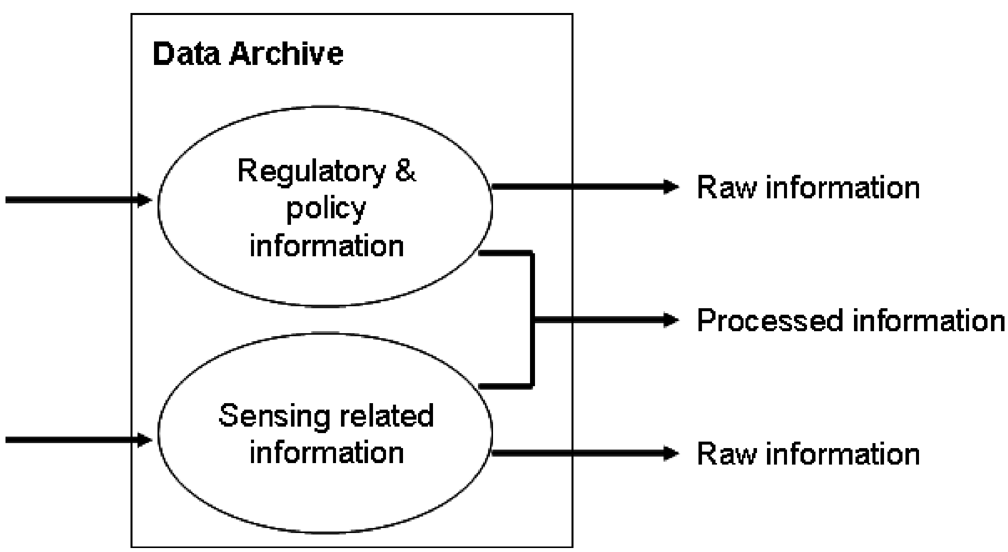


Figure D.1—Scope and usage of DA

D.3 Categories of information stored at DA

Examples of information stored at a DA are shown in D.3.1 and D.3.2.

D.3.1 Sensing-related information

Sensing-related information includes sensing information, sensing control information and control commands, sensor information, and requirements derived from regulation (cf. Clause 7 for parameter description):

- Spectrum measurement information and/or local decision of spectrum usage
- Location and time stamp of the spectrum measurement information and spectrum usage information
- Sensing method, sensor specifications, and information accuracy such as confidence level

Note that sensing-related information is dynamically changed and that its validity is in short time and for a relative small geographical area.

D.3.2 Regulatory and policy information

This information can be obtained from a database containing information on license holders, facility operation parameters (e.g., frequency, location, etc.), and any special conditions that apply. Based on this information, requirement as sensing-related information can be derived (cf. 7.1.5):

- Transmitter location or geographic area of operation
- Effective radiated power
- Transmitter height above average terrain
- Antenna height above ground level
- Call sign
- License holders spectrum masks,
- License holders receiver sensitivity, etc.

Note that regulatory and policy information is relative stable information. The information does not change frequently for a relative large area.

D.4 DA Requirements

- Information acquisition capability

  The DA shall have the capability to obtain sensing information from spectrum sensors or other sources and shall have the capability to obtain regulatory and policy information from regulatory repositories or other sources.

- Information storage capability

  The DA shall have the capability to store sensing information and regulatory/policy information.

— Information provision capability

  The DA shall have the capability to provide storage information to other clients (e.g., CE or another DA).

— IEEE 1900.6 compliance

  The DA shall comply with the IEEE 1900.6 reference model (cf. Clause 5) and shall be able to exchange information according to the IEEE 1900.6 logical interface.

— Systematic organization of information

  Sensing information and regulatory/policy information shall be processed and stored systematically in the DA. To improve the performance of information exchange, stored data should be classified, sorted, and indexed.

— Preliminary data processing capability of raw sensing information

  The DA may provide storage information in the original form, or it may provide the processed information as a result of regulatory information and sensing information according to some algorithms.

The input for these processes can be as follows:

— Enforcing regulatory body
— Protected users and their area
— Current location of the DA and CRs
— Sensing information

The output of these processes can be as follows:

— Reliable sensing information collected over a long period of time possibly obtained from multiple sensors.

D.5 Necessary interfaces

The following interfaces are required for a DA to exchange information:

— Interface to spectrum sensors
— Interface to CEs
— Interface to other DAs
— Interface to regulatory and policy repositories

D.6 DA services

The DA may provide the following services to other IEEE 1900.6 logical entities:

— The repository for sensing information from spectrum sensors.
— The register for CEs, sensors, and other DAs. Registered information may include the logical ID and geolocation of the entities. For example, the DA may keep the logical ID of sensors and

provide this information to the CE, where this information can be used to compute the confidence level.

— The information source for CEs or other DAs. To support this service, the DA may have the capability to respond to queries from CE/DAs.

D.7 Deployment examples

One possibility is to implement the DA as shown in Figure D.2. In this case, the DA needs to have an interface to the regulatory repository to obtain regulatory and policy information, as shown in Figure D.2.

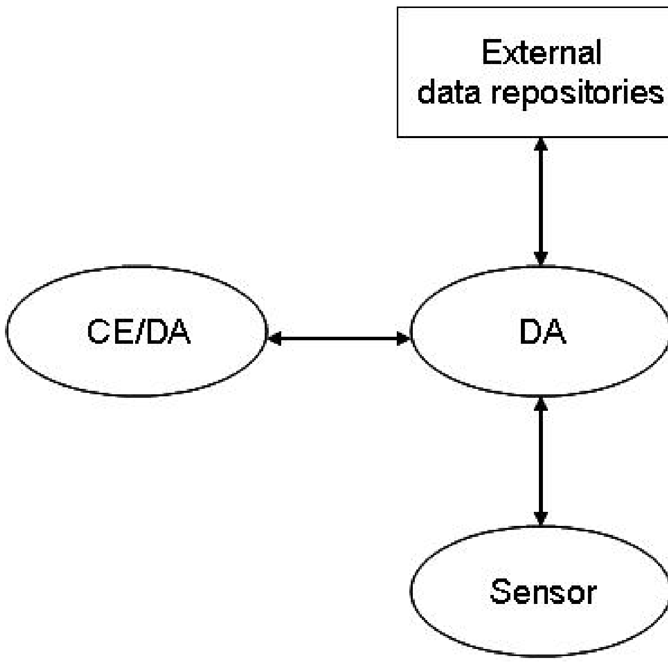


Figure D.2—DA has interface to regulatory repository

Another possible deployment is shown in Figure D.3. If the regulatory repository has the capability to store and exchange sensing information and has the IEEE 1900.6 interface, it is considered an IEEE 1900.6 DA.

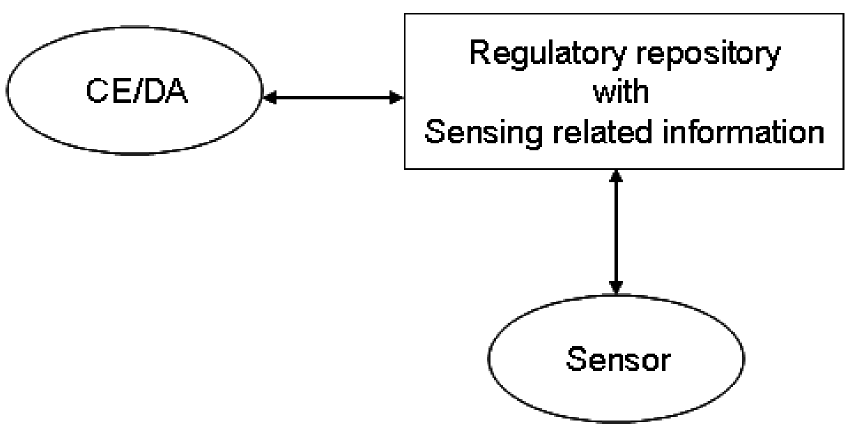


Figure D.3—DA includes the regulatory repository

Annex E

(informative)

Analysis of available/future technologies

To have an overview of the spectrum sensing technologies and techniques, two surveys of the algorithms being researched and the equipment available on the market have been edited.[13]

The first of these documents, titled *Review of Contemporary Spectrum Sensing Technologies*, catalogues a representative sample of available spectrum sensing technologies and then through analysis ascertains the interface features between the "sensing system" and the "CE" that are common among them (see Pucker et al. [B8]).

The second document, titled *Sensing Techniques for Cognitive Radio—State of the Art and Trends,* identifies the spectrum sensing techniques being used and researched in the field of cognitive radio. It provides a comprehensive overview on the techniques that have been proposed and the ones that may emerge in the next few years. Links to an in-depth bibliography are provided systematically to enable the reader to get more technical details (see Noguet et al. [B7]).

[13] These reports are available at http://www.scc41.org (IEEE 1900.6 folder).

Annex F

(informative)

Bibliography

[B1] FCC ET Docket No. 08-260, "Second report and order and memorandum opinion and order," 2008.[14,15]

[B2] IEEE P802.22™/D2.08, May 2009, Draft Standard for Wireless Regional Area Networks—Part 22: Cognitive Wireless RAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Policies and Procedures for Operation in the TV Bands.[16,17,18]

[B3] IEEE Std 754™-2008, IEEE Standard for Floating-Point Arithmetic.

[B4] IEEE Std 802.11™-2007, IEEE Standard for Information Technology—Telecommunications and Information Exchange between Systems—Local and Metropolitan Area Networks—Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.

[B5] ITU-T Recommendation X.210, Information Technology—Open Systems Interconnection—Basic Reference Model: Conventions for the Definition of OSI Services.[19]

[B6] National Imagery and Mapping Agency, *Department of Defense World Geodetic System 1984—Its Definition and Relationships with Local Geodetic Systems*, Technical Report TR8350.2, Bethesda, MD, 2004.[20]

[B7] Noguet, D., Demessie, Y. A., Biard L., Bouzegzi A., Debbah M., Haghighi K., Jallon P., Laugeois M., Marques P., Murroni M., Palicot J., Sun C., Thilakawardana S., and Yamaguchi A., *Sensing Techniques for Cognitive Radio—State of the Art and Trends*, IEEE SCC41—IEEE 1900.6 Working Group, White Paper, Apr. 2009.[21]

[B8] Pucker L., *Review of Contemporary Spectrum Sensing Technologies*, IEEE SCC41—IEEE 1900.6 Working Group, White Paper, Sept. 2008.[22]

[B9] SDR Forum, "Use Cases for Cognitive Applications in Public Safety Communications Systems—Volume 1: Review of the 7 July Bombing of the London Underground," Approved 8 November 2007, SDRF-07-P-0019-V1.0.0.[23]

---

[14] Available at: http://hraunfoss.fcc.gov/edocs_public/attachmatch/FCC-08-260A1.pdf.
[15] FCC publications are available from the Federal Communications Commission, 445 12th Street SW, Washington, DC 20554 (http://www.fcc.gov).
[16] IEEE publications are available from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, Piscataway, NJ 08854, USA (http://standards.ieee.org/).
[17] The IEEE standards or products referred to in this clause are trademarks owned by the Institute of Electrical and Electronics Engineers, Incorporated.
[18] This IEEE standards project was not approved by the IEEE-SA Standards Board at the time this publication went to press. For information about obtaining a draft, contact the IEEE.
[19] ITU-T publications are available from the International Telecommunications Union, Place des Nations, CH-1211, Geneva 20, Switzerland/Suisse (http://www.itu.int/).
[20] Available at: http://earth-info.nga.mil/GandG/publications/tr8350.2/tr8350_2.html.
[21] Available at: http://grouper.ieee.org/groups/scc41/6/documents/white_papers/P1900.6_WhitePaper_Sensing_final.pdf.
[22] Available at: http://grouper.ieee.org/groups/scc41/6/documents/white_papers/P1900.6_Sensor_Survey.pdf.
[23] Available at: http://groups.sdrforum.org/download.php?sid=769.