1900.4™

IEEE Standard for Architectural Building Blocks Enabling Network-Device Distributed Decision Making for Optimized Radio Resource Usage in Heterogeneous Wireless Access Networks

IEEE Standards Coordinating Committee 41

Sponsored by the
IEEE Standards Coordinating Committee 41 on
Dynamic Spectrum Access Networks

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

27 February 2009

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

Sponsor

IEEE Standard Coordination Committee 41 on Dynamic Spectrum Access Networks

Approved 30 January 2009

IEEE-SA Standards Board

Abstract: The building blocks comprising (i) network resource managers, (ii) device resource managers, and (iii) the information to be exchanged between the building blocks, for enabling coordinated network-device distributed decision making that will aid in the optimization of radio resource usage, including spectrum access control, in heterogeneous wireless access networks are defined. The standard is limited to the architectural and functional definitions at a first stage. The corresponding protocols definition related to the information exchange will be addressed at a later stage.
Keywords: architectural building blocks, dynamic spectrum access, heterogeneous wireless networks, network-device distributed optimization of spectrum usage

---

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

PDF: ISBN 978-0-7381-5895-2 STD95901
Print: ISBN 978-0-7381-5896-9 STDPD95901

Introduction

This introduction is not part of 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.

Multimode reconfigurable devices are increasingly being adopted within the wireless industry. The choice among various supported air interfaces on a single wireless device is already a reality today, with devices offering, for example, second and third generation cellular radio access technologies and IEEE 802® wireless standards. Last but not least, devices and networks with dynamic spectrum access capabilities that allow the use of spectrum resource simultaneously among different systems are emerging and will be part of the radio eco space. There is a need to define a standard addressing the overall system architecture and information exchange between the network and devices that will allow the devices to optimally choose among the available radio resources and simultaneously use several of these resources such that the overall efficiency and capacity of the resulting composite network is improved.

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. In February 2007, the IEEE 1900.4 working group was launched, originating from the IEEE 1900.B study group, and entitled “Architectural Building Blocks Enabling Network-Device Distributed Decision Making for Optimized Radio Resource Usage in Heterogeneous Wireless Access Networks.”

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 this 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 P1900.4 Working Group had the following entity membership:

Alcatel-Lucent
BAE Systems, NES
BitMeister, Inc.
Cosmo Research Corp
France Telecom
Hitachi
Intel
ISB
King's College London (KCL)
KDDI R&D Laboratories, Inc.
Kozo Keikaku Engineering, Inc.
Motorola Labs
NCSR Demokritos
NEC
National Institute of Information and Communications Technology (NICT)
PULTEK Corp.
Technical University of Catalonia (UPC)
Tokyo University of Science
Toshiba Research Europe Limited
University of Athens (UoA)
University of Piraeus Research Center, Department of Digital Systems (UPRC)
Willnet, Inc.
Worldpicom Corporation

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

Alcatel-Lucent
Bio Energy Solutions
BitMeister, Inc.
Cosmo Research Corp
Cyverse Corporation
France Telecom
Fuji Infox-Net Co., LTD
Hirotech, Inc.
ISB Corporation
Infinico Corp
Intel
KDDI R&D Laboratories, Inc.
Kozo Keikaku Engineering, Inc.
Motorola Labs
National Communications System (DHS/NPPD/CS&C/NCS)
National Institute of Information and Communications Technology (NICT)
National and Capodistrian University of Athens
Nokia
PULTEK Corp.
The Boeing Company
The Software Defined Radio Forum Inc.
Tokyo University of Science
Toshiba Research Europe Limited
Universitat Politecnica de Catalunya
University of Piraeus Research Center (UPRC)
Willnet, Inc.
Worldpicom Corporation

The P1900.4 Working Group gratefully acknowledges the contributions of the following organizations and participants. Without their assistance and dedication, this standard would not have been completed.

Organization	*Participant*
Alcatel-Lucent	Klaus Nolte
BAE Systems, NES	Ralph Martinez, *Co-Vice Chair and Technical Editor*
	Christian Rodriguez
Bit Meister	Jun Watanabe
Cosmo Research	Yoshiaki Mayuzumi
France Telecom	Patricia Martigne
	Paul Houzé, *Chair*
	Sana Ben Jemaa
	Sana Horrich
	Servane Bonjour
	Sebastien Jeux

Hitachi .... Junji Yamamoto
Seishi Hanaoka
Intel .... Vladimir Ivanov
Oleg Semenov
ISB .... Toru Kabe
King's College London (KCL) .... Oliver Holland, *Technical Editor*
Andrej Mihailovic
Alireza Attar
KDDI R&D .... Kazunori Takeuchi
Akira Yamaguchi
Tadayuki Fukuhara
Kozo Keikaku Engineering, Inc .... Masayuki Umeda
Motorola Labs .... Soodesh Buljore
Markus Muck
David Grandlaise
Remy Pintenet
Sophie Gault
Vincent Merat
Frederick Martin
Joanna Guenin
John Strassner
NCSR Demokritos .... Lambros Sarakis
Nikos Dimitriou
NEC .... Julio Arauz
Masayuki Ariyoshi
Stefan Aust
Tetsuya Ito
Kazushi Muraoka
National Institute of Information and Communications Technology (NICT) .... Hiroshi Harada, *Co-Vice Chair*
Homare Murakami
Kentaro Ishizu
Stanislav Filin, *Technical Editor*
Goh Miyamoto
Ha Nguyen Tran
Pultek .... Tomomi Aketo
Tokyo University of Science .... Mikio Hasegawa
Toshiba Research Europe Limited .... Gary Clemo
Tim Farnham
Mahesh Sooriyabandara
University of Athens (UoA), Department of Informatics and Telecommunications .... Makis Stamatelatos
Costas Polychronopoulos
Technical University of Catalonia (UPC) .... Oriol Sallent
Ramon Agusti
Jad Nasreddine
Juan Sanchez
University of Piraeus Research Center, Department of Digital Systems (UPRC) .... Kostas Tsagkaris, *Technical Editor*
Panagiotis Demestichas
Aristi Galani
Willnet .... Tadashi Mori
Worldpicom .... Yukio Mori

When the IEEE-SA Standards Board approved this standard on 30 January 2009, it had the following membership:

Robert M. Grow, *Chair*
Thomas Prevost, *Vice Chair*
Steve M. Mills, *Past Chair*
Judith Gorman, *Secretary*

Victor Berman
Richard DeBlasio
Andy Drozd
Mark Epstein
Alexander Gelman
William Goldbach
Arnie. Greenspan
Ken Hanus
Jim Hughes
Richard Hulett
Young Kyun Kim
Joseph L. Koepfinger*
John Kulick
David J. Law
Glenn Parsons
Ron Petersen
Chuck Powers
Narayanan Ramachandran
Jon Walter Rosdahl
Anne-Marie Sahazizian
Malcolm Thaden
Howard Wolfman
Don Wright

*Member Emeritus

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

Satish K. Aggarwal, *NRC Representative*
Michael Janezic, *NIST Representative*

Lisa Perry
IEEE Standards Program Manager, Document Development

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

Noelle Humenick
IEEE Standards Corporate Client Manager

Contents

IEEE Standard for Architectural Building Blocks Enabling Network-Device Distributed Decision Making for Optimized Radio Resource Usage in Heterogeneous Wireless Access Networks

IMPORTANT NOTICE: This standard is not intended to ensure safety, security, health, or environmental protection in all circumstances. 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

1.1 Scope

The standard defines the building blocks comprising (i) network resource managers, (ii) device resource managers, and (iii) the information to be exchanged between the building blocks, for enabling coordinated network-device distributed decision making that will aid in the optimization of radio resource usage, including spectrum access control, in heterogeneous wireless access networks. The standard is limited to the architectural and functional definitions at a first stage. The corresponding protocols definition related to the information exchange will be addressed at a later stage.

1.2 Purpose

The purpose is to improve overall composite capacity and quality of service of wireless systems in a multiple Radio Access Technologies (RATs) environment, by defining an appropriate system architecture and protocols that will facilitate the optimization of radio resource usage, in particular, by exploiting information exchanged between network and mobile Terminals, whether or not they support multiple simultaneous links and dynamic spectrum access.

1.3 Document overview

The structure of this document is as follows:

— Clause 2 lists normative references
— Clause 3 lists definitions and abbreviations used in this document
— Clause 4 presents overall system description of IEEE Std 1900.4, comprising system overview, summary of use cases, and assumptions
— Clause 5 specifies IEEE Std 1900.4 requirements, comprising system requirements, functional requirements, and information model requirements
— Clause 6 defines IEEE Std 1900.4 architecture. It is comprised of system description and functional description of an IEEE 1900.4 system
— Clause 7 defines the IEEE Std 1900.4 information model
— Clause 8 describes IEEE Std 1900.4 procedures
— Informative Annex A describes IEEE Std 1900.4 use cases
— Normative Annex B gives class definitions for the IEEE Std 1900.4 information model
— Normative Annex C provides data type definitions for the IEEE Std 1900.4 information model
— Informative Annex D describes the IEEE Std 1900.4 information model extensions and usage example
— Informative Annex E describes IEEE Std 1900.4 deployment examples
— Informative Annex F lists informative references related to IEEE Std 1900.4

2. Normative references

The following referenced documents 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.[1, 2]

ISO/IEC 8824, Information Processing Systems—Open Systems Interconnection—Specification of Abstract Syntax Notation One (ASN.1).[3]

ITU-T Recommendation X.701, System Management Overview.[4]

Unified Modeling Language (UML), Version 2.1.2.[5]

---

[1] The IEEE standards or products referred to in this clause are trademarks of the Institute of Electrical and Electronics Engineers, Inc.

[2] IEEE publications are available from the Institute of Electrical and Electronics Engineers, Inc., 445 Hoes Lane, Piscataway, NJ 08854, USA (http://standards.ieee.org/).

[3] ISO/IEC publications are available from the ISO Central Secretariat, 1, ch. De la Voie-Creuse, Case Postale 56, CH-1211, Genève 20, Switzerland/Suisse (http://www.iso.ch/). ISO/IEC publications are also available in the United States from Global Engineering Documents, 15 Inverness Way East, Englewood, CO 80112, USA (http://global.ihs.com/). Electronic copies are available in the United States from the American National Standards Institute, 25 West 43rd Street, 4th Floor, New York, NY 10036, USA (http://www.ansi.org/).

[4] ITU-T publications are available from the International Telecommunications Union, Place des Nations, CH-1211, Geneva 20, Switzerland/Suisse (http://www.itu.int/).

[5] The UML is available from the Object Management Group Web site http://www.omg.org/spec/UML/2.1.2/June 2008.

3. Definitions, acronyms, and abbreviations

3.1 Definitions

For the purposes of this document, the following terms and definitions apply.

3.1.1 base station: This term is used to refer to any radio node on the network side from radio interface, independently of its commonly used name in a particular standard. Examples of common name are Base Station in IEEE Std 802.16, Base Transceiver System in cdma2000, Node B in UMTS, Access Point in IEEE Std 802.11, broadcasting transmitter, etc.

3.1.2 Composite Wireless Network (CWN): This term is used to refer to a network composed of several radio access networks with corresponding base stations, a packet-based core network connecting these radio access networks, and IEEE 1900.4 entities deployed in this network.[6]

NOTE—This definition does not exclude the case where some broadcasting system or future technology system is part of the composite wireless network.[7]

3.1.3 context information: This term is used to refer to any information that together with policies is needed for decision making on radio resource usage optimization in this standard. Radio access network (RAN) context information is distinguished from terminal context information. Also, context information is distinguished from policies.

3.1.4 distributed radio resource usage optimization: The distributed optimization of radio resource usage by a composite wireless network to satisfy global network objectives and by terminals to satisfy local device and user objectives (see IEEE Std 1900.1).

3.1.5 dynamic spectrum assignment: The dynamic assignment of frequency bands to radio access networks within a composite wireless network operating in a given region and time to optimize spectrum usage (see IEEE Std 1900.1).

3.1.6 dynamic spectrum sharing: The process and mechanisms for a type of spectrum access that occurs when different radio access networks and Terminals dynamically access spectrum bands which are overlapping, in whole or in part, causing less than an admissible level of mutual interference, according to regulatory rules, and may be done with or without negotiation.

NOTE—The IEEE 1900.4 definition of dynamic spectrum sharing is intentionally more specific than the IEEE 1900.1 definition of dynamic frequency sharing (see A.2).

3.1.7 IEEE 1900.4 compliant terminal: This term is used to refer to any IEEE 1900.4 compliant radio node on the user side, that is, a reconfigurable terminal containing the Terminal Reconfiguration Manager, Terminal Measurement Collector, and Terminal Reconfiguration Controller IEEE 1900.4 entities.

3.1.8 multi-homing capability: This term is used to refer to a capability of a reconfigurable terminal to have more than one simultaneous active connections with radio access networks.

3.1.9 Network Reconfiguration Manager: The entity that manages the Composite Wireless Network and Terminals in terms of network-terminal distributed optimization of spectrum usage. This management is

[6] The IEEE Std 1900.4 definition of composite wireless network is intentionally more specific than the IEEE Std1900.1 definition of composite network. The IEEE 1900.4 definition describes the components of composite wireless networks in the context of the IEEE 1900.4 system architecture.

[7] Notes in text, tables, and figures are given for information only and do not contain requirements needed to implement the standard.

done within the framework of spectrum assignment policies conveyed by the Operator Spectrum Manager and in a manner consistent with available context information.

3.1.10 Operator Spectrum Manager: The entity that enables the operator to control the dynamic spectrum assignment decisions of the Network Reconfiguration Manager.

3.1.11 radio access network (RAN): The network that connects base stations to the packet-based core network or external networks. If not specified, radio access network includes base stations.

3.1.12 radio access network (RAN) context information: This term is used to refer to any information that together with policies is needed for decision making on radio resource usage optimization in this standard and has RANs as its source.

3.1.13 radio access network (RAN) Measurement Collector: The entity that collects RAN context information and provides it to the Network Reconfiguration Manager.

3.1.14 radio access network (RAN) Reconfiguration Controller: The entity that enables the Network Reconfiguration Manager to control reconfiguration of radio access networks.

3.1.15 radio enabler: A logical communication channel between NRM and TRM. Radio enabler may be mapped onto one or several radio access networks used for data transmission (in-band channel) and/or onto one or several dedicated radio access networks (out-of-band channel).

3.1.16 radio interface: This term is used to refer to an air interface specifications that shall be fulfilled to setup and maintain connection between terminal and base station. Radio interface may be characterized by multiple access method, modulation, etc. Examples are GSM, WCDMA, WiFi®,[8] WiMAX™,[9] etc radio interfaces.[10]

3.1.17 radio resource selection policy: A policy generated by the Network Reconfiguration Manager which guides the Terminal Reconfiguration Managers in terms of their radio resource usage optimization decisions.

3.1.18 reconfigurable terminal: This term is used to refer to any radio node on the user side that can reconfigure its hardware and/or software in order to change its operating parameters in the physical and link layers, such as carrier frequency, signal bandwidth, radio interface, etc. A reconfigurable terminal may have multi-homing capability.

3.1.19 spectrum assignment policy: A policy generated by the Operator Spectrum Manager that guides the Network Reconfiguration Manager in terms of its radio resource usage optimization decisions.

3.1.20 Terminal: This is used as a short version of the term "**IEEE 1900.4 compliant terminal**."

3.1.21 terminal context information: This term is used to refer to any information that has the user and/or Terminal as its source, which together with policies is needed in order for decision making on radio resource usage optimization within this standard to be possible.

3.1.22 Terminal Measurement Collector (TMC): The entity that collects terminal context information and provides it to Terminal Reconfiguration Manager.

---

[8] WiFi® is a word mark of the WiFi Alliance.
[9] WiMAX™ is a trademark of the WiMAX Forum.
[10] 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.

3.1.23 Terminal Reconfiguration Controller (TRC): The entity that enables the Terminal Reconfiguration Manager to control reconfiguration of the Terminal.

3.1.24 Terminal Reconfiguration Manager (TRM): 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 Network Reconfiguration Manager and in a manner consistent with the user's preferences and the available context information.

3.1.25 user preferences: This term is used to refer to input parameters to decision making process on radio resource usage optimization originated from user and expressing his or her preferences. These parameters may describe, for example, preferred operator and radio interface, perceived audio/image/video quality, maximum cost, minimum data rate, etc.

3.2 Acronyms and abbreviations

ASN	Abstract Syntax Notation
BS	base station
CWN	Composite Wireless Network
NRM	Network Reconfiguration Manager
OSI	Open Systems Interconnection
OSM	Operator Spectrum Manager
QoS	quality-of-service
RAN	radio access network
RMC	RAN Measurement Collector
RRC	RAN Reconfiguration Controller
SAP	service access point
SMAE	system management application entity
TMC	Terminal Measurement Collector
TRC	Terminal Reconfiguration Controller
TRM	Terminal Reconfiguration Manager
UML	Unified Modeling Language

4. Overall system description

4.1 System overview

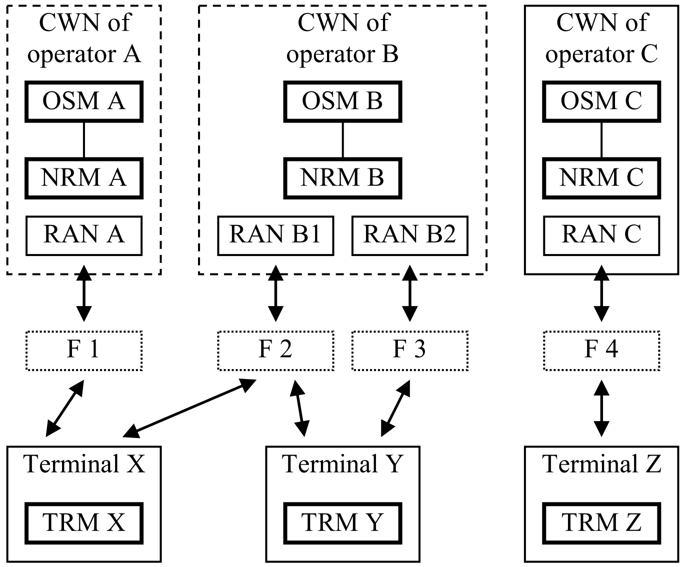
The field of application of this standard is a heterogeneous wireless environment that might include the following (see Figure 1):

— Multiple operators

— Multiple radio access networks (RANs)

— Multiple radio interfaces

— Multiple Terminals



CWN – Composite Wireless Network
OSM – Operator Spectrum Manager
RAN – Radio Access Network
NRM – Network Reconfiguration Manager
TRM – Terminal Reconfiguration Manager
F 1, 2, 3, and 4 – Frequency Bands 1, 2, 3, and 4

Figure 1—Heterogeneous wireless environment considered in IEEE Std 1900.4

Within Figure 1, the Operator Spectrum Manager (OSM) may help the operator to coordinate the assignment of spectrum to the different RANs it owns in order to optimize radio resource usage within its Composite Wireless Network (CWN).

Within the stated field of application, the standard provides common means to

- Improve overall network capacity and quality of service
- Facilitate optimization of radio resource usage
- Support reconfiguration capabilities of RANs and Terminals
- Collect RAN and terminal context information
- Support exchange of information between the network and Terminals for radio resource usage optimization related distributed decision making
- Request and control reconfiguration of RANs and Terminals

For this purpose, the standard defines the following:

- Network resource managers
- Device resource managers
- Interfaces between these building blocks

Within Figure 1, the Network Reconfiguration Manager (NRM) is the IEEE 1900.4 entity (representing a network resource manager) that manages the CWN and Terminals in terms of network-terminal distributed optimization of spectrum usage. This management is done within the framework of spectrum assignment policies conveyed by the OSM and in a manner consistent with available context information. Terminal Reconfiguration Manager (TRM) is an IEEE 1900.4 entity (representing a device resource manager) 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.

4.2 Summary of use cases

The following three use cases are defined within this standard:

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

In the dynamic spectrum assignment use case, frequency bands are dynamically assigned to the RANs among the participating networks in order to optimize spectrum usage. In other words, the assigned frequency bands are not fixed, and can be dynamically changed.

OSMs generate spectrum assignment policies expressing the regulatory framework and operators objectives for spectrum usage optimization. The OSMs provide these spectrum assignment policies to the corresponding NRMs.

The NRMs analyze spectrum assignment policies and available context information and dynamically make spectrum assignment decisions to improve spectrum usage and quality of service.

After the new spectrum assignment decisions have been made, the NRMs request corresponding reconfiguration of their RANs. Following the RANs' reconfiguration, Terminals need to reconfigure correspondingly.

Example manifestations for this dynamic spectrum assignment use case are spectrum sharing, and spectrum renting between RANs. In the spectrum renting example, frequency bands of one RAN are assigned to another RAN on a temporary basis. In the spectrum sharing example, one frequency band is shared by several RANs.

Single operator and multiple operator scenarios are described within the dynamic spectrum assignment use case. Within the multiple operator scenario, there is either one NRM outside of operators CWNs or one NRM inside each operator CWN.

In the dynamic spectrum sharing use case, frequency bands assigned to RANs are fixed. However, a particular frequency band can be shared by several RANs. In other words, the dynamic spectrum sharing use case describes how fixed frequency bands are shared and/or used dynamically by RANs and Terminals.

NRMs analyze available context information and dynamically make spectrum access decisions to improve spectrum usage and quality of service. NRMs make these spectrum access decisions within the framework defined by spectrum assignment policies. Following these decisions, NRMs request corresponding reconfiguration of their RANs.

NRMs dynamically generate radio resource selection policies and send them to their 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 to improve spectrum usage and quality of service. These spectrum access decisions are made within the framework of the radio resource selection policies. Following these decisions, each TRM requests corresponding reconfiguration of its Terminal.

Dynamic spectrum sharing use case includes primary/secondary spectrum usage as a special case.

In the distributed radio resource usage optimization use case, frequency bands assigned to RANs are fixed. Reconfiguration of RANs is not involved in this use case.

Distributed radio resource usage optimization use case considers Terminals with or without multi-homing capability. Decision on Terminal reconfiguration is made by its TRM and is supported by NRM.

NRMs analyze available context information, dynamically generate radio resource selection policies, and send them to their 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 spectrum usage and quality of service. TRMs make these reconfiguration decisions within the framework defined by radio resource selection policies. Following these decisions, TRMs request corresponding reconfiguration of their Terminals.

Detailed description of these three use cases and their scenarios is given in Annex A.

4.3 Assumptions

4.3.1 General

This standard does not specify MAC and PHY layers of RANs and Terminals.

This standard does not specify the way measurements are done on network and terminal side.

A heterogeneous wireless environment exists, including one or several CWNs.

Terminals are present within this heterogeneous wireless environment.

These Terminals are reconfigurable with or without multi-homing capability.

4.3.2 Dynamic spectrum assignment

Assignment of spectrum to RANs can be dynamically changed, where spectrum assignment may be characterized by carrier frequency, signal bandwidth, and radio interface to be used in the assigned spectrum.

Concurrently with the new spectrum assignments, RANs can be reconfigured.

Concurrently with RAN reconfigurations, Terminals can be reconfigured.

4.3.3 Dynamic spectrum sharing

Assignment of spectrum to RANs is fixed.

Some RANs are allowed to concurrently operate in more than one spectrum assignment for dynamic spectrum sharing.

Some spectrum assignments are allowed to be shared by several RANs during dynamic spectrum sharing.

Some RANs can be reconfigured during dynamic spectrum sharing (maintaining their allocated spectrum assignments).

Some RANs cannot be reconfigured (for example, RANs of primary systems).

Concurrently with RAN reconfiguration, Terminals can be reconfigured during dynamic spectrum sharing.

4.3.4 Distributed radio resource usage optimization

Assignment of spectrum to RANs is fixed.

Reconfiguration of RANs is not involved in this use case.

Terminals with or without multi-homing capability are reconfigured during distributed radio resource usage optimization.

5. Requirements

5.1 System requirements

5.1.1 Decision making

There shall be an entity on network side, called Network Reconfiguration Manager (NRM), responsible for managing the CWN and Terminals for network-terminal-distributed optimization of spectrum usage.

There shall be an entity on terminal side, called Terminal Reconfiguration Manager (TRM), responsible for managing the Terminal for network-terminal-distributed optimization of spectrum usage.

The TRM shall manage the Terminal within the framework defined by the NRM and in a manner consistent with user's preferences and available context information.

Decision making in this standard is based on policy-based management framework.

There shall be an entity on network side responsible for generating spectrum assignment policies.

The NRM shall be able to obtain spectrum assignment policies.

Spectrum assignment policies shall adhere to regulations.

Spectrum assignment policies shall express operator needs and/or radio resource usage objectives related to dynamic spectrum assignment.

The NRM shall make dynamic spectrum assignment decisions compliant with received spectrum assignment policies.

The NRM shall provide information on its dynamic spectrum assignment decisions to the entity on network side within the operator's control, which is responsible for generating spectrum assignment policies.

The NRM shall make dynamic spectrum sharing decisions compliant with its spectrum assignment decisions and with received spectrum assignment policies.

The NRM shall generate radio resource selection policies.

The NRM shall provide radio resource selection policies to TRM.

The NRM may have means to specify geo-location-based radio resource selection policies to TRM.

Radio resource selection policies and context information should be sent to an optimized set of TRMs in order not to overload selected RANs with broadcast. Target set of TRMs to be addressed may be based on combination of the following:

- RAN topology information: TRMs under coverage of list of base station IDs, such as call IDs, base transceiver station IDs, access points IDs, radio network controller IDs (if any), etc.
- TRMs inside given geo-localized area
- TRMs operating in given frequency bands
- Historical data related to Terminal radio resource usage patterns

The TRM shall make decisions on Terminal reconfiguration compliant with received radio resource selection policies.

If specified by radio resource selection policies, TRM of geo-localization-capable Terminal shall make decisions based on Terminal geo-location.

Several NRMs may collaborate in the process of decision making related to radio resource usage optimization.

If there are several NRMs, there may be interface between these NRMs.

This interface may be used to transmit the following:

- RAN context information
- Terminal context information
- Spectrum assignment policies
- RAN reconfiguration decisions
- Radio resource selection policies

5.1.2 Context awareness

There shall be entities on network side and terminal side responsible for context information collection.

Context information collection entity on network side shall collect RAN context information.

RAN context information may include the following:

- RAN radio resource optimization objectives
- RAN radio capabilities
- RAN measurements
- RAN transport capabilities

The NRM shall be able to obtain RAN context information from context information collection entity on network side.

The NRM may receive this context information periodically and/or in response to request from the NRM and/or on event.

Context information collection entity on network side may be implemented in a distributed manner.

Context information collection entity on terminal side shall collect terminal context information.

Terminal context information may include the following:

- User preferences
- Required QoS levels
- Terminal capabilities
- Terminal measurements
- Terminal geo-location information
- Geo-location based terminal measurements

The TRM shall be able to obtain terminal context information from context information collection entity on terminal side.

The NRM and the TRM shall exchange context information.

The NRM shall send RAN context information to the TRM.

The NRM may send to the TRM's terminal context information related to other Terminals.

The NRM may send this context information to the TRM periodically and/or in response to request from the NRM and/or on event.

The TRM shall send terminal context information related to its Terminal to the NRM.

The TRM may send this context information to the NRM periodically and/or in response to request from the NRM and/or on event.

5.1.3 Reconfiguration

There shall be entities on network side and terminal side responsible for reconfiguration.

The NRM shall send reconfiguration requests to reconfiguration entity on network side.

Following received reconfiguration requests, reconfiguration entity on network side shall request and control reconfiguration of RANs.

Reconfiguration entity on network side may be implemented in a distributed manner.

The TRM shall send reconfiguration requests to reconfiguration entity on terminal side.

Following received reconfiguration requests, reconfiguration entity on terminal side shall request and control reconfiguration of Terminal.

If a maximum time interval for reconfiguration is specified by radio resource selection policies, reconfiguration of Terminal shall be performed within this time interval starting from the time when these radio resource selection policies are received.

5.2 Functional requirements

5.2.1 NRM functionality

The NRM shall have capability to make dynamic spectrum assignment, dynamic spectrum sharing, and distributed radio resource usage optimization decisions.

The NRM shall have capability to request reconfiguration of RANs corresponding to these dynamic spectrum assignment and dynamic spectrum sharing decisions. Actions should be taken by the NRM in the case where reconfiguration is not possible.

The NRM shall have the capability to evaluate the efficiency of spectrum usage under the current spectrum assignment.

The evaluation results shall be made available inside the NRM to assist in improving the efficiency of future dynamic spectrum assignment, dynamic spectrum sharing, and distributed radio resource usage optimization decisions.

The NRM shall have capability to generate radio resource selection policies.

There shall be no conflict in the radio resource selection policies generated by the NRM.

Radio resource selection policies shall be defined in a way that they correspond to a targeted group of Terminals (could be composed of any number of Terminals).

The targeted group of Terminals should be defined based on the needs of CWN radio resource usage optimization objectives, Terminal location, and radio resource usage patterns of the Terminals.

Radio resource selection policies shall guide TRMs in Terminals' reconfiguration decisions.

The NRM may specify radio resource selection policies referring to specific geo-location-based terminal measurements.

The NRM shall have capability to specify and control the time interval within which Terminal reconfiguration shall be performed.

The TRM shall perform Terminal reconfiguration within this time interval.

The NRM shall have capability to evaluate the efficiency of current radio resource selection policies.

These evaluation results shall be made available to the NRM to assist in improving the efficiency of future radio resource selection policies.

The NRM shall have capability to receive, process, and store the following context information:

- RAN context information (see 5.1.2)
- Terminal context information (see 5.1.2)

The NRM shall have the capability to use this context information for radio resource usage optimization purposes.

The NRM should have the capability to select RANs for exchanging radio resource selection policies and context information between the NRM and the TRM, that is, to map radio enabler onto specific RANs.

The NRM functions should have capability to perform the following in cooperation with each other:

- Make dynamic spectrum assignment, dynamic spectrum sharing, and distributed radio resource usage optimization decisions
- Request reconfiguration of RANs
- Evaluate the efficiency of spectrum usage
- Generate radio resource selection policies
- Evaluate the efficiency of current radio resource selection policies
- Select RANs for exchanging radio resource selection policies and context information between the NRM and the TRM

If there are several NRMs, there may be an interface between these NRMs.

This interface shall be used to exchange the following:

- RAN context information
- Terminal context information
- Spectrum assignment policies
- RAN reconfiguration decisions
- Radio resource selection policies

5.2.2 TRM functionality

The TRM shall have the capability to make dynamic spectrum sharing and distributed radio resource usage optimization decisions, as well as, to support dynamic spectrum assignment decisions received from the NRM.

The TRM shall have the capability to request reconfiguration of its Terminal corresponding to these decisions. Actions should be taken by the TRM in cases where reconfiguration is not possible.

The TRM shall have the capability to receive, process, and store the following context information:

- Terminal context information
- RAN context information

The TRM shall have the capability to use this context information for radio resource usage optimization purposes.

The TRM shall have the capability to select RANs for exchanging radio resource selection policies and context information between the NRM and the TRM.

TRM functions should have the capability to perform the following in cooperation with each other:

- Make dynamic spectrum sharing and distributed radio resource usage optimization decisions
- Request reconfiguration of its Terminal

— Select RANs for exchanging radio resource selection policies and context information between the NRM and the TRM

5.3 Information model requirements

Information model shall provide a specified representation of information within the scope of this standard.

Information model shall consider two sets of managed objects, that is, CWN and Terminals.

CWN-related classes shall abstract operator, RAN, BS, and cell concepts within the scope of this standard.

Terminal-related classes shall abstract user, application, Terminal, frequency channel, and active connection concepts within the scope of this standard.

Information model shall abstract policy concept within the scope of this standard, including spectrum assignment policy and radio resource selection policy.

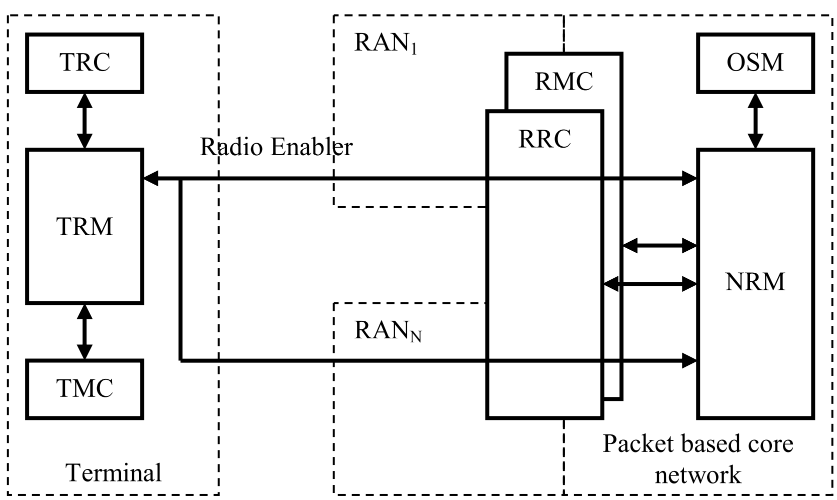
Information model may include time/duration reference related to the validity of the provided information. For instance the time at which measurements were made or the valid period in which they are to be taken.

Information model should provide geo-location related information items.

6. Architecture

6.1 System description

According to system requirements, the following system architecture is defined in this standard (see Figure 2).



TRM – Terminal Reconfiguration Manager
TRC – Terminal Reconfiguration Controller
TMC – Terminal Measurement Collector
OSM – Operator Spectrum Manager
NRM – Network Reconfiguration Manager
RRC – RAN Reconfiguration Controller
RMC – RAN Measurement Collector
RAN – radio access network

Figure 2—System architecture

6.1.1 Entities

The following four entities are defined to represent network resource managers (see Figure 2):

- Operator Spectrum Manager (OSM)
- RAN Measurement Collector (RMC)
- Network Reconfiguration Manager (NRM)
- RAN Reconfiguration Controller (RRC)

The OSM is the entity that enables operator to control NRM dynamic spectrum assignment decisions.

The RMC is the entity that collects RAN context information and provides it to NRM. RMC may be implemented in a distributed manner.

The NRM is the entity that manages CWN and Terminals for network-terminal distributed optimization of spectrum usage. NRM may be implemented in a distributed manner.

The RRC is the entity that controls reconfiguration of RANs based on requests from NRM. RRC may be implemented in a distributed manner.

Three following entities are defined to represent device resource managers (see Figure 2):

- Terminal Measurement Collector (TMC)
- Terminal Reconfiguration Manager (TRM)
- Terminal Reconfiguration Controller (TRC)

The TMC is the entity that collects terminal context information and provides it to the TRM.

The TRM is the entity that manages the Terminal for network-terminal distributed optimization of spectrum usage within the framework defined by the NRM and in a manner consistent with user preferences and available context information.

The TRC is the entity that controls reconfiguration of Terminal based on requests from the TRM.

Radio enabler is the logical communication channel between the NRM and the TRM. Radio enabler may be mapped onto one or several RANs used for data transmission (in-band channel) and/or onto one or several dedicated RANs (out-of-band channel).

6.1.2 Interfaces between entities

The following key interfaces are defined (see Figure 2):

- Interface between the NRM and the TRM
- Interface between the TRM and the TRC
- Interface between the TRM and the TMC
- Interface between the NRM and the RRC
- Interface between the NRM and the RMC
- Interface between the NRM and the OSM

6.1.2.1 Interface between the NRM and the TRM

Interface between the NRM and the TRM is used to transmit the following:

- From NRM to TRM:
  - Radio resource selection policies
  - RAN context information
  - Terminal context information
- From TRM to NRM:
  - Terminal context information related to Terminal of this TRM

6.1.2.2 Interface between the TRM and the TRC

Interface between the TRM and the TRC is used to transmit the following:

- From TRM to TRC:
  - Terminal reconfiguration requests
- From TRC to TRM:
  - Terminal reconfiguration responses.

6.1.2.3 Interface between the TRM and the TMC

Interface between the TRM and the TMC is used to transmit the following:

- From TRM to TMC:
  - Terminal context information requests
- From TMC to TRM:
  - Terminal context information

6.1.2.4 Interface between the NRM and the RRC

Interface between the NRM and the RRC is used to transmit the following:

- From NRM to RRC:
  - RAN reconfiguration requests
- From RRC to NRM:
  - RAN reconfiguration responses

6.1.2.5 Interface between the NRM and the RMC

Interface between the NRM and the RMC is used to transmit the following:

- From NRM to RMC:
  - RAN context information requests
- From RMC to NRM:
  - RAN context information

6.1.2.6 Interface between the NRM and the OSM

Interface between the NRM and the OSM is used to transmit the following:

- From OSM to NRM:
  - Spectrum assignment policies
- From NRM to OSM:
  - Information on spectrum assignment decisions

6.1.2.7 Interface between several NRMs

If there are several NRMs, a corresponding interface may be defined between these NRMs (not shown in Figure 2).

This interface is used to transmit the following:

- RAN context information
- Terminal context information
- Spectrum assignment policies
- RAN reconfiguration decisions
- Radio resource selection policies

6.1.3 Reference model

In general, each IEEE 1900.4 entity (OSM, RMC, NRM, RRC, TMC, TRM, and TRC) has the reference model shown in Figure 3. IEEE 1900.4 entities are modeled as a system management application entity (SMAE) (see ITU-T X.701 for SMAE specification).[11] The IEEE 1900.4 entity, as SMAE, is located on the application layer and has access to any layer of the OSI model.

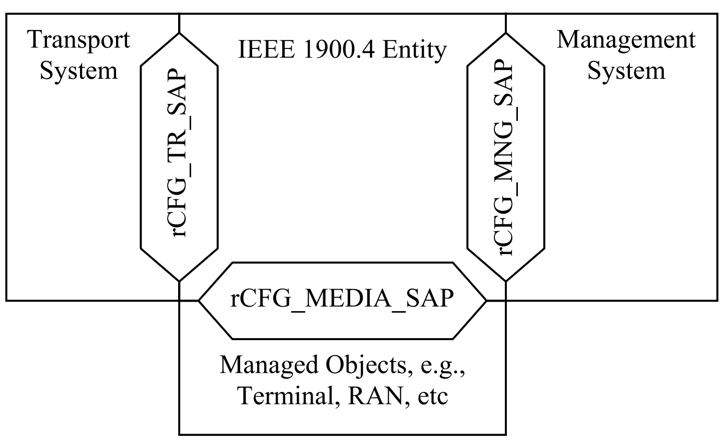


Figure 3—IEEE 1900.4 reference model

[11] Information on references can be found in Clause 2.

Each IEEE 1900.4 entity implements one or more of the following service access points (SAP):

— rCFG_TR_SAP – transport SAP

— rCFG_MEDIA_SAP – reconfiguration and measurement SAP

— rCFG_MNG_SAP – management SAP

Transport SAP provides transport service for message exchange between IEEE 1900.4 entities. It abstracts transport mechanisms from IEEE 1900.4 entities by providing a set of generic primitives and mapping these primitives on transport protocols.

For example, this SAP is used to exchange radio resource selection policies and context information between the NRM and the TRM over radio enabler.

If there are several NRMs and there is interface between them, this SAP is used to exchange context information, spectrum assignment policies, reconfiguration decisions, and radio resource selection policies between these NRMs.

Reconfiguration and Measurement SAP provides reconfiguration and measurement services for managing RANs and Terminals. It provides a set of generic primitives for IEEE 1900.4 entities to collect RAN and terminal context information, as well as, to control reconfiguration of RANs and Terminals. These generic primitives are mapped onto specific protocols depending on the managed RANs and Terminals.

Management SAP provides management service for managing IEEE 1900.4 entities by legacy management systems. This SAP provides a set of generic primitives for IEEE 1900.4 entities to exchange information with these legacy management systems.

6.2 Functional description

According to functional requirements, the following functional architecture is defined in this standard (see Figure 4).

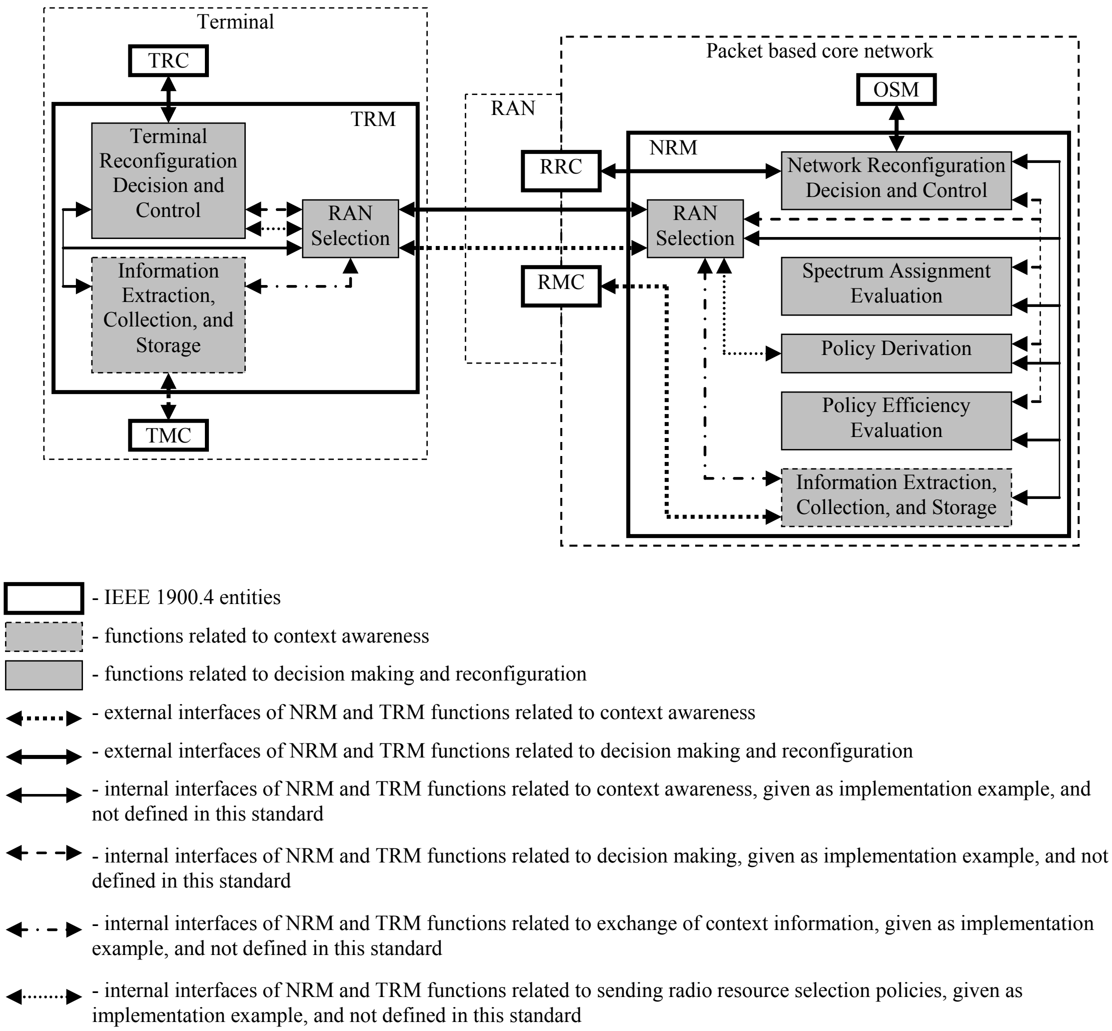


Figure 4—Functional architecture

6.2.1 NRM functions

The following functions are defined inside NRM (see Figure 4):

— Policy Derivation

— Policy Efficiency Evaluation

— Network Reconfiguration Decision and Control

— Spectrum Assignment Evaluation

— Information Extraction, Collection, and Storage

— RAN Selection

Policy Derivation function generates radio resource selection policies that guide TRMs in Terminals reconfiguration decisions.

The radio resource selection policies are derived according to context information from the Information Extraction, Collection, and Storage function.

The Policy Efficiency Evaluation function evaluates the efficiency of current radio resource selection policies.

Evaluation results may be used by the Policy Derivation function during generating radio resource selection policies.

The Network Reconfiguration Decision and Control function makes decisions on the RAN's reconfiguration compliant with spectrum assignment policies received from OSM. After making these decisions, Network Reconfiguration Decision and Control function sends corresponding reconfiguration requests to the RRC. Also, the Network Reconfiguration Decision and Control function sends information on the decisions that have been made to the OSM.

The Spectrum Assignment Evaluation function evaluates the efficiency of spectrum usage under the current spectrum assignment.

Evaluation results may be used by the Network Reconfiguration Decision and Control function while making decisions on the RAN's reconfiguration.

The Information Extraction, Collection, and Storage function receives, processes, and stores the following context information:

- RAN context information
- Terminal context information

RAN context information is received from the RMC periodically and/or by request and/or on event-basis.

Terminal context information is received from the TRM periodically and/or by request and/or on event-basis.

The Information Extraction, Collection, and Storage function provides information to functions inside the NRM.

The Information Extraction, Collection, and Storage function forwards RAN context information to the TRM.

The Information Extraction, Collection, and Storage function may forward terminal context information, related to other Terminals, to the TRM.

The RAN Selection function selects RANs for exchanging radio resource selection policies and context information between the NRM and one or several TRMs. Radio resource selection policies are sent from NRM to TRM. From NRM to TRM, RAN context information is sent and terminal context information may be sent. From TRM, NRM receives terminal context information related to Terminal managed by this TRM.

Policy Derivation, Policy Efficiency Evaluation, RAN Selection, Network Reconfiguration Decision and Control, and Spectrum Assignment Evaluation functions cooperate during their operation. They represent different aspects of decision making and reconfiguration. During their operation, these functions use information from the Information Extraction, Collection, and Storage function.

6.2.2 TRM functions

The following functions are defined inside TRM (see Figure 4):

- Terminal Reconfiguration Decision and Control
- Information Extraction, Collection, and Storage
- RAN Selection

The Terminal Reconfiguration Decision and Control function makes decisions on Terminal reconfiguration. After making these decisions, the Terminal Reconfiguration Decision and Control function sends corresponding reconfiguration requests to the TRC.

The Information Extraction, Collection, and Storage function receives, processes, and stores the following context information:

- Terminal context information
- RAN context information

Terminal context information is received from the TMC periodically and/or by request and/or on event-basis.

Terminal context information regarding other Terminals is received from the NRM periodically and/or by request and/or on event-basis.

RAN context information is received from the NRM periodically and/or by request and/or on event-basis.

The Information Extraction, Collection, and Storage function provides information to functions inside the TRM.

The Information Extraction, Collection, and Storage function forwards terminal context information to the NRM.

The RAN Selection function selects RANs for exchanging radio resource selection policies and context information between the NRM and the TRM through the radio enabler. Radio resource selection policies are sent from NRM to TRM. From NRM to TRM, RAN context information and terminal context information are sent. From TRM to NRM, terminal context information is sent.

The Terminal Reconfiguration Decision and Control and the RAN Selection functions cooperate during their operation. They represent different aspects of decision making and reconfiguration. During their operation, these functions use information from the Information Extraction, Collection, and Storage function.

The Terminal Reconfiguration Decision and Control function makes reconfiguration decisions within the framework determined by the received radio resource selection policies.

6.2.3 Interfaces of NRM and TRM functions

6.2.3.1 NRM interfaces

The following interfaces between the NRM and other IEEE 1900.4 entities on network side are defined (see Figure 4):

- Interface between Network Reconfiguration Decision and Control function and OSM

- Interface between Network Reconfiguration Decision and Control function and RRC
- Interface between Information Extraction, Collection, and Storage function and RMC

Interface between Network Reconfiguration Decision and Control function and OSM is used to transmit the following:

- From OSM to Network Reconfiguration Decision and Control function:
  - Spectrum assignment policies
- From Network Reconfiguration Decision and Control function to OSM:
  - NRM spectrum assignment decisions

Interface between Information Extraction, Collection, and Storage function and RMC is used to transmit the following:

- From Information Extraction, Collection, and Storage function to RMC:
  - RAN context information requests
- From RMC to Information Extraction, Collection, and Storage function:
  - RAN context information

Interface between Network Reconfiguration Decision and Control function and RRC is used to transmit the following:

- From Network Reconfiguration Decision and Control function to RRC:
  - RAN reconfiguration requests
- From RRC to Network Reconfiguration Decision and Control function:
  - RAN reconfiguration responses

If NRM is implemented in a distributed manner, that is, there are several NRMs, and there is interface between these NRMs, the following interfaces are additionally defined (not shown in Figure 4):

- Interface between Information Extraction, Collection, and Storage functions of several NRMs
- Interface between Network Reconfiguration Decision and Control functions of several NRMs
- Interface between Policy Derivation functions of several NRMs

Interface between Information Extraction, Collection, and Storage functions of several NRMs is used to exchange the following context information:

- RAN context information
- Terminal context information

Interface between Network Reconfiguration Decision and Control functions of several NRMs is used to exchange the following:

- Spectrum assignment policies
- RAN reconfiguration decisions

Interface between Policy Derivation functions of several NRMs is used to exchange the following:

- Radio resource selection policies

An interface between the RAN Selection function in the NRM and the RAN Selection function in the TRM is defined (see Figure 4). The RAN Selection functions distribute on this interface the radio resource selection policies to a specific defined group of Terminals, based on location and/or radio resource usage volume.

This interface is used by the NRM to transmit the following to the TRM:

- Radio resource selection policies
- RAN context information
- Terminal context information

This interface is used by the NRM to receive the following by the TRM:

- Terminal context information

6.2.3.2 TRM interfaces

The following interfaces between TRM and other IEEE 1900.4 entities on terminal side are defined (see Figure 4):

- Interface between Information Extraction, Collection, and Storage function and TMC
- Interface between Terminal Reconfiguration Decision and Control function and TRC

Interface between Information Extraction, Collection, and Storage function and TMC is used to transmit the following:

- From Information Extraction, Collection, and Storage function to TMC:
    - Terminal context information requests
- From TMC to Information Extraction, Collection, and Storage function:
    - Terminal context information

Interface between Terminal Reconfiguration Decision and Control function and TRC is used to transmit the following:

- From Terminal Reconfiguration Decision and Control function to TRC:
    - Terminal reconfiguration requests
- From TRC to Terminal Reconfiguration Decision and Control function:
    - Terminal reconfiguration responses

Interface between RAN Selection function in TRM and RAN Selection function in NRM is defined (see Figure 4). The RAN Selection functions distribute on this interface the radio resource selection policies to specific defined group of Terminals, based on location and/or radio resource usage volume.

This interface is used by the TRM to transmit the following to the NRM:

- Terminal context information

This interface is used by the TRM to receive the following from the NRM:

- Radio resource selection policies

- RAN context information
- Terminal context information

The system architecture defined in this standard can be deployed in different ways. Concrete deployment examples are given in Annex E, showing IEEE 1900.4 entities, interfaces, and SAPs.

7. Information model

7.1 Introduction

IEEE 1900.4 uses an information model based on an object-oriented approach, whereby given that CWN and Terminals are controlled by an IEEE 1900.4 system, they are viewed as the two sets of managed objects.

To this end, the terminal-related classes abstract the user, application, device, and radio resource selection policy concepts, for instance structuring different profiles, capabilities, and measurements related to the Terminal.

The CWN-related classes present an abstract view of the CWN, capturing the operator and RANs, where operator concept includes assigned channels, regulatory rules, and spectrum assignment policies and RAN concept includes BSs and cells.

It must be noted that the presented conceptual abstraction is fully aligned to the scope of this standard. For example, **Application** class in the hierarchy of terminal-related classes does not incorporate generic application attributes; rather, it only incorporates those that have been identified within the standard scope.

Policy information as represented/abstracted by policy classes is a fundamental part of this standard ensuring the communication of policies to NRMs and TRMs to define framework of their operation. This standard therefore presents a concrete representation of events that trigger policies activation and execution, the conditions within which policies must act, and the precise actions that must be undertaken should, for example, a Terminal is found to be violating a policy.

In summary, the information model classes are grouped into the following categories:

- Common base class
- Policy classes
- Terminal-related classes
- CWN-related classes

For each of these categories, the simplified UML (see UML Version 2.1.2) diagrams are described in this clause. Detailed UML diagrams, as well as, attributes and methods of each class are given in precise detail in Annex B and Annex C. Annex D defines utility classes.

The rest of this clause is organized as follows. Subclause 7.2 specifies information modeling approach used in this standard, while 7.3 describes information model classes.

7.2 Information modeling approach

In order to fulfill the requirements, to keep the necessary level of information abstraction, to be prepared for future extensions, and to be easily used by different tools, the following properties have been taken into account for the information modeling:

a) The information model is developed in an extensible form in order to accommodate future radio access technologies and allow for custom extensions to existing data models.

b) The information model uses an object-oriented approach.

c) The information model supports sufficiently simple relationships between different classes.

d) The information model allows for inclusion of both uniform and non-uniform data structures (e.g., lists).

e) The information model allows for definition of new abstract data types to describe the information model items.

f) The information model allows providing information items allowing for specification of precision and accuracy.

g) The information model includes exclusivity or consistency relationships between objects to determine conflicts (for instance whether two different channels or radio technologies can be monitored at the same time).

h) The information model provides means for unique identification of managed objects.

i) The information model utilizes platform-independent unambiguous information/data type definitions.

j) The information model allows for inclusion of information about information objects distribution (e.g., to identify the targeted nodes in a multicasting case).

k) The information model is open to incorporate

- Corresponding information elements towards developing a shared knowledge framework about the information objects themselves. Such framework may include information about the updates, status etc.
- A notifications list, such as configuration changes, threshold crossings etc to align the shared knowledge framework.
- Additional information elements to ensure alternative information retrieval for supporting an efficient retrieval mechanism to obtain performance, quality-of-service and related information and measurements data.
- Information elements that can provide value (instantiate) through mechanisms such as statistical operations to reduce data transfers.
- Managed objects in order to coordinate the measurements scheduling.

7.3 Information model classes

The IEEE 1900.4 information model classes are depicted together with the relations among them in order to present the breaking down of the conceptualization in the adopted several levels of abstraction/details. The corresponding cardinality information has been also included.

7.3.1 Common base class

Common base class is shown in Figure 5.

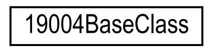


Figure 5—Common base class

The following common base class is defined:

— **19004BaseClass**
This class comprises basic properties to be supported by all objects of the IEEE 1900.4 information model. It is considered as an abstract class that is only used for inheritance. The properties supported are, e.g., an attribute representing the class name of an instance, and three generic events that an instance of this class can report to a managing system. These three events are: creation of a new instance of this class, deletion of an instance of this class, change of an attribute value in an instance of this class.

7.3.2 Policy classes

Policy classes are shown in Figure 6.

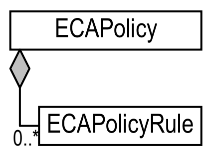


Figure 6—Policy classes

The following policy classes are defined:

— **ECA Policy**
This class is used to describe policies of type Event-Condition-Action (ECA). An instance of this class comprises a set of Event-Condition-Action rules that have to be obeyed when applying the policy.

— **ECA Policy Rule**
An instance of this class describes (in terms of three attributes)
a) an event that triggers the evaluation of the policy condition
b) a condition that shall be fulfilled before applying the policy action
c) the action that has to be performed if the event has occurred and the condition is fulfilled
All these attributes refer to information entities that are available in the system that applies the policy.

7.3.3 Terminal-related classes

A simplified UML diagram of terminal-related classes is shown in Figure 7. A detailed UML diagram of terminal-related classes is available in Annex B.

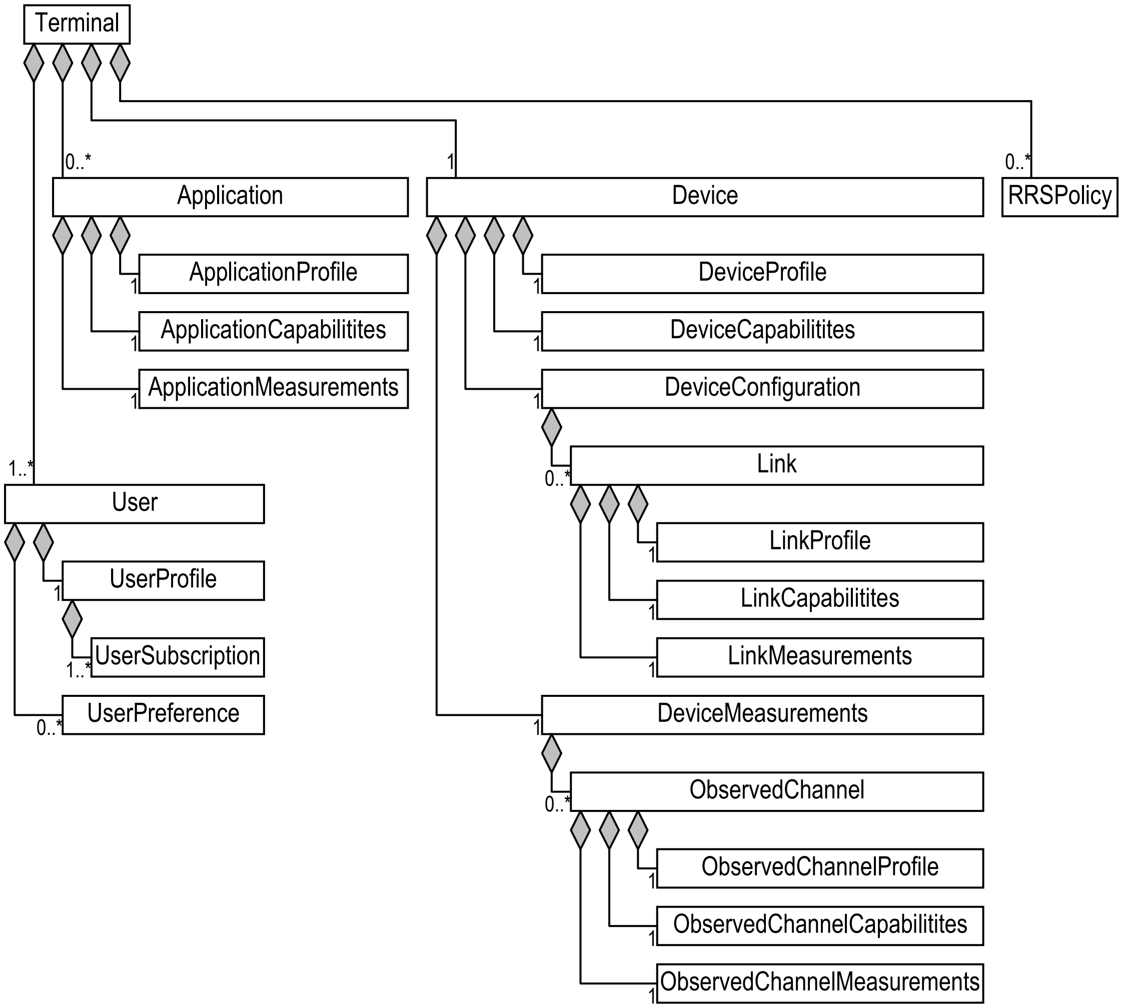


Figure 7—Terminal-related classes

The following terminal-related classes are defined:

- **Terminal**
  The instance of this class contains instances of all terminal-related classes using composition.
  - **User**
    This class describes information related to a user of the Terminal. Each instance of **Terminal** class can have one or several instances of **User** class as members.
    - **User Profile**
      This class contains general information about one user of the Terminal, for example, user ID. Each instance of **User** class can have only one instance of **User Profile** class as a member.
      - **User Subscription**
        This class contains information about one subscription of the user. It describes which RANs/services the user has been subscribed in and what is the associated cost. Each instance of **User Profile** class can have one or several instances of **User Subscription** class as members.

- **User Preference**
  This class describes in a formalized form one preference of the user, for example, preferred operator and radio interface, perceived audio/image/video quality, maximum cost, minimum data rate, etc. Each instance of **User** class can have zero or several instances of **User Preference** class as members.

— **Application**
This class describes one currently active application. Each instance of **Terminal** class can have zero or several instances of **Application** class as members.

- **Application Profile**
  This class contains general information about the application, for example, application ID, traffic class, direction (downlink or uplink), links used to deliver this application, QoS requirements etc. Each instance of **Application** class can have only one instance of **Application Profile** class as a member.

- **Application Capabilities**
  This class contains information about measurements (instantaneous measurement data and performance statistics derived from this data) supported by this application, for example, delay, loss, and bandwidth measurements. Each instance of **Application** class can have only one instance of **Application Capabilities** class as a member.

- **Application Measurements**
  This class contains measurements (instantaneous measurement data and performance statistics derived from this data) performed by this application, such as delay, loss, and bandwidth measurements. Each instance of **Application** class can have only one instance of **Application Measurements** class as a member.

— **Device**
This class describes all radio interface related hardware and software of a Terminal, as well as, measurement information related to radio resources within the Terminal. Each instance of **Terminal** class can have only one instance of **Device** class as a member.

- **Device Profile**
  This class contains general information about the Terminal, for example, Terminal ID. Each instance of **Device** class can have only one instance of **Device Profile** class as a member.

- **Device Capabilities**
  This class contains information about Terminal capabilities including both transmission and measurement capabilities, for example, supported radio interfaces, maximum transmission power, etc. Each instance of **Device** class can have only one instance of **Device Capabilities** class as a member.

- **Device Configuration**
  This class contains information about the current configuration of Terminal. Each instance of **Device** class can have only one instance of **Device Configuration** class as a member.

  - **Link**
    This class contains information about one active connection between Terminal and RANs. Each instance of **Device Configuration** class can have zero or several instances of **Link** class as members.

    - **Link Profile**
      This class contains general information about this active connection, for example, link ID, serving cell ID, channel used, etc. Each instance of **Link** class can have only one instance of **Link Profile** class as a member.

- **Link Capabilities**
  This class contains information about measurements (instantaneous measurement data and performance statistics derived from this data) supported on this active connection, such as block error rate, power, and signal-to-interference-plus-noise-ratio measurements. Each instance of **Link** class can have only one instance of **Link Capabilities** class as a member.

- **Link Measurements**
  This class contains current measurements (instantaneous measurement data and performance statistics derived from this data) related to this active connection, such as block error rate, power, and signal-to-interference-plus-noise-ratio measurements. Each instance of **Link** class can have only one instance of **Link Measurements** class as a member.

- **Device Measurements**
  This class contains current measurements (instantaneous measurement data and performance statistics derived from this data) related to Terminal, for example, battery capacity and Terminal location measurements, as well as, measurements related to observed channels not having active connections with the Terminal. Each instance of **Device** class can have only one instance of **Device Measurements** class as a member.

  - **Observed Channel**
    This class describes one frequency channel that does not have active connection with the Terminal, but is observed by this Terminal. Each instance of **Device Measurements** class can have zero or several instances of **Observed Channel** class as members.

    - **Observed Channel Profile**
      This class contains general information about this frequency channel, for example, channel ID, frequency range, etc. Each instance of **Observed Channel** class can have only one instance of **Observed Channel Profile** class as a member.

    - **Observed Channel Capabilities**
      This class contains information about measurements (instantaneous measurement data and performance statistics derived from this data) supported on this frequency channel, such as interference and load measurements. Each instance of **Observed Channel** class can have only one instance of **Observed Channel Capabilities** class as a member.

    - **Observed Channel Measurements**
      This class contains current measurements (instantaneous measurement data and performance statistics derived from this data) related to this frequency channel, such as interference and load measurements. Each instance of **Observed Channel** class can have only one instance of **Observed Channel Measurements** class as a member.

- **RRS Policy**
  This class describes one radio resource selection (RRS) policy related to this Terminal. Each instance of **Terminal** class can have zero or several instances of **RRS Policy** class as members.

7.3.4 CWN-related classes

A simplified UML diagram of CWN-related classes is shown in Figure 8. A detailed UML diagram of CWN-related classes is available in Annex B.

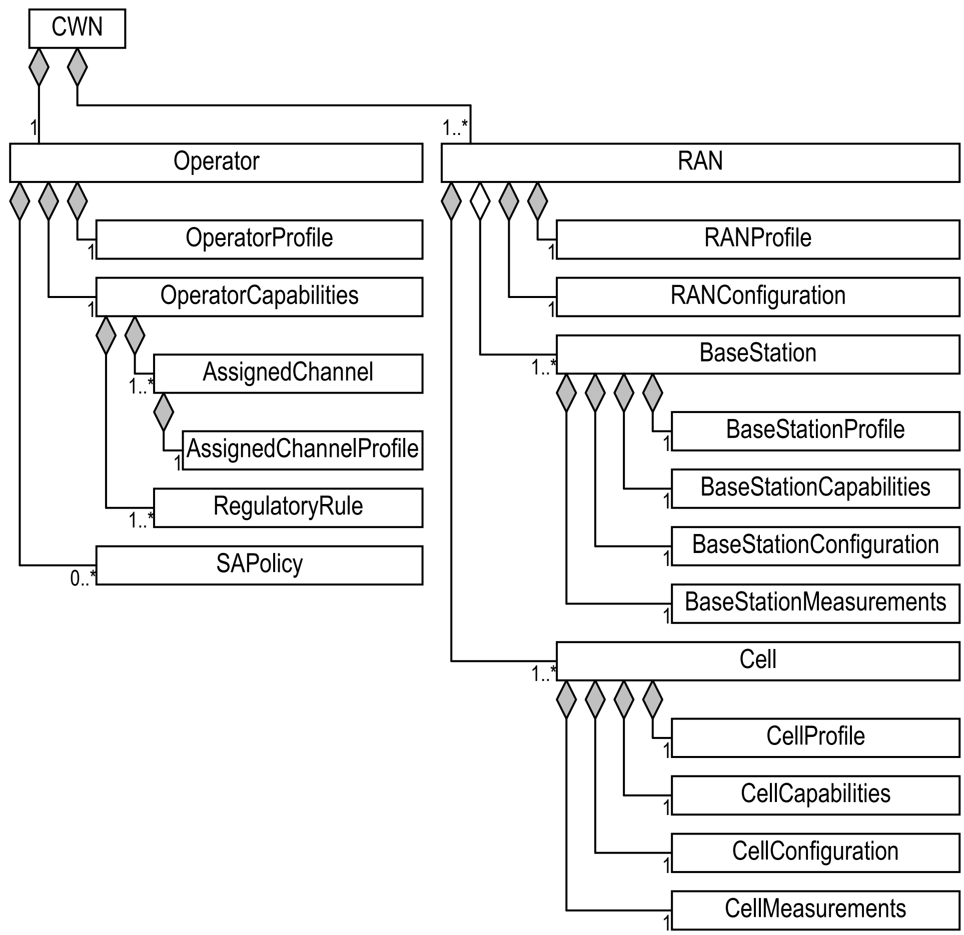


Figure 8—CWN-related classes

The following CWN-related classes are defined:

- **CWN**
  The instance of this class contains instances of all CWN-related classes using composition.
  - **Operator**
    This class describes the operator of this CWN. Each instance of **CWN** class can have only one instance of **Operator** class as a member.
    - **Operator Profile**
      This class contains general information about the operator, for example, operator ID. Each instance of **Operator** class can have only one instance of **Operator Profile** class as a member.
    - **Operator Capabilities**
      This class describes operator capabilities. Each instance of **Operator** class can have only one instance of **Operator Capabilities** class as a member.

- **Assigned Channel**
  This class describes one frequency channel assigned to this operator. Each instance of **Operator** class can have one or several instances of **Assigned Channel** class as members.

  - **Assigned Channel Profile**
    This class contains general information about this frequency channel, for example, frequency channel ID, frequency range, and allowed radio interfaces. Each instance of **Assigned Channel** class can have only one instance of **Assigned Channel Profile** class as a member.

- **Regulatory Rule**
  This class describes in a formalized form one regulatory rule to be applied to one or several assigned channels. Each instance of **Operator Capabilities** class can have one or several instances of **Regulatory Rule** class as members.

- **SA Policy**
  This class describes one spectrum assignment (SA) policy specified by this operator. Each instance of **Operator** class can have zero or several instances of **SA Policy** class as members.

- **RAN**
  This class describes one RAN of this CWN. Each instance of **CWN** class can have one or several instances of **RAN** class as members.

- **RAN Profile**
  This class contains general information about this RAN, for example, RAN ID. Each instance of **RAN** class can have only one instance of **RAN Profile** class as a member.

- **RAN Configuration**
  This class describes current configuration of this RAN, for example, RAN users. Each instance of **RAN** class can have only one instance of **RAN Configuration** class as a member.

- **Base Station**
  This class describes one base station of the RAN. Each instance of **RAN** class can relate to one or several instances of **Base Station** class.

  - **Base Station Profile**
    This class contains general information about this base station, for example, base station ID, vendor, and location. Each instance of **Base Station** class can have only one instance of **Base Station Profile** class as a member.

  - **Base Station Capabilities**
    This class contains information about base station capabilities including both transmission and measurement capabilities, for example, supported radio interfaces, supported channels, transport capability, etc. Each instance of **Base Station** class can have only one instance of **Base Station Capabilities** class as a member.

  - **Base Station Configuration**
    This class contains information about the current configuration of the base station, for example, frequency channels and radio interfaces used. Each instance of **Base Station** class can have only one instance of **Base Station Configuration** class as a member.

  - **Base Station Measurements**
    This class contains current measurements (instantaneous measurement data and

performance statistics derived from this data) performed by this base station, for example, transmission power and load measurements. Each instance of **Base Station** class can have only one instance of **Base Station Measurements** class as a member.

- **Cell**
  This class describes one cell of the base station. Each instance of RAN class can have one or several instances of Cell class as members.
  - **Cell Profile**
    This class contains general information about this cell, for example, cell ID, location, coverage area, etc. Each instance of **Cell** class can have only one instance of **Cell Profile** class as a member.
  - **Cell Capabilities**
    This class contains information about cell capabilities, for example, supported radio interfaces, supported channels, supported measurements, etc. Each instance of **Cell** class can have only one instance of **Cell Capabilities** class as a member.
  - **Cell Configuration**
    This class contains information about the current configuration of the cell, for example, Terminals served and transport service used. Each instance of **Cell** class can have only one instance of **Cell Configuration** class as a member.
  - **Cell Measurements**
    This class contains current measurements (instantaneous measurement data and performance statistics derived from this data) related to this cell, for example, transmission power, cell and traffic loads, throughput, and interference measurements. Each instance of **Cell** class can have only one instance of **Cell Measurements** class as a member.

8. Procedures

8.1 Introduction

This clause describes the IEEE 1900.4 generic procedures.

As described in the reference use cases, three scenarios for heterogeneous wireless environment are considered in this standard:

- Single operator
- Multiple operator 1 (NRM is inside operator)
- Multiple operator 2 (NRM is outside operators)

Subclause 8.2 defines the following IEEE 1900.4 generic procedures:

- Collecting context information (see 8.2.1)
- Generating spectrum assignment policies (see 8.2.2)
- Making spectrum assignment decision (see 8.2.3)
- Performing spectrum access on network side (see 8.2.4)
- Generating radio resource selection policies (see 8.2.5)
- Performing reconfiguration on terminal side (see 8.2.6)

Each of these IEEE 1900.4 generic procedures has three descriptions, corresponding to three reference network architectures.

Based on the IEEE 1900.4 generic procedures defined in 8.2, subclause 8.3 gives examples of realization of the IEEE 1900.4 reference use cases.

Figure 9, Figure 10, and Figure 11 describe how IEEE 1900.4 system architecture (see Clause 6) can be applied to single operator, multiple operator 1 (NRM is inside operator), and multiple operator 2 (NRM is outside operators) scenarios. These figures show differences between these scenarios and are required for understanding the IEEE 1900.4 generic procedures described in this clause. More deployment examples can be found in Annex E.

Figure 9 illustrates the single operator scenario.

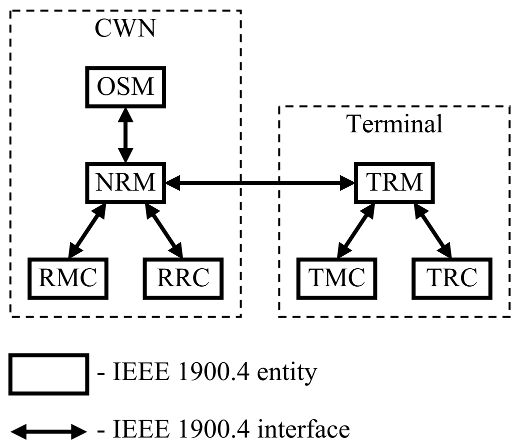


Figure 9—Single operator scenario

Single operator scenario considers one CWN and one Terminal (multiple Terminals are not required to define generic procedures).

The following IEEE 1900.4 entities are used:

- OSM
- NRM
- RMC
- RRC
- TRM
- TMC
- TRC

The following IEEE 1900.4 interfaces are used:

- Interface between OSM and NRM
- Interface between NRM and RMC
- Interface between NRM and RRC

— Interface between NRM and TRM

— Interface between TRM and TMC

— Interface between TRM and TRC

Figure 10 illustrates the multiple operator scenario 1 (NRM is inside operator).

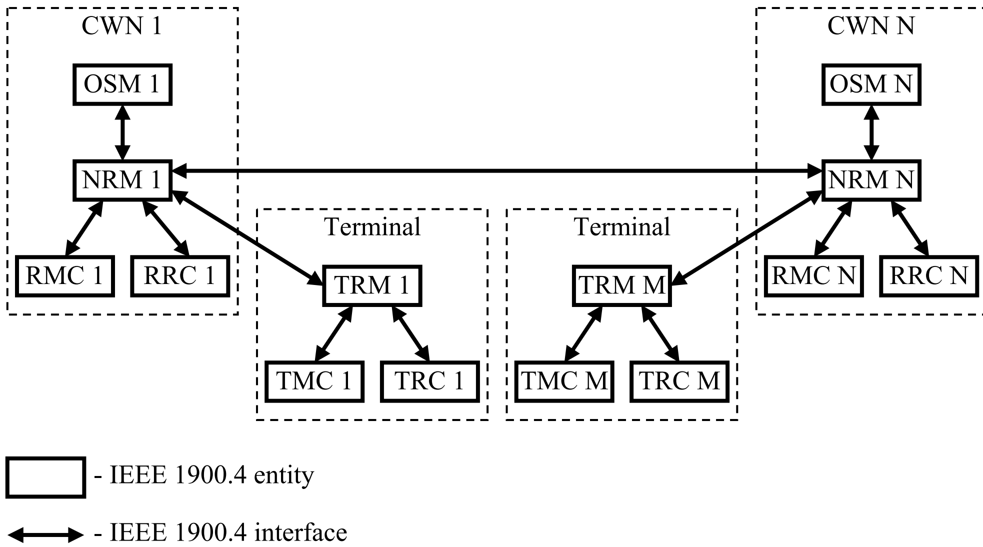


Figure 10—Multiple operator scenario 1 (NRM is inside operator)

Multiple operator scenario 1 (NRM is inside operator) considers multiple CWNs and multiple Terminals. The number of CWNs is denoted as N and the number of Terminals is denoted as M.

The following IEEE 1900.4 entities are used:

— OSM 1, …, OSM N

— NRM 1, …, NRM N

— RMC 1, …, RMC N

— RRC 1, …, RRC N

— TRM 1, …, TRM M

— TMC 1, …, TMC M

— TRC 1, …, TRC M

The following IEEE 1900.4 interfaces are used:

— Interface between OSM i and NRM i, i = 1, …, N

— Interface between NRM i and RMC i

— Interface between NRM i and RRC i

— Interface between NRM i and NRM j, j = 1, …, N, where $i \neq j$

— Interface between NRM i and TRM k, k = 1, …, M

— Interface between TRM k and TMC k

— Interface between TRM k and TRC k

Figure 11 illustrates the multiple operator scenario 2 (NRM is outside operators).

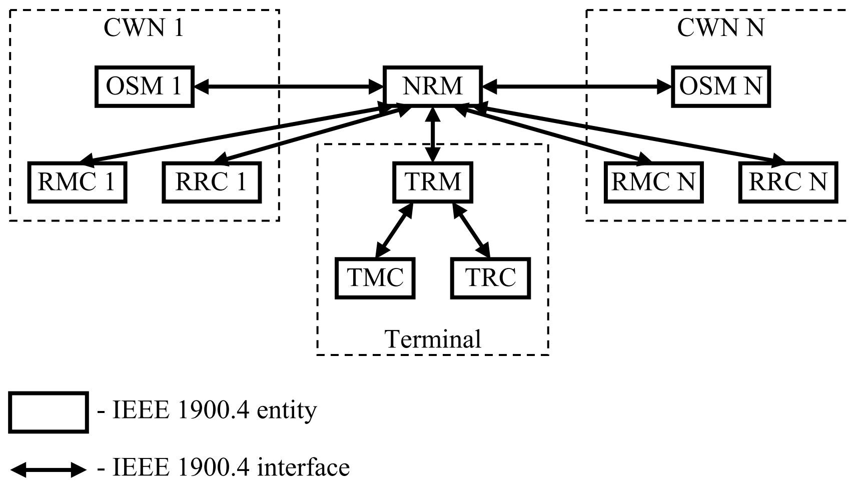


Figure 11—Multiple operator scenario 2 (NRM is outside operators)

Multiple operator scenario 2 (NRM is outside operators) considers multiple CWNs and one Terminal (multiple Terminals are not required to define generic procedures). The number of CWNs is denoted as N.

The following IEEE 1900.4 entities are used in this reference network architecture:

— OSM 1, …, OSM N

— NRM

— RMC 1, …, RMC N

— RRC 1, …, RRC N

— TRM

— TMC

— TRC

The following IEEE 1900.4 interfaces are used in this reference network architecture:

— Between OSM i and NRM, i = 1, …, N

— Between NRM and RMC i

— Between NRM and RRC i

— Between NRM and TRM

— Between TRM and TMC

— Between TRM and TRC

8.2 Generic procedures

8.2.1 Collecting context information

8.2.1.1 Single operator

Collecting context information procedure for single operator scenario is shown in Figure 12.

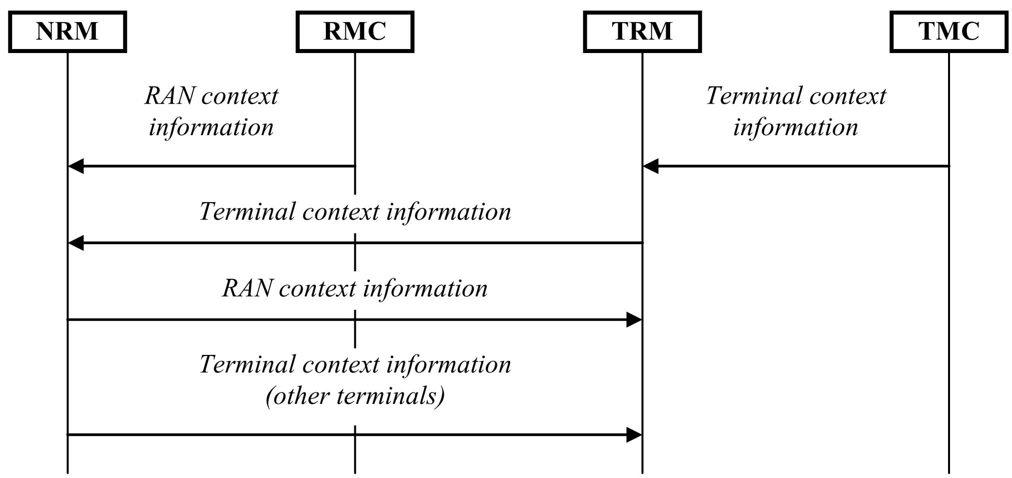


Figure 12—Collecting context information procedure for single operator scenario

Collecting context information procedure for single operator scenario is as follows:

- TMC forwards terminal context information about its Terminal to TRM
- RMC forwards RAN context information about its RANs to NRM
- TRM sends terminal context information about its Terminal to NRM
- NRM sends RAN context information about its RANs to TRM
- NRM may send terminal context information about other Terminals to TRM

8.2.1.2 Multiple operator 1 (NRM is inside operator)

Collecting context information procedure for multiple operator scenario 1 (NRM is inside operator) is shown in Figure 13.

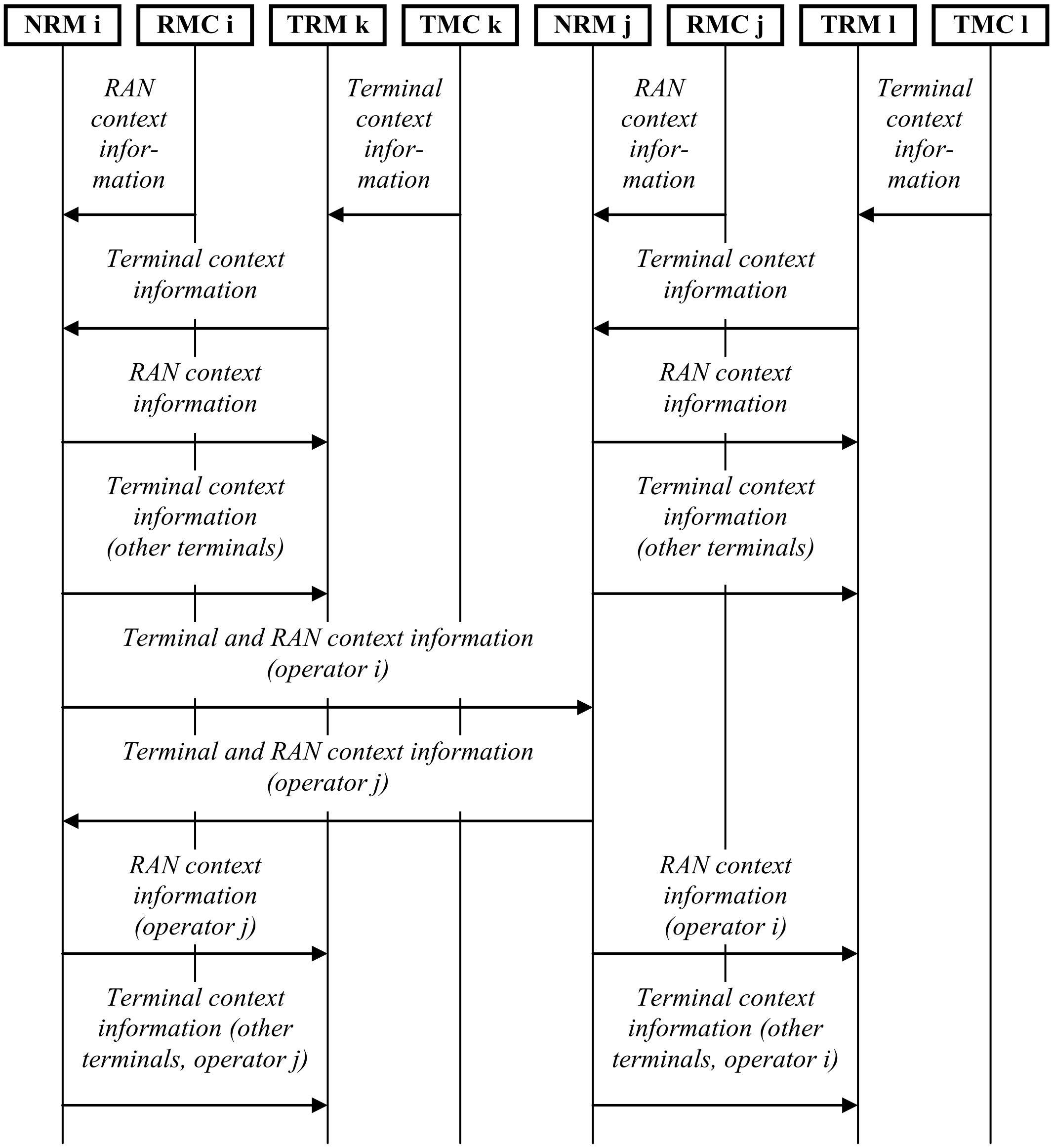


Figure 13—Collecting context information procedure for multiple operator scenario 1 (NRM is inside operator)

Collecting context information procedure for multiple operator scenario 1 (NRM is inside operator) is as follows:

- TMC k forwards terminal context information about its Terminal to TRM k
- RMC i forwards RAN context information about RANs of operator i to NRM i
- TRM k sends terminal context information about its Terminal to NRM i
- NRM i sends RAN context information about RANs of operator i to TRM k
- NRM i may send terminal context information about other Terminals of operator i to TRM k
- TMC l forwards terminal context information about its Terminal to TRM l
- RMC j forwards RAN context information about RANs of operator j to NRM j

- TRM l sends terminal context information about its Terminal to NRM j
- NRM j sends RAN context information about RANs of operator j to TRM l
- NRM j may send terminal context information about other Terminals of operator j to TRM l
- NRM i sends terminal and RAN context information about Terminals and RANs of operator i to NRM j
- NRM j sends terminal and RAN context information about Terminals and RANs of operator j to NRM i
- NRM i sends RAN context information about RANs of operator j to TRM k
- NRM i may send terminal context information about Terminals of operator j to TRM k
- NRM j sends RAN context information about RANs of operator i to TRM l
- NRM j may send terminal context information about Terminals of operator i to TRM l

8.2.1.3 Multiple operator 2 (NRM is outside operators)

Collecting context information procedure for multiple operator scenario 2 (NRM is outside operators) is shown in Figure 14.

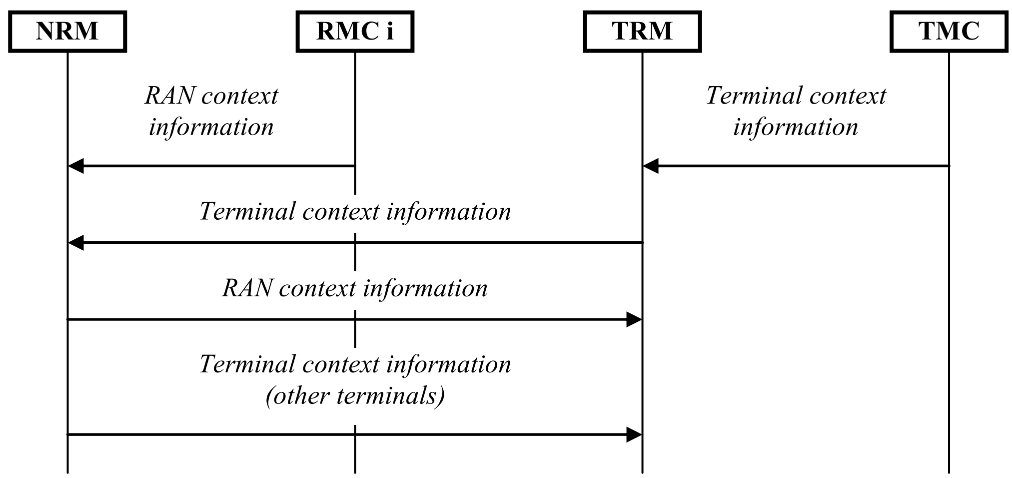


Figure 14—Collecting context information procedure for multiple operator scenario 2 (NRM is outside operators)

Collecting context information procedure for multiple operator scenario 2 (NRM is outside operators) is as follows:

- TMC forwards terminal context information about its Terminal to TRM
- RMC i forwards RAN context information about RANs of operator i to NRM
- TRM sends terminal context information about its Terminal to NRM
- NRM sends RAN context information about RANs of operators 1, …, N to TRM
- NRM may send terminal context information about other Terminals to TRM

8.2.2 Generating spectrum assignment policies

8.2.2.1 Single operator

Generating spectrum assignment policies procedure for single operator scenario is shown in Figure 15.

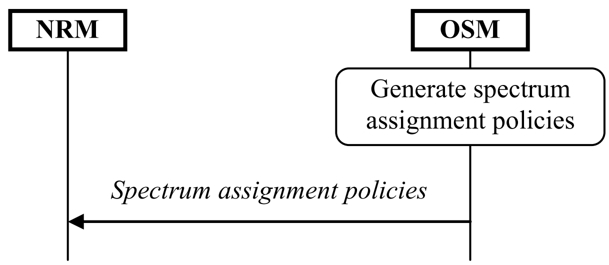


Figure 15—Generating spectrum assignment policies procedure for single operator scenario

Generating spectrum assignment policies procedure for single operator scenario is as follows:

- OSM generates spectrum assignment policies
- OSM sends spectrum assignment policies to its NRM

8.2.2.2 Multiple operator 1 (NRM is inside operator)

Generating spectrum assignment policies procedure for multiple operator scenario 1 (NRM is inside operator) is shown in Figure 16.

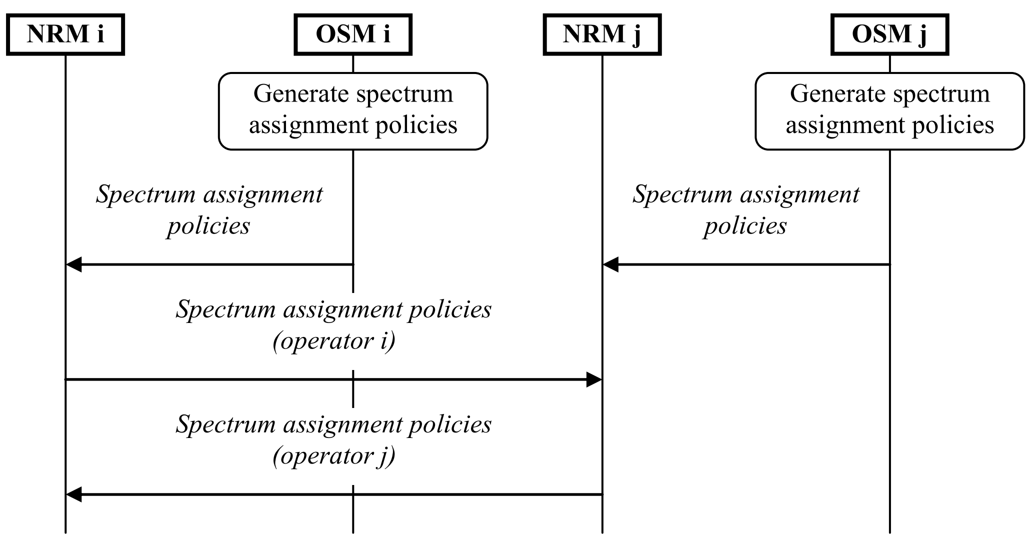


Figure 16—Generating spectrum assignment policies procedure for multiple operator scenario 1 (NRM is inside operator)

Generating spectrum assignment policies procedure for multiple operator scenario 1 (NRM is inside operator) is as follows:

- OSM i generates spectrum assignment policies on behalf of operator i
- OSM i sends spectrum assignment policies to NRM i
- OSM j generates spectrum assignment policies on behalf of operator j
- OSM j sends spectrum assignment policies to NRM j
- NRM i sends spectrum assignment policies of operator i to NRM j
- NRM j sends spectrum assignment policies of operator j to NRM i

8.2.2.3 Multiple operator 2 (NRM is outside operators)

The generating spectrum assignment policies procedure for multiple operator scenario 2 (NRM is outside operators) is shown in Figure 17.

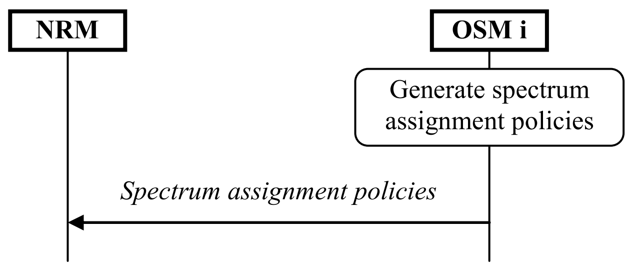


Figure 17—Generating spectrum assignment policies procedure for multiple operator scenario 2 (NRM is outside operators)

The generating spectrum assignment policies procedure for multiple operator scenario 2 (NRM is outside operators) is as follows:

- OSMs 1, …, N generate spectrum assignment policies on behalf of operators 1, …, N
- OSMs 1, …, N send spectrum assignment policies to its NRM

8.2.3 Making spectrum assignment decision

8.2.3.1 Single operator

The making spectrum assignment decision procedure for single operator scenario is shown in Figure 18.

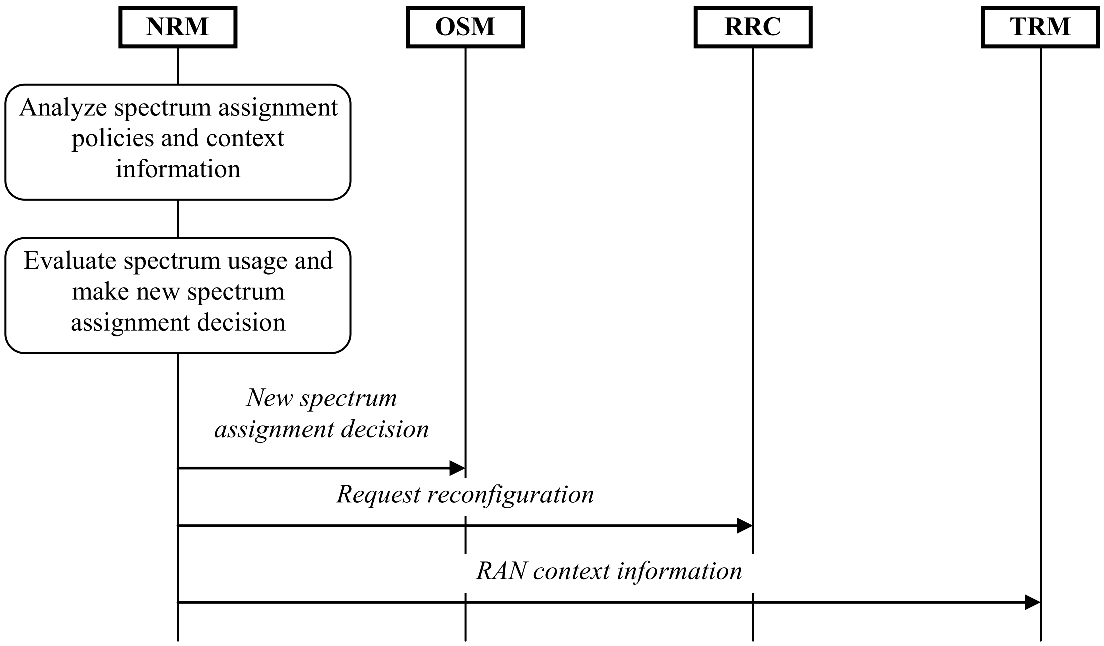


Figure 18 —Making spectrum assignment decision procedure for single operator scenario

The making spectrum assignment decision procedure for single operator scenario is as follows:

- NRM analyzes spectrum assignment policies and context information
- NRM evaluates spectrum usage and makes new spectrum assignment decision
- NRM reports new spectrum assignment decision to its OSM
- NRM requests corresponding reconfiguration of its RANs to its RRC
- NRM sends new RAN context information to its TRM

8.2.3.2 Multiple operator 1 (NRM is inside operator)

The making spectrum assignment decision procedure for multiple operator scenario 1 (NRM is inside operator) is shown in Figure 19.

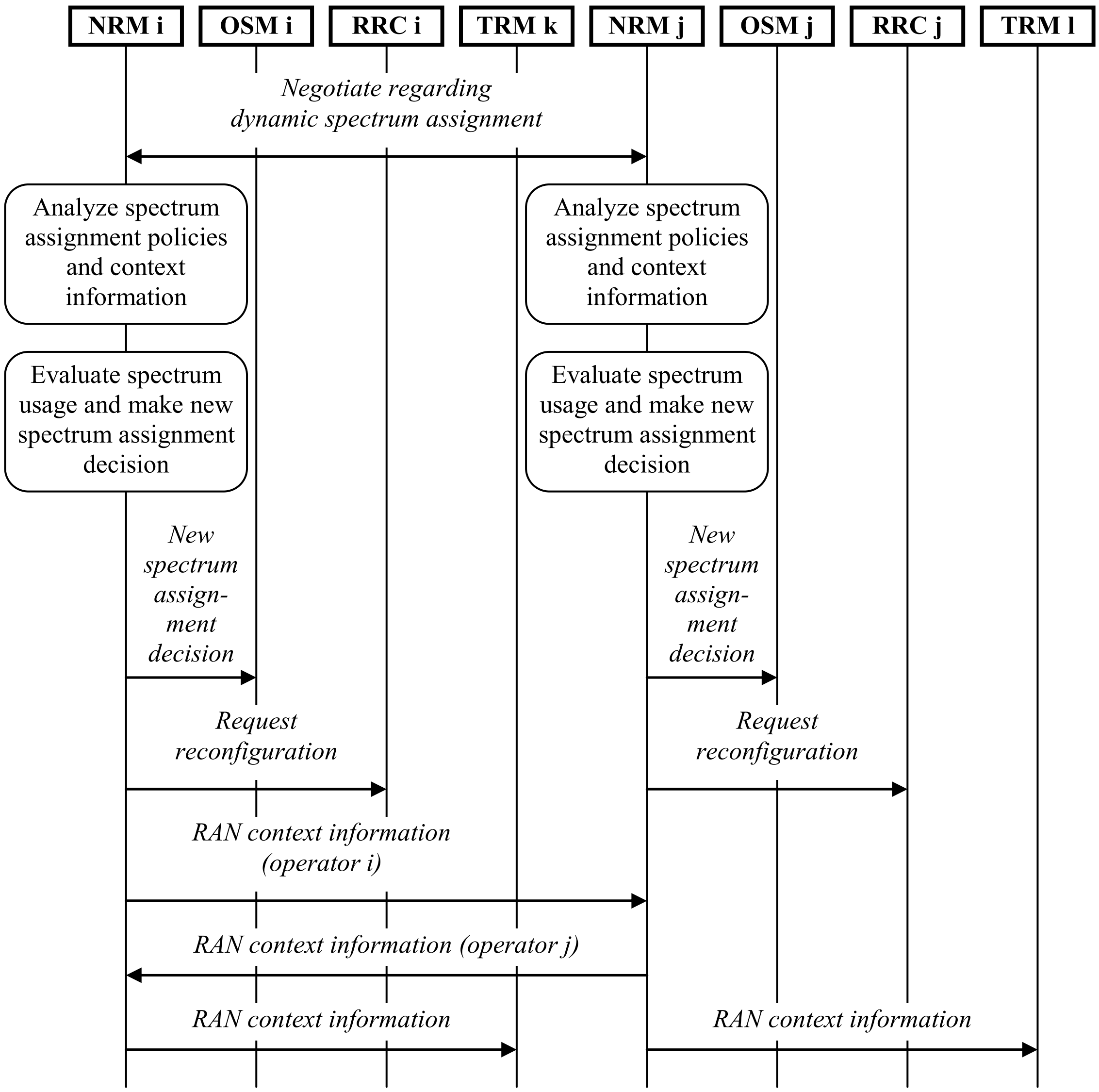


Figure 19—Making spectrum assignment decision procedure for multiple operator scenario 1 (NRM is inside operator)

The making spectrum assignment decision procedure for multiple operator scenario 1 (NRM is inside operator) is as follows:

— NRM i, and NRM j negotiate regarding dynamic spectrum assignment
— NRM i analyzes spectrum assignment policies and context information
— NRM i evaluates spectrum usage and makes new spectrum assignment decision
— NRM i reports new spectrum assignment decision to OSM i
— NRM i requests corresponding reconfiguration of RANs of operator i to RRC i
— NRM j analyzes spectrum assignment policies and context information
— NRM j evaluates spectrum usage and makes new spectrum assignment decision

— NRM j reports new spectrum assignment decision to OSM j

— NRM j requests corresponding reconfiguration of RANs of operator j to RRC j

— NRM i sends new RAN context information about RANs of operator i to NRM j

— NRM j sends new RAN context information about RANs of operator j to NRM i

— NRM i sends new RAN context information to TRM k

— NRM j sends new RAN context information to TRM l

8.2.3.3 Multiple operator 2 (NRM is outside operators)

The making spectrum assignment decision procedure for multiple operator scenario 2 (NRM is outside operators) is shown in Figure 20.

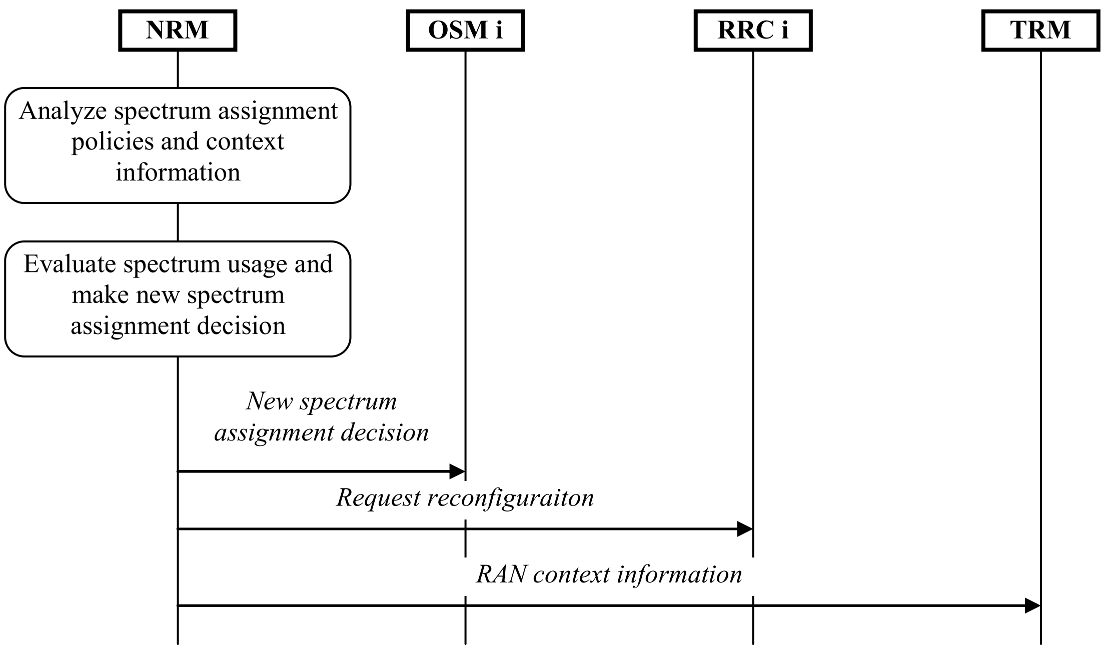


Figure 20 —Making spectrum assignment decision procedure for multiple operator scenario 2 (NRM is outside operators)

Making spectrum assignment decision procedure for multiple operator scenario 2 (NRM is outside operators) is as follows:

— NRM analyzes spectrum assignment policies and context information

— NRM evaluates spectrum usage and makes new spectrum assignment decision

— NRM reports new spectrum assignment decision to OSM 1, …, N

— NRM requests corresponding reconfiguration of RANs to RRC i

— NRM sends new RAN context information to its TRM

8.2.4 Performing spectrum access on network side

8.2.4.1 Single operator

The performing spectrum access on network side procedure for single operator scenario is shown in Figure 21.

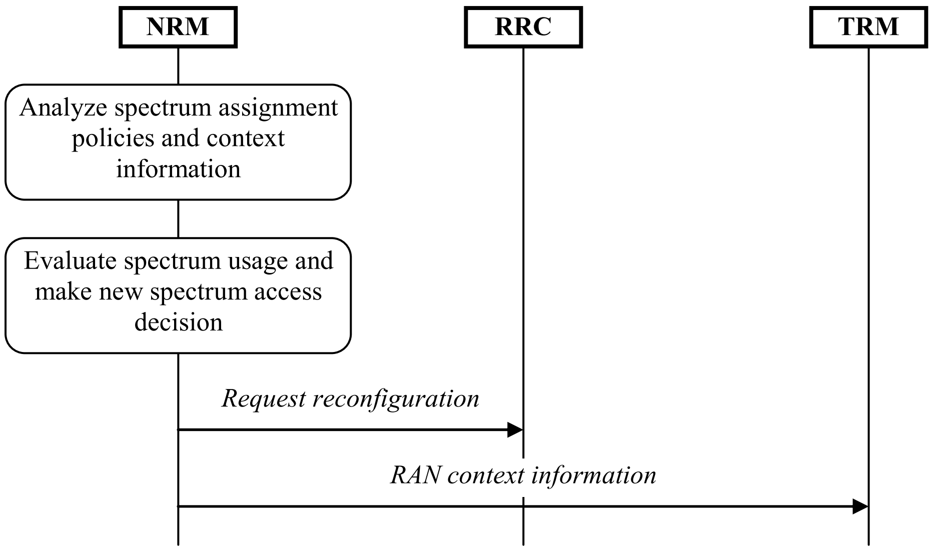


Figure 21 —Performing spectrum access on network side procedure for single operator scenario

The performing spectrum access on network side procedure for single operator scenario is as follows:

- NRM analyzes spectrum assignment policies and context information
- NRM evaluates spectrum usage and makes new spectrum access decision
- NRM requests corresponding reconfiguration of its RANs to its RRC
- NRM sends new RAN context information to its TRM

8.2.4.2 Multiple operator 1 (NRM is inside operator)

The performing spectrum access on network side procedure for multiple operator scenario 1 (NRM is inside operator) is shown in Figure 22.

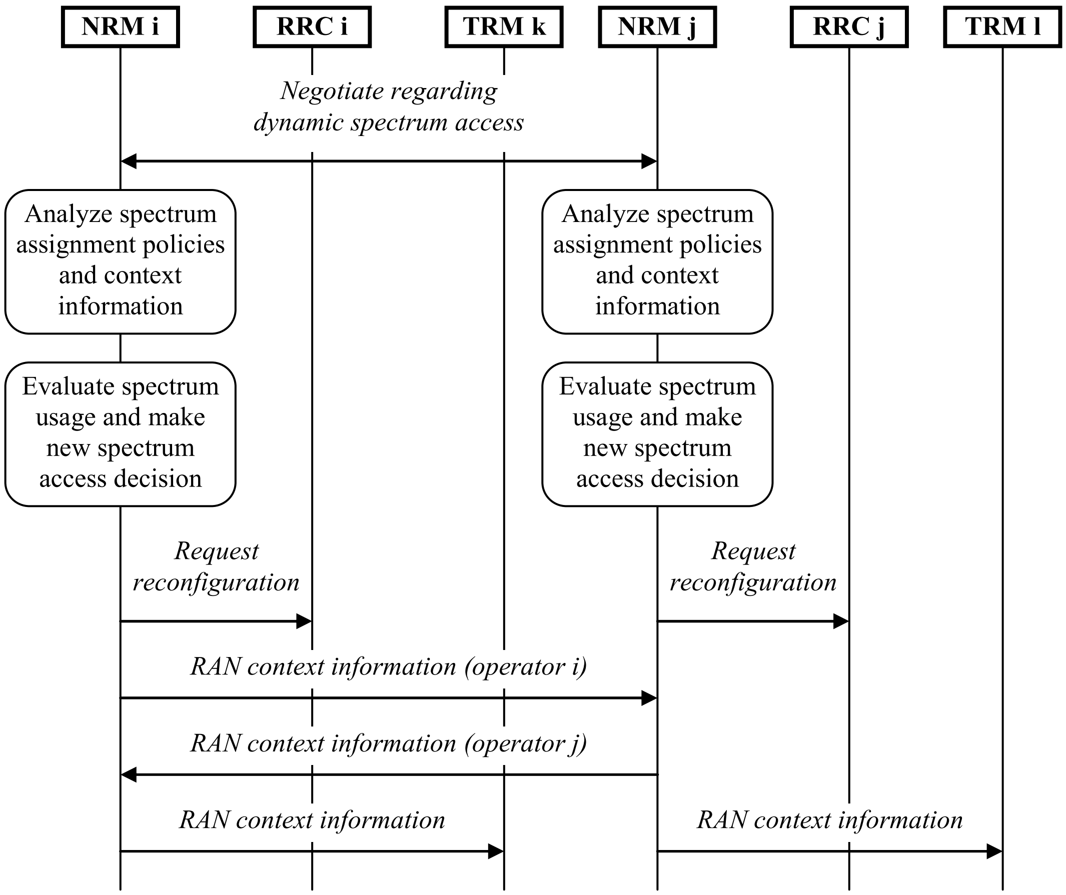


Figure 22—Performing spectrum access on network side procedure for multiple operator scenario 1 (NRM is inside operator)

The performing spectrum access on network side procedure for multiple operator scenario 1 (NRM is inside operator) is as follows:

- NRM i and NRM j negotiate regarding dynamic spectrum sharing
- NRM i analyzes spectrum assignment policies and context information
- NRM i evaluates spectrum usage and makes new spectrum access decision
- NRM i requests corresponding reconfiguration of RANs of operator i to RRC i
- NRM j analyzes spectrum assignment policies and context information
- NRM j evaluates spectrum usage and makes new spectrum access decision
- NRM j requests corresponding reconfiguration of RANs of operator j to RRC j
- NRM i sends new RAN context information about RANs of operator i to NRM j
- NRM j sends new RAN context information about RANs of operator j to NRM i
- NRM i sends new RAN context information to TRM k
- NRM j sends new RAN context information to TRM l

8.2.4.3 Multiple operator 2 (NRM is outside operators)

The performing spectrum access on network side procedure for multiple operator scenario 2 (NRM is outside operators) is shown on Figure 23.

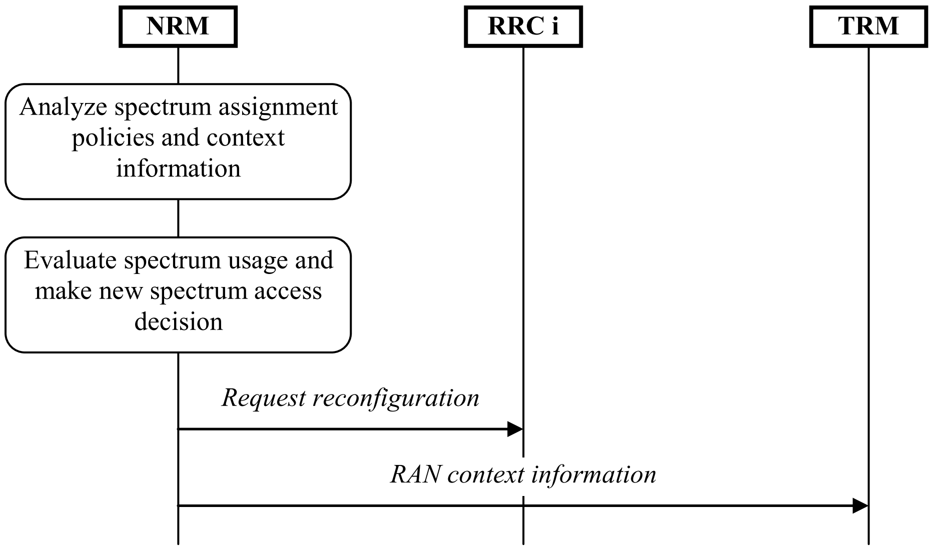


Figure 23—Performing spectrum access on network side procedure for multiple operator scenario 2 (NRM is outside operators)

The performing spectrum access on network side procedure for multiple operator scenario 2 (NRM is outside operators) is as follows:

- NRM analyzes spectrum assignment policies and context information
- NRM evaluates spectrum usage and makes new spectrum access decision
- NRM requests corresponding reconfiguration of RANs of corresponding operators i to corresponding RRCs i
- NRM sends new RAN context information to its TRM

8.2.5 Generating radio resource selection policies

8.2.5.1 Single operator

The generating radio resource selection policies procedure for single operator scenario is shown in Figure 24.

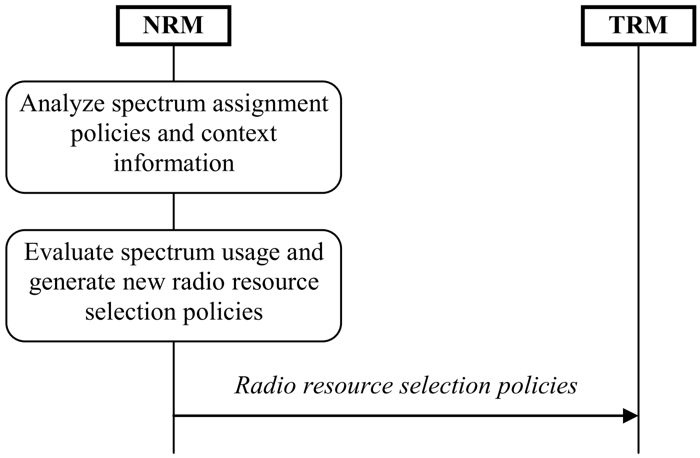


Figure 24—Generating radio resource selection policies procedure for single operator scenario

Generating radio resource selection policies procedure for single operator scenario is as follows:

- NRM analyzes spectrum assignment policies and context information
- NRM evaluates spectrum usage and generates new radio resource selection policies
- NRM sends radio resource selection policies to its TRM.

8.2.5.2 Multiple operator 1 (NRM is inside operator)

The generating radio resource selection policies procedure for multiple operator scenario 1 (NRM is inside operator) is shown in Figure 25.

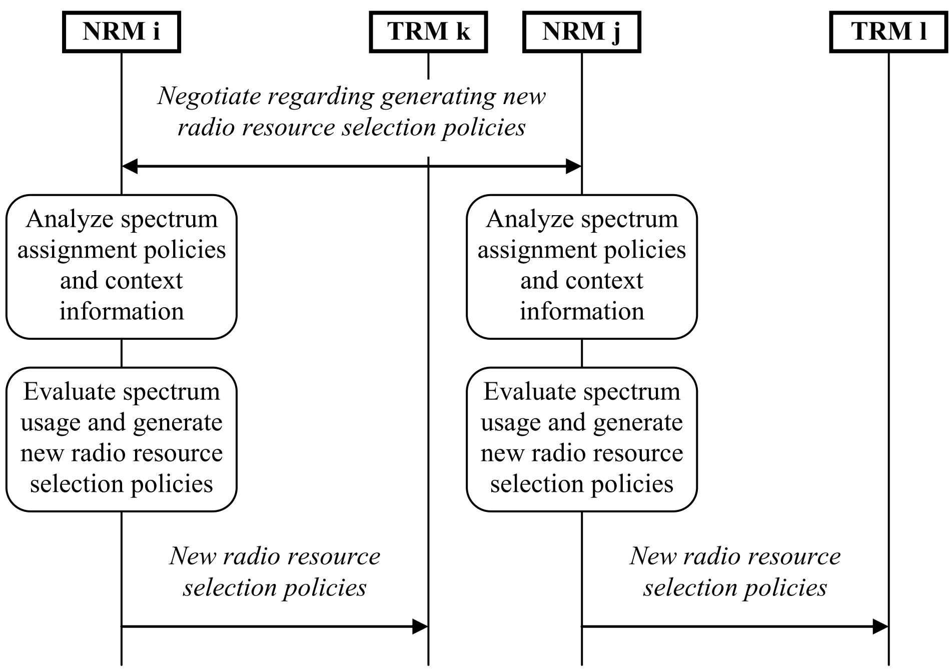


Figure 25—Generating radio resource selection policies procedure for multiple operator scenario 1 (NRM is inside operator)

The generating radio resource selection policies procedure for multiple operator scenario 1 (NRM is inside operator) is as follows:

- NRM i and NRM j negotiate regarding generating new radio resource selection policies
- NRM i analyzes spectrum assignment policies and context information
- NRM i evaluates spectrum usage and generates new radio resource selection policies on behalf of operator i
- NRM i sends radio resource selection policies to its TRM
- NRM j analyzes spectrum assignment policies and context information
- NRM j evaluates spectrum usage and generates new radio resource selection policies on behalf of operator j
- NRM j sends radio resource selection policies to its TRM

8.2.5.3 Multiple operator 2 (NRM is outside operators)

See the procedure for single operator scenario in 8.2.5.1.

8.2.6 Performing reconfiguration on terminal side

8.2.6.1 Single operator

The performing reconfiguration on terminal side procedure for single operator scenario is shown in Figure 26.

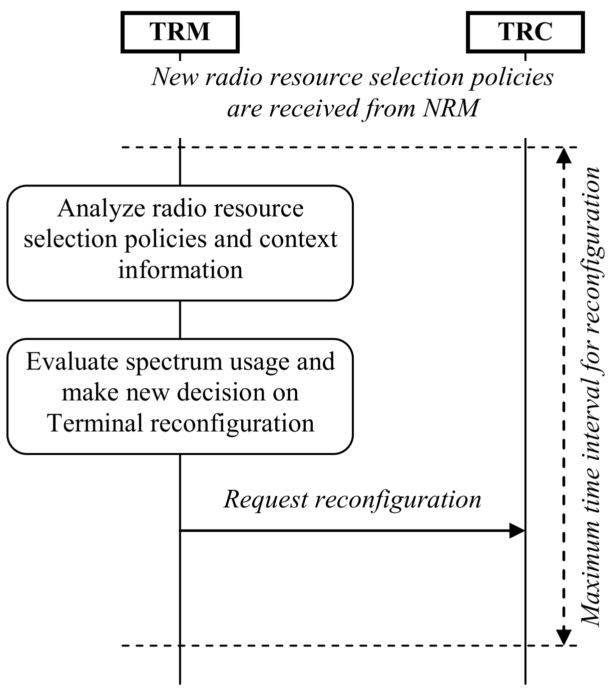


Figure 26—Performing reconfiguration on terminal side procedure for single operator scenario

The performing reconfiguration on terminal side procedure for single operator scenario is as follows:

- Upon reception of new radio resource selection policies from NRM, TRM performs the following:
  - If maximum time interval for reconfiguration is specified, then within this time interval TRM performs the following:
    - TRM analyzes radio resource selection policies and context information
    - TRM evaluates spectrum usage and makes new decision on its Terminal reconfiguration
    - TRM requests corresponding reconfiguration of its Terminal to its TRC
  - If maximum time interval for reconfiguration is not specified, then TRM performs the same sequence of actions but without time constraint

8.2.6.2 Multiple operator 1 (NRM is inside operator)

See the procedure for single operator scenario in 8.2.6.1.

8.2.6.3 Multiple operator 2 (NRM is outside operators)

See the procedure for single operator scenario in 8.2.6.1.

8.3 Examples of use case realization

8.3.1 Dynamic spectrum assignment

An example of dynamic spectrum assignment use case realization using the defined IEEE 1900.4 generic procedures is shown in Figure 27.

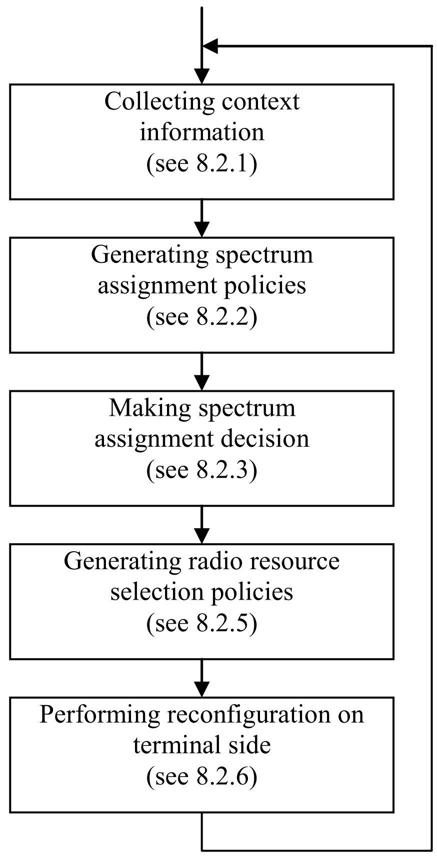


Figure 27 —Example of dynamic spectrum assignment use case realization

Dynamic spectrum assignment use case can be realized as follows using the IEEE 1900.4 procedures:

— Collecting context information procedure is performed (8.2.1)
— Generating spectrum assignment policies procedure is preformed (8.2.2)
— Making spectrum assignment decision procedure is preformed (8.2.3)
— Generating radio resource selection policies procedure is preformed (8.2.5)
— Performing reconfiguration on terminal side procedure is performed (8.2.6)

8.3.2 Dynamic spectrum sharing

An example of dynamic spectrum sharing use case realization using the defined IEEE 1900.4 generic procedures is shown in Figure 28.

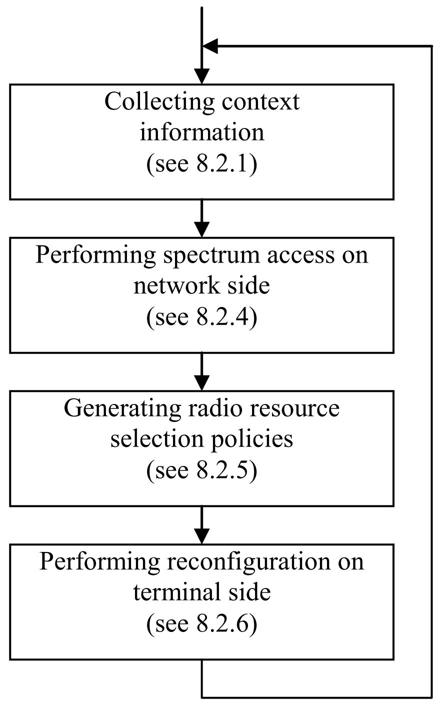


Figure 28—Example of dynamic spectrum sharing use case realization

Dynamic spectrum sharing use case can be realized as follows using the IEEE 1900.4 procedures:

- Collecting context information procedure is performed (8.2.1)
- Performing spectrum access on network side procedure is preformed (8.2.4)
- Generating radio resource selection policies procedure is preformed (8.2.5)
- Performing reconfiguration on terminal side procedure is performed (8.2.6)

8.3.3 Distributed radio resource usage optimization

An example of distributed radio resource usage optimization use case realization using the defined IEEE 1900.4 generic procedures is shown in Figure 29.

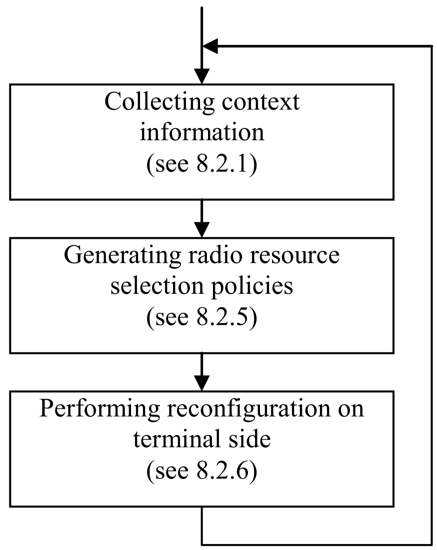


Figure 29—Example of distributed radio resource usage optimization use case realization

Distributed radio resource usage optimization use case can be realized as follows using the IEEE 1900.4 procedures:

- Collecting context information procedure is performed (8.2.1)
- Generating radio resource selection policies procedure is preformed (8.2.5)
- Performing reconfiguration on terminal side procedure is performed (8.2.6)

Annex A

(informative)

Use cases

A.1 Dynamic spectrum assignment

In the dynamic spectrum assignment use case, frequency bands are dynamically assigned to the RANs among the participating networks in order to optimize spectrum usage. In other words, the assigned frequency bands are not fixed, and can be dynamically changed.

Following the dynamic spectrum assignment decisions, corresponding RANs are reconfigured. Following the RANs' reconfiguration, Terminals need to reconfigure correspondingly.

The following three scenarios are defined to illustrate dynamic spectrum assignment use case:

- Single operator scenario
- Multiple operator scenario 1 (NRM is inside operator)
- Multiple operator scenario 2 (NRM is outside operators)

A.1.1 Single operator scenario

The single operator scenario assumes that several frequency bands are assigned to an operator that has several RANs. This operator has flexibility for distributing these frequency bands between its RANs. OSM will enable the operator to evaluate efficiency of current spectrum assignment; however, it cannot facilitate dynamic spectrum reconfiguration for its RAN. NRM introduced in this standard enables dynamic reconfiguration of RANs inside the operator to improve usage of its frequency bands.

A single operator dynamic spectrum assignment example is shown in Figure A.1. The operator operates RAN 1 and RAN 2 in two frequency bands. The frequency band assigned to RAN 1 is overused. The frequency band assigned to RAN 2 is underused. Spectrum usage is unbalanced.

OSM of this operator sends spectrum assignment policy to NRM requesting reconfiguration of its RAN 1 and RAN 2 networks to allow RAN 1 network to use part of RAN 2 frequency band.

NRM requests and controls the corresponding reconfiguration of RAN 1 and RAN 2 networks. Following the RANs' reconfiguration, Terminals need to reconfigure correspondingly.

After reconfiguration, part of RAN 1 starts to use part of RAN 2 frequency band. The usage of frequency bands of the operator is now balanced.

For the operator, this allows reconfiguring its networks to balance usage of its frequency bands.

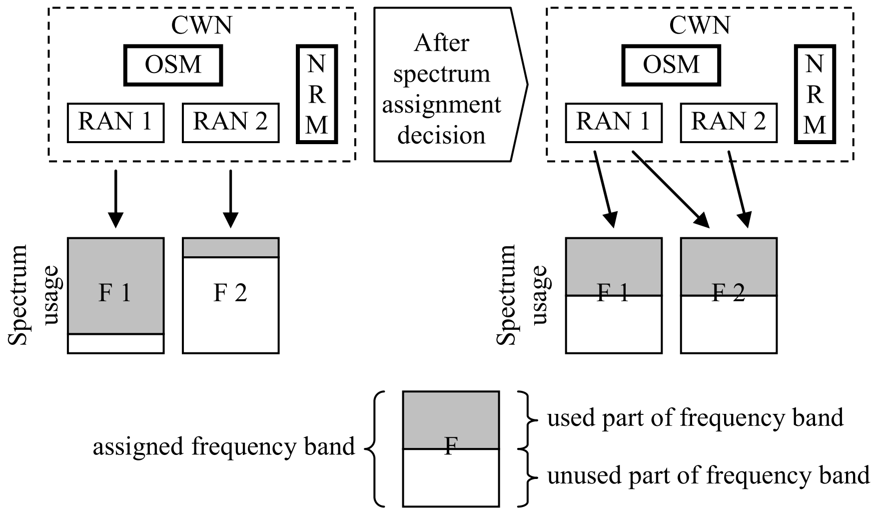


Figure A.1—Dynamic spectrum assignment: single operator scenario

The single operator dynamic spectrum assignment scenario is enabled by NRM. NRM allows reconfiguration of RANs of operator. This improves spectrum usage and increases quality of service for the operator.

The single operator dynamic spectrum assignment scenario is realized as follows:

a) OSM of operator generates spectrum assignment policies and sends them to NRM
b) NRM of operator receives these spectrum assignment policies
c) NRM obtains RAN context information
d) TRMs obtain terminal context information
e) NRM and TRMs exchange context information
f) NRM analyzes spectrum assignment policies and context information
g) NRM evaluates current spectrum assignment inside operator and makes new spectrum assignment decision
h) NRM informs OSM about its spectrum assignment decision
i) NRM requests and controls corresponding reconfiguration of RANs of operator
j) Following the RANs' reconfiguration, Terminals reconfigure correspondingly

A.1.2 Multiple operator scenario 1 (NRM is inside operator)

Multiple operator scenario 1 assumes that several frequency bands are allocated to several operators and operators have some level of flexibility for renting or sharing these frequency bands. OSM evaluates efficiency of current spectrum assignment for each operator. NRM decides dynamic spectrum reconfiguration for RANs of this operator. Cross-operator collaboration is performed via NRMs and/or OSMs of different operators. NRM introduced in this standard enables cross-operator optimization of spectrum usage by performing dynamic spectrum assignment.

A spectrum sharing example is shown in Figure A.2. Operator A and B operate RAN 1 and RAN 2 in their frequency bands. The frequency band assigned to operator A is overused. The frequency band assigned to operator B is underused. Spectrum usage is unbalanced.

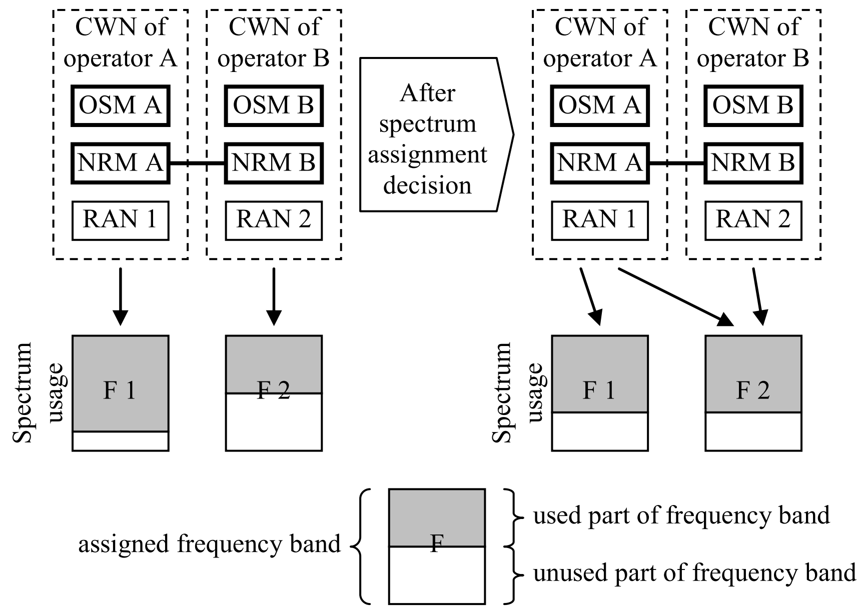


Figure A.2—Dynamic spectrum assignment: multiple operator scenario 1 (NRM is inside operator): spectrum sharing example

OSMs and/or NRMs of operators A and B negotiate to allow operator A to use frequency band of operator B. After negotiation, NRM of operator A requests and controls reconfiguration of RAN 1 and NRM of operator B requests and controls reconfiguration of RAN 2. Following RANs' reconfiguration, the Terminals need to reconfigure correspondingly.

After reconfiguration, part of BSs of RAN 1 starts operation in the frequency band of RAN 2. The usage of the frequency bands of operators A and B is now balanced.

For operator A, this allows the use of the frequency band of operator B to improve radio resource usage and quality of service. Operator B can get some revenue for sharing its frequency band with operator A.

A spectrum renting example is shown in Figure A.3. Operator A operates RAN 1 network in its frequency band. This frequency band is currently overused. Operator B has frequency band for future RAN 2. Currently, only a small part of this frequency band is occasionally used for trial. Spectrum usage is unbalanced.

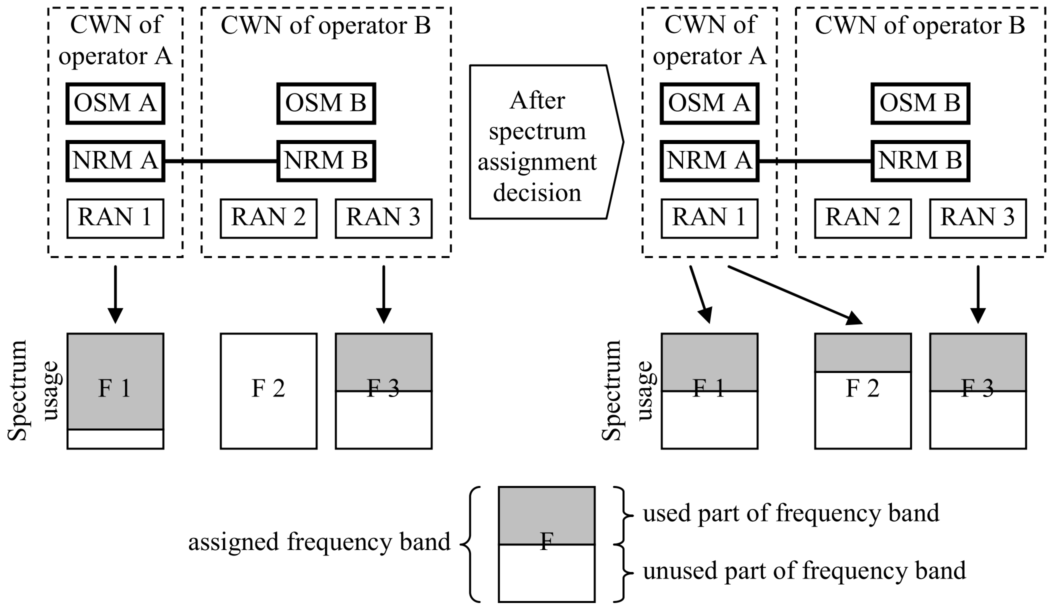


Figure A.3—Dynamic spectrum assignment: multiple operator scenario 1 (NRM is inside operator): spectrum renting example

OSMs and/or NRMs of operators A and B negotiate to rent part of RAN 2 frequency band of operator B to operator A. After negotiation, NRM of operator A requests and controls reconfiguration of RAN 1 of operator A. Following RANs' reconfiguration, the Terminals need to reconfigure correspondingly.

After reconfiguration, part of the BSs of RAN 1 of operator A starts operation in RAN 2 frequency band of operator B. The usage of frequency bands of operators A and B is now balanced.

For operator A, this allows the use of the frequency band of operator B to improve radio resource usage and quality of service. Operator B can get some revenue for renting its frequency band to operator A.

The multiple operator dynamic spectrum assignment scenario is enabled by NRM. NRM allows cross-operator balancing of spectrum usage. This increases quality of service of all involved operators and provides new mechanisms to receive additional revenue.

Multiple operator scenario 1 (NRM is inside operator) is realized as follows:

a) OSMs and/or NRMs of operators perform interactions between each other regarding dynamic spectrum assignment
b) OSM of each operator generates spectrum assignment policies and sends them to its NRM
c) NRM of each operator receives these spectrum assignment policies
d) NRMs obtain RAN context information
e) TRMs obtain terminal context information
f) NRMs and their TRMs exchange context information
g) Each NRM analyzes spectrum assignment policies and context information
h) Each NRM evaluates current spectrum assignment inside operator and makes new spectrum assignment decision

i) Each NRM informs OSM about its spectrum assignment decision

j) NRM of each operator requests and controls corresponding reconfiguration of RANs of this operator

k) Following the RANs' reconfiguration, Terminals reconfigure correspondingly

A.1.3 Multiple operator scenario 2 (NRM is outside operators)

Multiple operator scenario 2 assumes that several frequency bands are allocated to several operators and operators have some level of flexibility for renting or sharing these frequency bands. OSM evaluates efficiency of current spectrum assignment for each operator. However, OSMs of different operators cannot negotiate with each other and cross-operator dynamic spectrum sharing or renting is not possible. NRM introduced in this standard enables cross-operator optimization of spectrum usage by performing dynamic spectrum assignment.

A spectrum sharing example is shown in Figure A.4. Operator A and B operate RAN 1 and RAN 2 in their frequency bands. The frequency band assigned to operator A is overused. The frequency band assigned to operator B is underused. Spectrum usage is unbalanced.

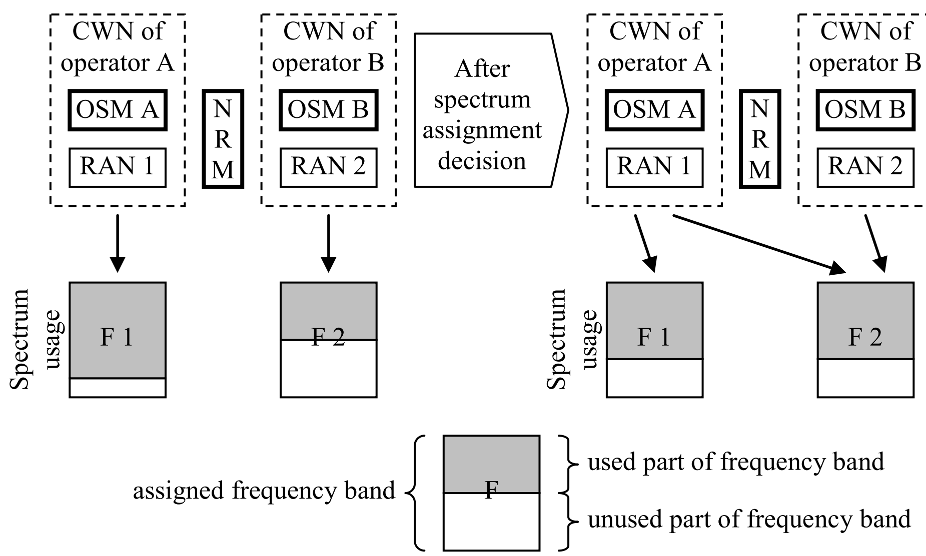


Figure A.4—Dynamic spectrum assignment: multiple operator scenario 2 (NRM is outside operators): spectrum sharing example

OSM of operator A sends spectrum assignment policy to NRM expressing need for additional spectrum resources. OSM of operator B sends spectrum assignment policy to NRM expressing possibility to share its frequency band.

Based on the analysis of spectrum assignment policies received from OSMs of operators A and B, as well as, on analysis of terminal and network context information NRM makes new spectrum assignment decision. NRM allows operator A to use the frequency band of operator B.

NRM informs OSMs of operators A and B about this spectrum assignment decision. Also, NRM requests and controls reconfiguration of RAN 1 of operator A. Following the RANs' reconfiguration, Terminals need to reconfigure correspondingly.

After reconfiguration, part of the BSs of RAN 1 of operator A starts operation in frequency band of operator B. The usage of frequency bands of operators A and B is now balanced.

For operator A, this allows the use of the frequency band of operator B to improve radio resource usage and quality of service. Operator B can get some revenue for sharing its frequency band with operator A.

A spectrum renting example is shown in Figure A.5. Operator A operates RAN 1 in its frequency band. This frequency band is currently overused. Operator B has a frequency band for future RAN 2. Currently, only a small part of this frequency band is occasionally used for trial. Spectrum usage is unbalanced.

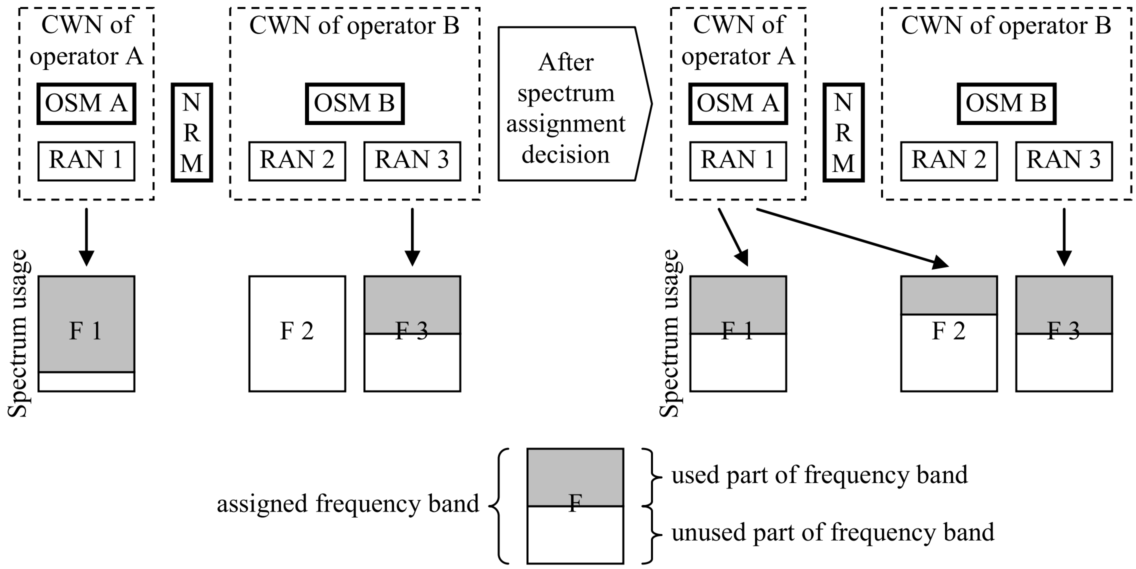


Figure A.5—Dynamic spectrum assignment: multiple operator scenario 2 (NRM is outside operators): spectrum renting example

The OSM of operator A sends spectrum assignment policy to NRM expressing the need for additional spectrum resources. The OSM of operator B sends spectrum assignment policy to NRM expressing the possibility to rent its RAN 2 frequency band.

Based on the analysis of the spectrum assignment policies received from the OSMs of operators A and B, as well as, on analysis of terminal and network context information, the NRM makes a new spectrum assignment decision. NRM decides to rent part of the RAN 2 frequency band of operator B to operator A.

The NRM informs the OSMs of operators A and B about this spectrum assignment decision. Also, the NRM requests and controls reconfiguration of RAN 1 of operator A. Following the RANs' reconfiguration, Terminals need to reconfigure correspondingly.

After reconfiguration, part of the BSs of RAN 1 of operator A starts operation in the RAN 2 frequency band of operator B. The usage of frequency bands of operators A and B is now balanced.

For operator A, this allows the use of the frequency band of operator B to improve radio resource usage and quality of service. Operator B can get some revenue for renting its frequency band to operator A.

The multiple operator dynamic spectrum assignment scenario is enabled by NRM. NRM allows cross-operator balancing of spectrum usage. This increases quality of service of all involved operators and provides new mechanisms to receive additional revenue.

Multiple operator scenario 2 (NRM is outside operators) is realized as follows:

a) OSMs of operators generate spectrum assignment policies and send them to NRM
b) NRM receives these spectrum assignment policies from the OSMs
c) NRM obtains RAN context information
d) TRMs obtain terminal context information
e) NRM and TRMs exchange context information
f) NRM analyzes spectrum assignment policies and context information
g) NRM evaluates current spectrum assignment inside multiple operators and makes new spectrum assignment decision
h) NRM informs OSMs about its spectrum assignment decision
i) NRM requests and controls corresponding reconfiguration of RANs of multiple operators
j) Following the RANs' reconfiguration, Terminals reconfigure correspondingly

A.2 Dynamic spectrum sharing

In the dynamic spectrum sharing use case, frequency bands assigned to the RANs are fixed. However, a particular frequency band can be shared by several RANs. In other words, the dynamic spectrum sharing use case describes how fixed frequency bands are shared and/or used dynamically.

Following the dynamic spectrum sharing decisions, corresponding RANs and Terminals are reconfigured.

One or several frequency bands are available for joint use by several RANs. These RANs and their Terminals dynamically access these frequency bands for improving spectrum usage and quality of service. Decisions on this dynamic spectrum sharing are jointly made by the NRM and the TRMs in a distributed manner. After the decisions have been made, NRM facilitates corresponding reconfiguration of RANs and TRMs facilitate corresponding reconfiguration of their Terminals.

An example of dynamic spectrum sharing is shown in Figure A.6. Operators A and B operate RAN 1 and RAN 2 in two frequency bands. These frequency bands are available for joint use by both operators according to regulatory rules. Currently, the frequency band of operator B is underused. Spectrum usage is unbalanced.

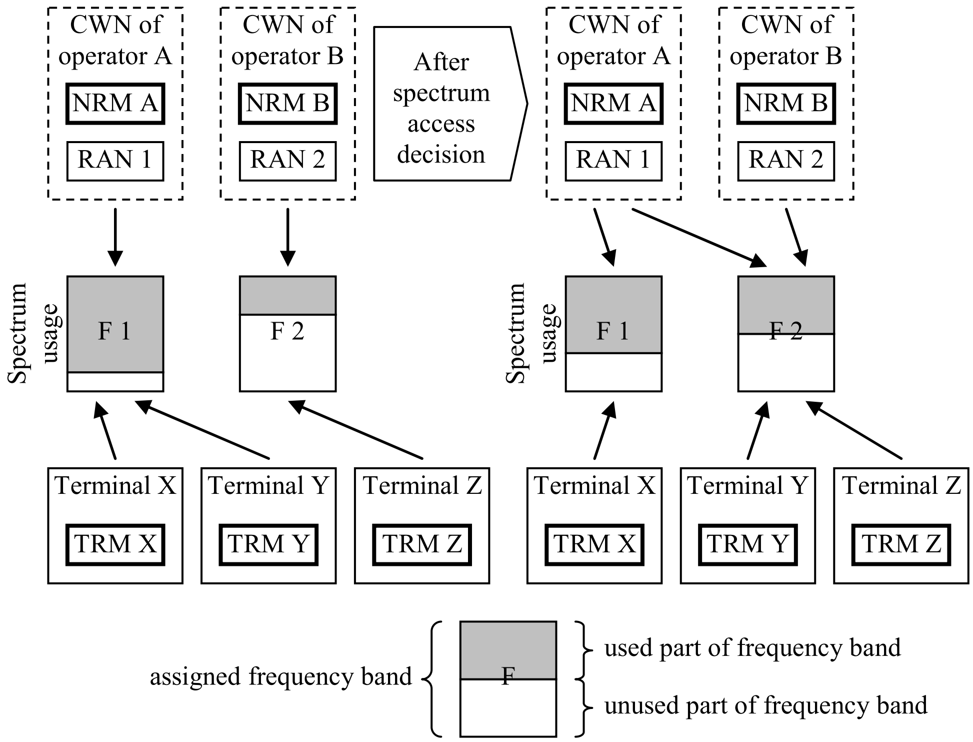


Figure A.6—Dynamic spectrum sharing

NRMs obtain RAN context information from the RANs of operators A and B. TRMs obtain terminal context information from their Terminals. NRMs and TRMs exchange context information with each other.

Based on the analysis of context information, NRMs detect that frequency band of operator A is overused while the frequency band of operator B is underused. As a result, NRMs make new dynamic spectrum sharing decision. It decides that part of the BSs of RAN 1 of operator A shall access the frequency band of operator B.

NRMs request and control corresponding reconfiguration of RAN 1 and RAN 2. Also, NRMs generate radio resource selection policies that guide Terminals to dynamically access the frequency band of operator B. NRMs send these radio resource selection policies to corresponding TRMs.

TRMs analyze received radio resource selection policies and available context information. Based on the analysis, some TRMs make the decision to access the frequency band of operator B. These TRMs request and control the corresponding reconfiguration of their Terminals.

After part of RAN 1 and part of its Terminals of operator A start operation in the frequency band of operator B, usage of the frequency bands of operators A and B is balanced.

Dynamic spectrum sharing is enabled by NRMs, TRMs, and collaboration between NRMs and TRMs. NRMs make spectrum access decisions for RANs, as well as, requests and controls corresponding reconfiguration of RANs. Also, NRMs generate radio resource selection policies and send them to TRMs. TRMs make final decision on spectrum access for Terminals, as well as, request and control corresponding reconfiguration of their Terminals. Finally, NRMs and TRMs obtain and exchange context information used for decision making. Dynamic spectrum sharing improves spectrum usage and increases quality of service.

Dynamic spectrum sharing use case is realized as follows:

a) NRMs obtain RAN context information
b) TRMs obtain terminal context information
c) NRMs and TRMs exchange context information
d) NRMs exchange context information
e) NRMs analyzes context information
f) NRMs make decisions on spectrum access to improve spectrum usage and quality of service
g) NRMs request and control corresponding reconfiguration of RANs
h) NRMs generate radio resource selection policies to guide TRMs in their spectrum access decisions and send them to their TRMs
i) TRMs analyze received radio resource selection policies and available context information
j) TRMs make decisions on spectrum access to improve spectrum usage and quality of service
k) TRMs request and control corresponding reconfiguration of their Terminals

A.3 Distributed radio resource usage optimization

Distributed radio resource usage optimization use case demonstrates how the IEEE 1900.4 system can be applied to legacy RANs in order to optimize radio resource usage and improve quality of service.

In the distributed radio resource usage optimization use case, frequency bands assigned to RANs are fixed. Also, reconfiguration of RANs is not involved in this use case. Decision making on reconfiguration of Terminals is performed in a distributed manner.

Distributed radio resource usage optimization use case considers Terminals with or without multi-homing capability.

An example of distributed radio resource usage optimization use case is shown in Figure A.7. Operators A and B operate RAN 1, RAN 2, and RAN 3 in three frequency bands. Terminal 1 with multi-homing capability can have simultaneous connections with these RANs. Currently, Terminal 1 is connected to RAN 1 of operator A and RAN 2 of operator B. Terminal 2 without multi-homing capability can have one active connection with any of these RANs. Currently, Terminal 2 is connected to RAN 2 of operator B. Currently, the RAN 2 frequency band of operator B is overused while the RAN 3 frequency band of operator B is underused. Spectrum usage is unbalanced.

NRMs of operators A and B analyze available context information and detect imbalance in spectrum usage. NRMs generate radio resource selection policies that recommend changing some connections from RAN 2 to RAN 3. These radio resource selection policies are sent from NRMs of operators A and B to TRMs of Terminals 1 and 2. Together with these radio resource selection policies, NRMs specify and send time intervals for reconfiguration of Terminals. TRMs of Terminals 1 and 2 analyze received radio resource selection policies and available context information. They detect imbalance in spectrum usage. The TRM of Terminal 1 makes the decision to change one of its two connections from RAN 2 to RAN 3. The TRM of Terminal 2 makes the decision to change its connection from RAN 2 to RAN 3.

The TRMs request and control the corresponding reconfiguration of their Terminals. After reconfiguration, Terminal 1 is connected to RAN 1 of operator A and RAN 3 of operator B, while Terminal 2 is connected to RAN 3 of operator B. The usage of RAN 2 and RAN 3 frequency bands is now balanced.

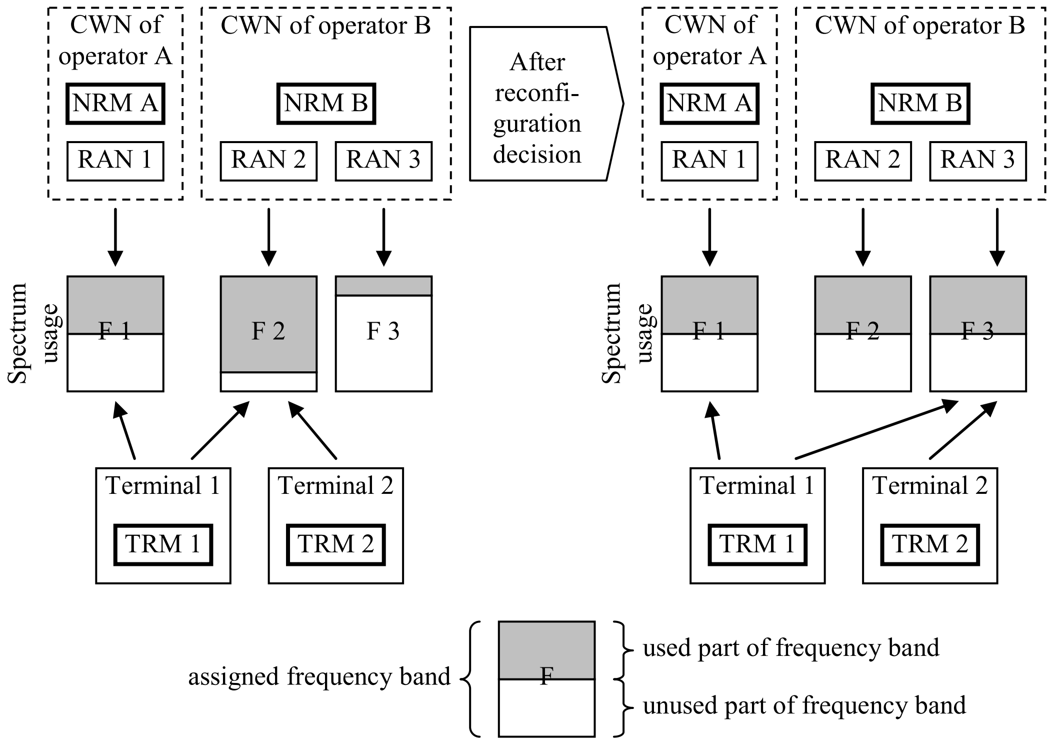


Figure A.7—Distributed radio resource usage optimization

Distributed radio resource usage optimization is enabled by NRMs, TRMs, and the collaboration between NRMs and TRMs. NRMs and TRMs obtain and exchange context information used for decision making. NRMs generate radio resource selection policies that guide TRMs in their decisions. TRMs make final decisions on reconfiguration of their Terminals, as well as, request and control corresponding reconfiguration of their Terminals. Distributed radio resource usage optimization improves spectrum usage and increases quality of service.

Distributed radio resource usage optimization use case is realized as follows:

a) NRMs obtain RAN context information
b) TRMs obtain terminal context information
c) NRMs and TRMs exchange context information
d) NRMs exchange context information
e) NRMs analyze context information
f) NRMs generate radio resource selection policies and time intervals for reconfiguration of Terminals and send them to their TRMs. These radio resource selection policies should correspond to specific groups of Terminals
g) TRMs analyze received radio resource selection policies and available context information
h) TRMs make decisions on their Terminals reconfiguration, within the specified reconfiguration time intervals to improve radio resource usage and quality of service
i) TRMs request and control corresponding reconfiguration of their Terminals within the provided time intervals

Annex B

(normative)

Class definitions for information model

B.1 Notational tools

The tables defining the classes use the following template (see Table B.1).

Table B.1—Template of table of class definition

<table>
<tr><td colspan="4">Class <Class name> [(abstract class)]</td></tr>
<tr><td colspan="4"><Description of the class></td></tr>
<tr><td>DERIVED FROM</td><td colspan="3"><List of super-classes></td></tr>
<tr><td colspan="4">ATTRIBUTES</td></tr>
<tr><td><Attribute name>
[<optional>]</td><td>Value type:
<Attribute value type></td><td>Possible access:
<Attribute access qualifier></td><td>Default value:
<Default value></td></tr>
<tr><td colspan="4"><Description of the attribute></td></tr>
<tr><td>CONTAINED IN</td><td colspan="3"><List of classes, whose instances may contain an instance of this class. If this class is an abstract class, that is, it is used for further refinement only and will never be instantiated, then this list is empty.></td></tr>
<tr><td>CONTAINS</td><td colspan="3"><List of classes, whose instances may be contained in an instance of this class. Constraints used are:
[*] – zero or more instances,
[+] – one or more instances,
[<n>] – exactly n instances,
[<m> – <n>] – not less than m and not more than n instances.></td></tr>
<tr><td>SUPPORTED EVENTS</td><td colspan="3"><List of event names that are detected by this class and lead potentially to a corresponding event report></td></tr>
</table>

A description of the template is provided within the following list:

- <Class name> is the name of the Class as it is appeared in the corresponding model. Additional information is also included in case the class in question has been specified as an abstract one.
- DERIVED FROM field identifies the super class of the class in case of sub-classing.
- ATTRIBUTES field describes the attributes that have been defined in the class. More specifically:
    - <Attribute name> identifies the name of an attribute, as it is included in the class definition.
    - <Attribute value type> holds the type of the attribute specified in ASN.1. Readers shall refer to the ASN.1 module for details (see Annex C).
    - <Attribute access qualifier> provides information about the level of accessibility of the attribute. This may include: 'Read,' 'Write,' 'Read-Write,' 'Add-Remove' (for list-type attributes), 'Read-Add-Remove,' and 'None' (for internal access only).
- CONTAINED IN field includes a list of classes whose instances may contain an instance of this class; containment is a strong aggregation relationship, that is, a contained instance is for its lifetime bound to its container object and it is contained only in this one container.

— CONTAINS field provides a list of classes whose instances may be contained in an instance of the class in question.

— SUPPORTED EVENTS field includes a list of event names that are detected by this class and lead potentially to a corresponding event report. Possible usage of these properties is explained in Annex D.

B.2 Common base class

UML class diagram for common base class without inheritance relations is shown in Figure B.1.

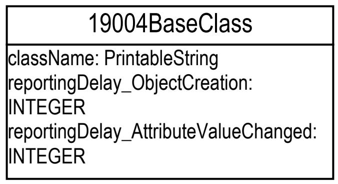


Figure B.1—UML class diagram for common base class

Table B.2 describes the 19004BaseClass class.

Table B.2—19004BaseClass class definition

Class **19004BaseClass** *(abstract class)*			
This class provides base interface for IEEE 1900.4 class definitions. Each class defined for IEEE 1900.4 shall be derived from this class if it can be instantiated more than once in the scope of the same immediate superior object.			
DERIVED FROM			
ATTRIBUTES			
className	*Value type:* PrintableString	*Possible access:* Read	*Default value:* *- not specified -*
This attribute allows to retrieve the name of the class an object belongs to.			
reportingDelay_Object Creation	*Value type:* INTEGER	*Possible access:* Read-Write	*Default value:* 0
Requested delay (in ms) between internal event detection and its (potential) event reporting. This is useful to report sustainable configuration changes only. By default, no delay is specified.			
reportingDelay_Attribute ValueChanged	*Value type:* INTEGER	*Possible access:* Read-Write	*Default value:* 0
Requested delay (in ms) between internal event detection and its (potential) event reporting. This is useful to report sustainable configuration changes only. By default, no delay is specified.			
CONTAINED IN			
CONTAINS			
SUPPORTED EVENTS	objectCreation, objectDeletion, attributeValueChanged		

B.3 Policy classes

UML class diagram for policy classes without inheritance relations is shown in Figure B.2.

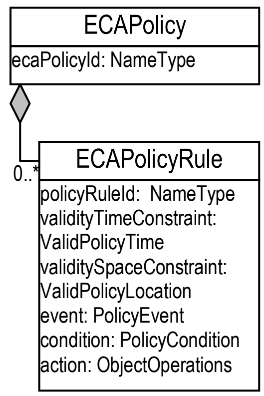


Figure B.2—UML class diagram for policy classes

Table B.3 describes the ECAPolicy class.

Table B.3—ECAPolicy class definition

Class **ECAPolicy**			
The instance of this class contains a set of policy rules governing the behavior of a decision making function. Within this model only ECA policy rules are defined.			
DERIVED FROM	**19004BaseClass**		
ATTRIBUTES			
ecaPolicyId	*Value type:* NameType	*Possible access:* Read	*Default value:* *- not specified -*
This attribute contains string or number assigned to uniquely identify the ECAPolicy object.			
CONTAINED IN			
CONTAINS	**ECAPolicyRule[*]**		
SUPPORTED EVENTS			

Table B.4 describes the ECAPolicyRule class.

Table B.4—ECAPolicyRule class definition

Class **ECAPolicyRule**			
This class describes an event condition action based policy rule.			
DERIVED FROM	**19004BaseClass**		
ATTRIBUTES			
policyRuleId	*Value type:* NameType	*Possible access:* Read	*Default value:* *- not specified -*
This attribute contains string or number assigned to uniquely identify the policy rule object.			
validityTimeConstraint	*Value type:* ValidPolicyTime	*Possible access:* Read-Write	*Default value:* NULL
If this time is reached the policy rule object has to be deleted at the Terminal. The type definition for this attribute is optional, that is, possibility not to define a time constraint is supported. Default: no time constraint			
validitySpaceConstraint	*Value type:* ValidPolicyLocation	*Possible access:* Read-Write	*Default value:* NULL
If the Terminal is outside this location area the policy rule has to be deleted at the Terminal. The type definition for this attribute is optional, that is, possibility not to define a space constraint is supported. Default: no space constraint			
event	*Value type:* PolicyEvent	*Possible access:* Read-Write	*Default value:* *- not specified -*
This attribute describes the event that triggers the evaluation of this policy rule for decision making.			
condition	*Value type:* PolicyCondition	*Possible access:* Read-Write	*Default value:* *- not specified -*
This attribute describes the condition that shall be true to apply the action of this policy rule.			
action	*Value type:* ObjectOperations	*Possible access:* Read-Write	*Default value:* *- not specified -*
This attribute describes the action to be performed in case the event has occurred and the condition holds.			
CONTAINED IN	**ECAPolicy**		
CONTAINS			
SUPPORTED EVENTS			

B.4 Terminal classes

UML class diagram for terminal classes without inheritance relations is shown in Figure B.3.

In addition to relations between terminal-related classes described in Clause 7, the following relations are defined:

- Each instance of **Application** class can be associated to zero or one instances of **Application** class.
- Each instance of **Application** class can be associated to one or several instances of **Link** class.

Table B.5 through Table B.27 describe each Terminal class in detail.

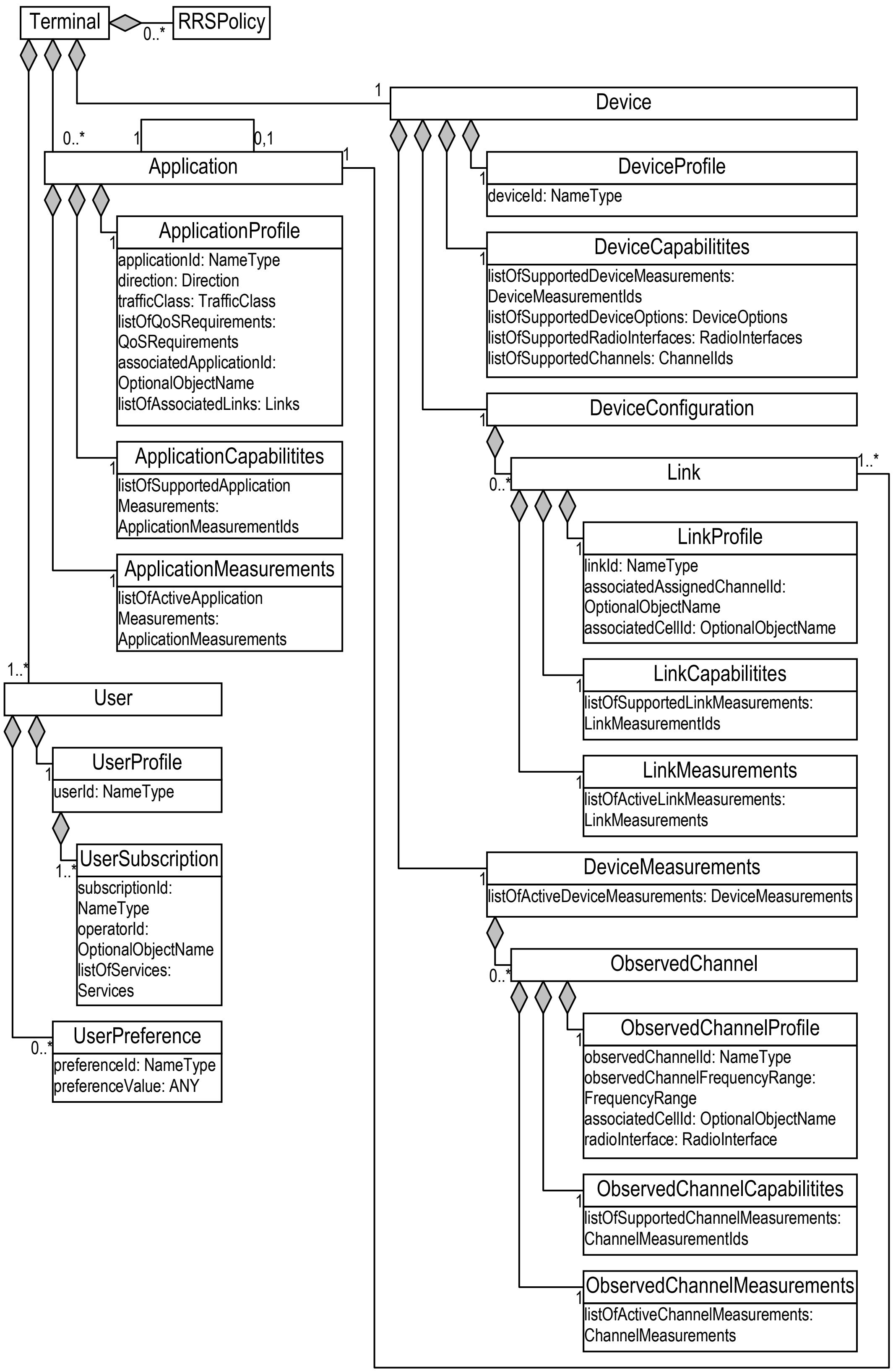


Figure B.3—UML class diagram for terminal classes

Table B.5—Terminal class definition

Class **Terminal**	
The instance of this class contains instances of all terminal-related classes using composition.	
DERIVED FROM	**19004BaseClass**
ATTRIBUTES	
CONTAINED IN	
CONTAINS	**User[+], Application[*], Device[1], RRSPolicy[*]**
SUPPORTED EVENTS	

Table B.6—User class definition

Class **User**	
This class describes information related to a user of the Terminal.	
DERIVED FROM	**19004BaseClass**
ATTRIBUTES	
CONTAINED IN	**Terminal**
CONTAINS	**UserProfile[1], UserPreference[*]**
SUPPORTED EVENTS	

Table B.7—UserProfile class definition

Class **UserProfile**			
This class contains general information about one user of the Terminal.			
DERIVED FROM			
ATTRIBUTES			
userId	*Value type:* NameType	*Possible access:* Read	*Default value:* *- not specified -*
This attribute contains string or number assigned to uniquely identify the user.			
CONTAINED IN	**User**		
CONTAINS	**UserSubscription[+]**		
SUPPORTED EVENTS			

Table B.8—UserSubscription class definition

Class **UserSubscription**			
This class contains information about one subscription of the user.			
DERIVED FROM	**19004BaseClass**		
ATTRIBUTES			
subscriptionId	*Value type:* NameType	*Possible access:* Read	*Default value:* *- not specified -*
This attribute contains string or number assigned to uniquely identify one subscription of the user.			
operatorId	*Value type:* OptionalObjectName	*Possible access:* Read	*Default value:* *- not specified -*
This optional attribute contains string or number assigned to uniquely identify the operator providing service within this subscription.			
listOfServices	*Value type:* Services	*Possible access:* Read	*Default value:* *- not specified -*
This attributes describes services provided within this subscription.			
CONTAINED IN	**UserProfile**		
CONTAINS			
SUPPORTED EVENTS			

Table B.9—UserPreference class definition

Class **UserPreference**			
This class describes in a formalized form one preference of the user, for example, preferred operator and radio interface, perceived audio/image/video quality, maximum cost, minimum data rate, etc.			
DERIVED FROM	**19004BaseClass**		
ATTRIBUTES			
preferenceId	*Value type:* NameType	*Possible access:* Read	*Default value: - not specified -*
This attribute contains string or number assigned to uniquely identify one preference of the user.			
preferenceValue	*Value type:* ANY	*Possible access:* Read	*Default value: - not specified -*
This attribute described in a formalized form one preference of the user.			
CONTAINED IN	**User**		
CONTAINS			
SUPPORTED EVENTS			

Table B.10—Application class definition

Class **Application**	
This class describes one currently active application.	
DERIVED FROM	**19004BaseClass**
ATTRIBUTES	
CONTAINED IN	**Terminal**
CONTAINS	**ApplicationProfile[1], ApplicationCapabilities[1], ApplicationMeasurements[1]**
SUPPORTED EVENTS	

Table B.11—ApplicationProfile class definition

Class **ApplicationProfile**			
This class contains general information about the application.			
DERIVED FROM			
ATTRIBUTES			
applicationId	*Value type:* NameType	*Possible access:* Read	*Default value: - not specified -*
This attribute contains string or number assigned to uniquely identify one application.			
direction	*Value type:* Direction	*Possible access:* Read	*Default value: - not specified -*
This attribute describes whether this application is downlink or uplink application.			
trafficClass	*Value type:* TrafficClass	*Possible access:* Read	*Default value: - not specified -*
This attributes describes traffic class of the application.			
listOfQoSRequirements	*Value type:* QoSRequirements	*Possible access:* Read	*Default value: - not specified -*
This attributes describes QoS requirements of the application.			
associatedApplicationId	*Value type:* OptionalObjectName	*Possible access:* Read	*Default value: - not specified -*
This attribute contains ID of associated application having other direction if any.			
listOfAssociatedLinks	*Value type:* Links	*Possible access:* Read	*Default value: - not specified -*
This attributes contains list of IDs of links used to transmit this application.			
CONTAINED IN	**Application**		
CONTAINS			
SUPPORTED EVENTS			

Table B.12—ApplicationCapabilities class definition

Class **ApplicationCapabilities**			
This class contains information about measurements (instantaneous measurement data and performance statistics derived from this data) supported by this application.			
DERIVED FROM			
ATTRIBUTES			
listOfSupportedApplication Measurements	*Value type:* ApplicationMeasurementIds	*Possible access:* Read	*Default value: - not specified -*
This attribute describes measurements supported by this application.			
CONTAINED IN	**Application**		
CONTAINS			
SUPPORTED EVENTS			

Table B.13—ApplicationMeasurements class definition

Class **ApplicationMeasurements**			
This class contains measurements (instantaneous measurement data and performance statistics derived from this data) performed by this application.			
DERIVED FROM			
ATTRIBUTES			
listOfActiveApplication Measurements	*Value type:* ApplicationMeasurements	*Possible access:* Read-Add-Remove	*Default value: - not specified -*
This attribute describes measurements that are currently performed by the application.			
CONTAINED IN	**Application**		
CONTAINS			
SUPPORTED EVENTS			

Table B.14—Device class definition

Class **Device**	
This class describes all radio interface related hardware and software of a Terminal, as well as, measurement information related to radio resources within the Terminal.	
DERIVED FROM	**19004BaseClass**
ATTRIBUTES	
CONTAINED IN	**Terminal**
CONTAINS	**DeviceProfile[1], DeviceCapabilities[1], DeviceConfiguration[1], DeviceMeasurements[1]**
SUPPORTED EVENTS	

Table B.15—DeviceProfile class definition

Class **DeviceProfile**			
This class contains general information about the Terminal.			
DERIVED FROM			
ATTRIBUTES			
deviceId	*Value type:* NameType	*Possible access:* Read	*Default value: - not specified -*
This attribute contains string or number assigned to uniquely identify the Terminal.			
CONTAINED IN	**Device**		
CONTAINS			
SUPPORTED EVENTS			

Table B.16—DeviceCapabilities class definition

Class **DeviceCapabilities**			
This class contains information about the Terminal capabilities including both transmission and measurement capabilities.			
DERIVED FROM			
ATTRIBUTES			
listOfSupportedDevice Measurements	*Value type:* DeviceMeasurementIds	*Possible access:* Read	*Default value: - not specified -*
This attribute describes measurements supported by the Terminal that are not related to any link or observed channel.			
listOfSupportedDeviceOptions	*Value type:* DeviceOptions	*Possible access:* Read	*Default value: - not specified -*
This attribute describes options supported by the Terminal.			
listOfSupportedRadioInterface s	*Value type:* RadioInterfaces	*Possible access:* Read	*Default value: - not specified -*
This attributes describes radio interfaces supported by this Terminal.			
listOfSupportedChannels	*Value type:* ChannelIds	*Possible access:* Read	*Default value: - not specified -*
This attributes describes frequency channels supported by the Terminal.			
CONTAINED IN	**Device**		
CONTAINS			
SUPPORTED EVENTS			

Table B.17—DeviceConfiguration class definition

Class **DeviceConfiguration**	
This class contains information about the current configuration of the Terminal.	
DERIVED FROM	
ATTRIBUTES	
CONTAINED IN	**Device**
CONTAINS	**Link[*]**
SUPPORTED EVENTS	

Table B.18—Link class definition

Class **Link**	
This class contains information about one active connection between the Terminal and RANs.	
DERIVED FROM	**19004BaseClass**
ATTRIBUTES	
CONTAINED IN	**DeviceConfiguration**
CONTAINS	**LinkProfile[1], LinkCapabilities[1], LinkMeasurements[1]**
SUPPORTED EVENTS	

Table B.19—LinkProfile class definition

Class **LinkProfile**			
This class contains general information about this active connection.			
DERIVED FROM			
ATTRIBUTES			
linkId	*Value type:* NameType	*Possible access:* Read	*Default value:* *- not specified -*
This attribute contains string or number assigned to uniquely identify this link.			
associatedAssignedChannelId	*Value type:* OptionalObjectName	*Possible access:* Read	*Default value:* *- not specified -*
This attribute contains ID of frequency channel used by this link.			
associatedCellId	*Value type:* OptionalObjectName	*Possible access:* Read	*Default value:* *- not specified -*
This attribute contains ID of cell used by this link.			
CONTAINED IN	**Link**		
CONTAINS			
SUPPORTED EVENTS			

Table B.20—LinkCapabilities class definition

Class **LinkCapabilities**			
This class contains information about measurements (instantaneous measurement data and performance statistics derived from this data) supported on this active connection.			
DERIVED FROM			
ATTRIBUTES			
listOfSupportedLink Measurements	*Value type:* LinkMeasurementIds	*Possible access:* Read	*Default value:* *- not specified -*
This attribute describes measurements supported on this links.			
CONTAINED IN	**Link**		
CONTAINS			
SUPPORTED EVENTS			

Table B.21—LinkMeasurements class definition

Class **LinkMeasurements**			
This class contains current measurements (instantaneous measurement data and performance statistics derived from this data) related to this active connection.			
DERIVED FROM			
ATTRIBUTES			
listOfActiveLink Measurements	*Value type:* LinkMeasurements	*Possible access:* Read-Add-Remove	*Default value:* *- not specified -*
This attribute describes measurements that are currently performed on this link.			
CONTAINED IN	**Link**		
CONTAINS			
SUPPORTED EVENTS			

Table B.22—DeviceMeasurements class definition

Class **DeviceMeasurements**			
This class contains current measurements (instantaneous measurement data and performance statistics derived from this data) related to the Terminal.			
DERIVED FROM			
ATTRIBUTES			
listOfActiveDevice Measurements	*Value type:* DeviceMeasurements	*Possible access:* Read-Add-Remove	*Default value:* *- not specified -*
This attribute describes measurements that are currently performed by the Terminal and are not related to any link or observed channel.			
CONTAINED IN	**Device**		
CONTAINS	**ObservedChannel[*]**		
SUPPORTED EVENTS			

Table B.23—ObservedChannel class definition

Class **ObservedChannel**	
This class describes one frequency channel that does not have active connection with the Terminal, but is observed by this Terminal.	
DERIVED FROM	**19004BaseClass**
ATTRIBUTES	
CONTAINED IN	**DeviceMeasurements**
CONTAINS	**ObservedChannelProfile[1], ObservedChannelCapabilities[1], ObservedChannelMeasurements[1]**
SUPPORTED EVENTS	

Table B.24—ObservedChannelProfile class definition

Class **ObservedChannelProfile**			
This class contains general information about this frequency channel.			
DERIVED FROM			
ATTRIBUTES			
observedChannelId	*Value type:* NameType	*Possible access:* Read	*Default value:* *- not specified -*
This attribute contains string or number assigned to uniquely identify this frequency channel.			
observedChannelFrequency Range	*Value type:* FrequencyRange	*Possible access:* Read	*Default value:* *- not specified -*
This attribute describes frequency range used by this channel.			
associatedCellId	*Value type:* OptionalObjectName	*Possible access:* Read	*Default value:* *- not specified -*
This attributes contains ID of cell using this frequency channel if any.			
radioInterface	*Value type:* RadioInterface	*Possible access:* Read	*Default value:* *- not specified -*
This attributes describes radio interface used in this frequency channel if any.			
CONTAINED IN	**ObservedChannel**		
CONTAINS			
SUPPORTED EVENTS			

Table B.25—ObservedChannelCapabilities class definition

Class **ObservedChannelCapabilities**			
This class contains information about measurements (instantaneous measurement data and performance statistics derived from this data) supported on this frequency channel.			
DERIVED FROM			
ATTRIBUTES			
listOfSupportedChannel Measurements	*Value type:* ChannelMeasurementIds	*Possible access:* Read	*Default value:* *- not specified -*
This attribute describes measurements supported on this frequency channel.			
CONTAINED IN	**ObservedChannel**		
CONTAINS			
SUPPORTED EVENTS			

Table B.26—ObservedChannelMeasurements class definition

Class **ObservedChannelMeasurements**			
This class contains current measurements (instantaneous measurement data and performance statistics derived from this data) related to this frequency channel.			
DERIVED FROM			
ATTRIBUTES			
listOfActiveChannel Measurements	*Value type:* ChannelMeasurements	*Possible access:* Read-Add-Remove	*Default value:* *- not specified -*
This attribute describes measurements that are currently performed on this frequency channel.			
CONTAINED IN	**ObservedChannel**		
CONTAINS			
SUPPORTED EVENTS			

Table B.27—RRSPolicy class definition

Class **RRSPolicy**	
This class describes one radio resource selection policy related to this Terminal.	
DERIVED FROM	**ECAPolicy**
ATTRIBUTES	
CONTAINED IN	**Terminal**
CONTAINS	
SUPPORTED EVENTS	

B.5 CWN classes

UML class diagram for CWN classes without inheritance relations is shown in Figure B.4.

In addition to relations between the CWN-related classes described in Clause 7, the following relations are defined:

— Each instance of **Base Station** class can be associated to one or several instances of **Cell** class.

— Each instance of **Base Station** class can be associated to one or several instances of **Assigned Channel** class.

— Each instance of **Cell** class can be associated to one or several instances of **Assigned Channel** class.

Table B.28 through Table B.48 describe each CWN class in detail.

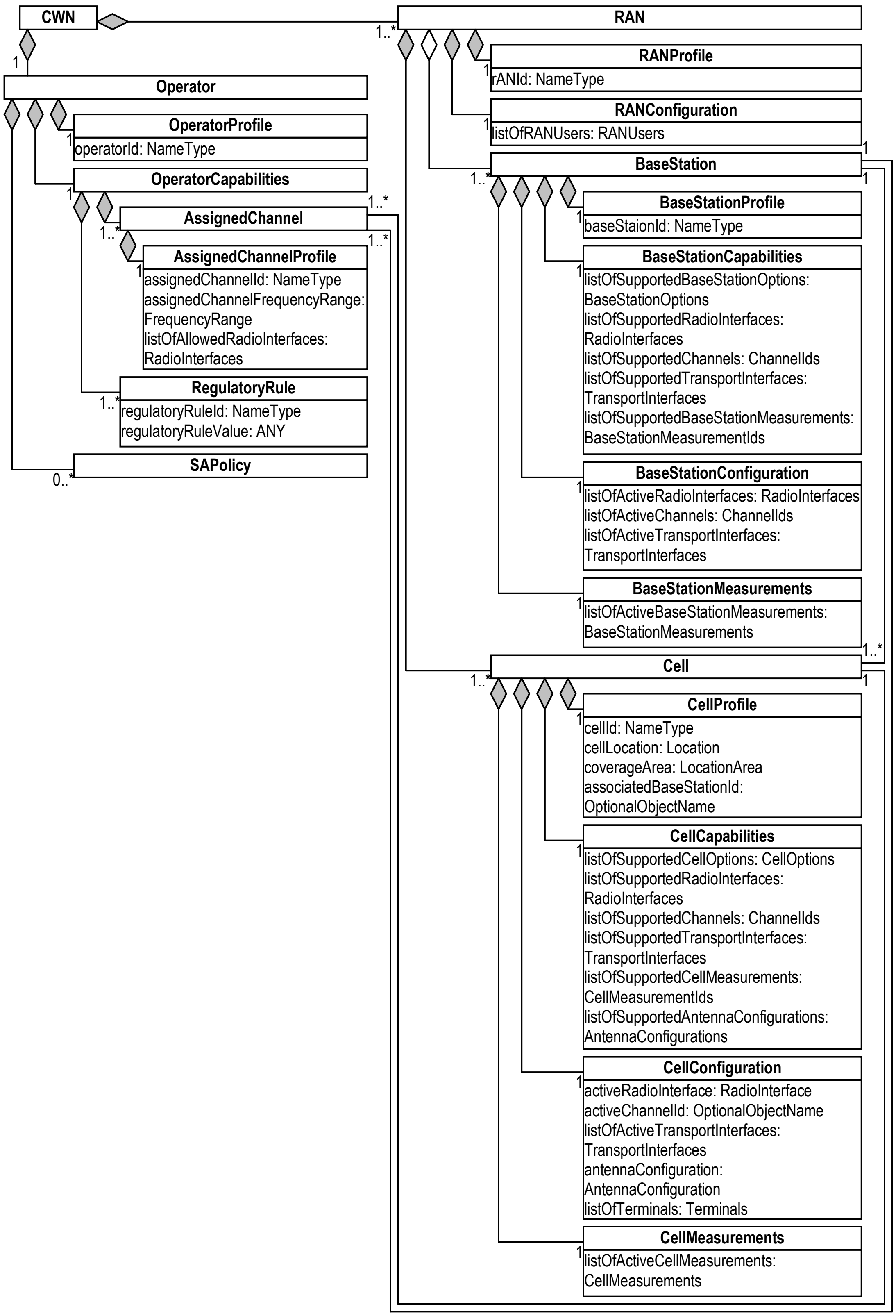


Figure B.4—UML class diagram for CWN classes

Table B.28—CWN class definition

Class **CWN**	
The instance of this class contains instances of all CWN-related classes using composition.	
DERIVED FROM	**19004BaseClass**
ATTRIBUTES	
CONTAINED IN	
CONTAINS	**Operator[1], RAN[+]**
SUPPORTED EVENTS	

Table B.29—Operator class definition

Class **Operator**	
This class describes operator of this CWN.	
DERIVED FROM	**19004BaseClass**
ATTRIBUTES	
CONTAINED IN	**CWN**
CONTAINS	**OperatorProfile[1], OperatorCapabilities[1], SAPolicy[*]**
SUPPORTED EVENTS	

Table B.30—OperatorProfile class definition

Class **OperatorProfile**			
This class contains general information about the operator.			
DERIVED FROM			
ATTRIBUTES			
operatorId	*Value type:* NameType	*Possible access:* Read	*Default value:* *- not specified -*
This attribute contains string or number assigned to uniquely identify this operator.			
CONTAINED IN	**Operator**		
CONTAINS			
SUPPORTED EVENTS			

Table B.31—OperatorCapabilities class definition

Class **OperatorCapabilities**	
This class describes operator capabilities.	
DERIVED FROM	
ATTRIBUTES	
CONTAINED IN	**Operator**
CONTAINS	**AssignedChannel[+], RegulatoryRule[+]**
SUPPORTED EVENTS	

Table B.32—AssignedChannel class definition

Class **AssignedChannel**			
This class describes one frequency channel assigned to this operator.			
DERIVED FROM	**19004BaseClass**		
ATTRIBUTES			
CONTAINED IN	**OperatorCapabilities**		
CONTAINS	**AssignedChannelProfile[1]**		
SUPPORTED EVENTS			

Table B.33—AssignedChannelProfile class definition

Class **AssignedChannelProfile**			
This class contains general information about this frequency channel.			
DERIVED FROM			
ATTRIBUTES			
assignedChannelId	*Value type:* NameType	*Possible access:* Read	*Default value:* *- not specified -*
This attribute contains string or number assigned to uniquely identify this frequency channel.			
assignedChannelFrequency Range	*Value type:* FrequencyRange	*Possible access:* Read	*Default value:* *- not specified -*
This attribute describes frequency range used by this channel.			
listOfAllowedRadioInterfaces	*Value type:* RadioInterfaces	*Possible access:* Read	*Default value:* *- not specified -*
This attributes contains list of IDs of radio interfaces allowed to be used on this frequency channel.			
CONTAINED IN	**AssignedChannel**		
CONTAINS			
SUPPORTED EVENTS			

Table B.34—RegulatoryRule class definition

Class **RegulatoryRule**			
This class describes in a formalized form one regulatory rule to be applied to one or several assigned channels.			
DERIVED FROM	**19004BaseClass**		
ATTRIBUTES			
regulatoryRuleId	*Value type:* NameType	*Possible access:* Read	*Default value:* *- not specified -*
This attribute contains string or number assigned to uniquely identify this regulatory rule			
regulatoryRuleValue	*Value type:* ANY	*Possible access:* Read	*Default value:* *- not specified -*
This attribute describes this regulatory rule.			
CONTAINED IN	**OperatorCapabilities**		
CONTAINS			
SUPPORTED EVENTS			

Table B.35—SAPolicy class definition

Class **SAPolicy**	
This class describes one spectrum assignment policy specified by the operator.	
DERIVED FROM	**ECAPolicy**
ATTRIBUTES	
CONTAINED IN	**Operator**
CONTAINS	
SUPPORTED EVENTS	

Table B.36—RAN class definition

Class **RAN**	
This class describes one RAN of this CWN.	
DERIVED FROM	**19004BaseClass**
ATTRIBUTES	
CONTAINED IN	**CWN**
CONTAINS	**RANProfile[1], RANConfiguration[1], BaseStation[+], Cell[+]**
SUPPORTED EVENTS	

Table B.37—RANProfile class definition

Class **RANProfile**			
This class contains general information about this RAN.			
DERIVED FROM			
ATTRIBUTES			
rANId	*Value type:* NameType	*Possible access:* Read	*Default value:* *- not specified -*
This attribute contains string or number assigned to uniquely identify this RAN.			
CONTAINED IN	**RAN**		
CONTAINS			
SUPPORTED EVENTS			

Table B.38—RANConfiguration class definition

Class **RANConfiguration**			
This class describes current configuration of this RAN.			
DERIVED FROM			
ATTRIBUTES			
listOfRANUsers	*Value type:* RANUsers	*Possible access:* Read	*Default value:* *- not specified -*
This attribute contains list of users of this RAN.			
CONTAINED IN	**RAN**		
CONTAINS			
SUPPORTED EVENTS			

Table B.39—BaseStation class definition

Class **BaseStation**	
This class describes one base station of the RAN.	
DERIVED FROM	**19004BaseClass**
ATTRIBUTES	
CONTAINED IN	**RAN**
CONTAINS	**BaseStationProfile[1], BaseStationCapabilities[1], BaseStationConfiguration[1], BaseStationMeasurements[1]**
SUPPORTED EVENTS	

Table B.40—BaseStationProfile class definition

Class **BaseStationProfile**			
This class contains general information about this base station.			
DERIVED FROM			
ATTRIBUTES			
baseStationId	*Value type:* NameType	*Possible access:* Read	*Default value: - not specified -*
This attribute contains string or number assigned to uniquely identify this base station.			
CONTAINED IN	**BaseStation**		
CONTAINS			
SUPPORTED EVENTS			

Table B.41—BaseStationCapabilities class definition

Class **BaseStationCapabilities**			
This class contains information about base station capabilities including both transmission and measurement capabilities.			
DERIVED FROM			
ATTRIBUTES			
listOfSupportedBaseStation Options	*Value type:* BaseStationOptions	*Possible access:* Read	*Default value: - not specified -*
This attribute describes options supported by this base station.			
listOfSupportedRadio Interfaces	*Value type:* RadioInterfaces	*Possible access:* Read	*Default value: - not specified -*
This attribute contains list of radio interfaces supported by this base station.			
listOfSupportedChannels	*Value type:* ChannelIds	*Possible access:* Read	*Default value: - not specified -*
This attributes contains list of IDs of frequency channels supported by this base station			
listOfSupportedTransport Interfaces	*Value type:* TransportInterfaces	*Possible access:* Read	*Default value: - not specified -*
This attributes describes transport interfaces supported by this base station.			
listOfSupportedBaseStation Measurements	*Value type:* BaseStationMeasurementIds	*Possible access:* Read	*Default value: - not specified -*
This attributes describes measurements supported by this base station and not related to any cell.			
CONTAINED IN	**BaseStation**		
CONTAINS			
SUPPORTED EVENTS			

Table B.42—BaseStationConfiguration class definition

Class **BaseStationConfiguration**			
This class contains information about the current configuration of the base station.			
DERIVED FROM			
ATTRIBUTES			
listOfActiveRadioInterfaces	*Value type:* RadioInterfaces	*Possible access:* Read	*Default value:* *- not specified -*
This attribute contains list of radio interfaces that are currently used by this base station.			
listOfActiveChannels	*Value type:* ChannelIds	*Possible access:* Read	*Default value:* *- not specified -*
This attributes contains list of IDs of frequency channels that are currently used by this base station.			
listOfActiveTransportInterfaces	*Value type:* TransportInterfaces	*Possible access:* Read	*Default value:* *- not specified -*
This attributes describes transport interfaces that are currently used by this base station.			
CONTAINED IN	**BaseStation**		
CONTAINS			
SUPPORTED EVENTS			

Table B.43—BaseStationMeasurements class definition

Class **BaseStationMeasurements**			
This class contains current measurements (instantaneous measurement data and performance statistics derived from this data) performed by this base station.			
DERIVED FROM			
ATTRIBUTES			
listOfActiveBaseStation Measurements	*Value type:* BaseStationMeasurements	*Possible access:* Read-Add-Remove	*Default value:* *- not specified -*
This attributes describes measurements that are currently performed by this base station and not related to any cell.			
CONTAINED IN	**BaseStation**		
CONTAINS			
SUPPORTED EVENTS			

Table B.44—Cell class definition

Class **Cell**	
This class describes one cell of the base station.	
DERIVED FROM	**19004BaseClass**
ATTRIBUTES	
CONTAINED IN	**RAN**
CONTAINS	**CellProfile[1], CellCapabilities[1], CellConfiguration[1], CellMeasurements[1]**
SUPPORTED EVENTS	

Table B.45—CellProfile class definition

Class **CellProfile**			
This class contains general information about this cell.			
DERIVED FROM			
ATTRIBUTES			
cellId	*Value type:* NameType	*Possible access:* Read	*Default value: - not specified -*
This attribute contains string or number assigned to uniquely identify this cell.			
cellLocation	*Value type:* Location	*Possible access:* Read	*Default value: - not specified -*
This attribute describes location of the cell.			
coverageArea	*Value type:* LocationArea	*Possible access:*	*Default value: - not specified -*
This attributes describes coverage area of the cell			
associatedBaseStationId	*Value type:* OptionalObjectName	*Possible access:* Read	*Default value: - not specified -*
This attributes contains ID of the base station to which this cell belongs.			
CONTAINED IN	**Cell**		
CONTAINS			
SUPPORTED EVENTS			

Table B.46—CellCapabilities class definition

Class **CellCapabilities**			
This class contains information about cell capabilities.			
DERIVED FROM			
ATTRIBUTES			
listOfSupportedCellOptions	*Value type:* CellOptions	*Possible access:* Read	*Default value: - not specified -*
This attribute describes options supported by this cell.			
listOfSupportedRadio Interfaces	*Value type:* RadioInterfaces	*Possible access:* Read	*Default value: - not specified -*
This attribute contains list of radio interfaces supported by this cell.			
listOfSupportedChannels	*Value type:* ChannelIds	*Possible access:* Read	*Default value: - not specified -*
This attributes contains list of IDs of frequency channels supported by this cell.			
listOfSupportedTransport Interfaces	*Value type:* TransportInterfaces	*Possible access:* Read	*Default value: - not specified -*
This attributes describes transport interfaces supported by this cell.			
listOfSupportedCell Measurements	*Value type:* CellMeasurementIds	*Possible access:* Read	*Default value: - not specified -*
This attributes describes measurements supported by this cell.			
listOfSupportedAntenna Configurations	*Value type:* AntennaConfigurations	*Possible access:* Read	*Default value: - not specified -*
This attributes describes antenna configurations supported by this cell			
CONTAINED IN	**Cell**		
CONTAINS			
SUPPORTED EVENTS			

Table B.47—CellConfiguration class definition

Class **CellConfiguration**			
This class contains information about the current configuration of the cell.			
DERIVED FROM			
ATTRIBUTES			
activeRadioInterface	*Value type:* RadioInterface	*Possible access:* Read	*Default value:* *- not specified -*
This attribute describes radio interface that is currently used by this cell.			
activeChannelId	*Value type:* OptionalObjectName	*Possible access:* Read	*Default value:* *- not specified -*
This attributes contains ID of frequency channel that is currently used by this cell.			
listOfActiveTransportInterfaces	*Value type:* TransportIntefaces	*Possible access:* Read	*Default value:* *- not specified -*
This attributes describes transport interfaces that are currently used by this cell.			
antennaConfiguration	*Value type:* AntennaConfiguration	*Possible access:* Read	*Default value:* *- not specified -*
This attributes describes current antenna configuration of this cell.			
listOfTerminals	*Value type:* Terminals	*Possible access:* Read	*Default value:* *- not specified -*
This attributes contains list of Terminals connected to this cell.			
CONTAINED IN	**Cell**		
CONTAINS			
SUPPORTED EVENTS			

Table B.48—CellMeasurements class definition

Class **CellMeasurements**			
This class contains current measurements (instantaneous measurement data and performance statistics derived from this data) related to this cell.			
DERIVED FROM			
ATTRIBUTES			
listOfCellMeasurements	*Value type:* CellMeasurements	*Possible access:* Read-Add- Remove	*Default value:* *- not specified -*
This attributes describes measurements that are currently performed by this cell.			
CONTAINED IN	**Cell**		
CONTAINS			
SUPPORTED EVENTS			

B.6 Relations between terminal and CWN classes

UML class diagram defining relations between terminal and CWN classes is shown in Figure B.5. This figure shows only a part of terminal related and CWN-related classes.

The following relations are defined:

- Each instance of **User** class can be associated to one or several instances of **Operator** class.
- Each instance of **RAN** class can be associated to zero or several instances of **User** class.
- Each instance of **Cell** class can be associated to zero or several instances of **Terminal** class.

- Each instance of **Observed Channel** class can be associated to zero or several instances of **Cell** class.
- Each instance of **Cell** class can be associated to zero or several instances of **Link** class.
- Each instance of **Assigned Channel** class can be associated to zero or several instances of **Link** class.

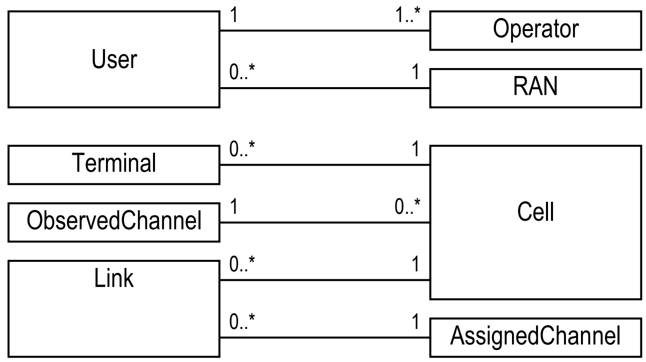


Figure B.5—UML class diagram defining relations between terminal and CWN classes

Annex C

(normative)

Data type definitions for information model

C.1 Function definitions

The managed system is modeled as a tree of objects that are instances of the classes defined within the information model. A number of functions are assumed to be available at the managed system side that work on this object tree and fulfill the required (by the managing system) actions or provide the requested information (to the managing system). These functions are described below in order to define the behavior of a managed system with respect to policy data (and their types) it has received from a managing system.

The following functions are assumed to be available locally (not necessarily exposed to an outside interface) at the managed system.

<u>Select a set of objects:</u>

```
ObjectSet (baseObject, level, filter)
```

<u>Returns:</u> List of object names

<u>Parameters:</u>

baseObject:	‘root’ or name of an object
level:	‘self’ | ‘directContained’ | ‘contained’ | ‘allSubtree’
filter:	logical expression combined from simple attribute value expressions

<u>Comments:</u>
baseObject and level together define the scope of objects (pre-selection step): ‘self’ means just the baseObject, ‘directContained’ means all objects that are directly contained in the baseObject, ‘contained’ also those objects that are indirectly contained (i.e., also contained in contained etc.) objects and, finally ‘allSubtree’ includes on top of ‘contained’ also the baseObject itself.
Simple attribute value expressions are defined as

```
<attributeName> ‘<’|’=’|’>’|’!=’|’<=’|’>=’
                                  <attributeName>|<constantValue>
```

If a simple attribute value expression is evaluated for an object found in the pre-selection step and the name of the attribute(s) is not contained in the definition of the object’s class, the expression is treated as false. All objects of the pre-selection step that pass the filter form the resulting object set.

Count objects of an object set:

```
ObjectCount (baseObject, level, filter)
```

Returns: Number of objects contained in ObjectSet (baseObject, level, filter).

Parameters: See function ObjectSet

Comments: None

Count events occurring in an object set:

```
EventCount (eventType, timeInterval, baseObject, level,
                                              filter)
```

Returns:

Number of events of type “eventType” occurring in the set of objects returned by “ObjectSet(baseObject, level, filter).” Only those events are counted that occurred during the last “timeInterval” seconds.

Parameters:

eventType: Type of events to be counted
timeInterval: Event count is the number of events occurring during the last timeInterval milliseconds
See function ObjectSet for the other parameters

Comments: None

The following function is needed for describing policy actions.

Ensure that attribute values are in a given value set or range:

```
Ensure (baseObject, level, filter, valueAssertion,
       timeslot)
```

Returns: Boolean value indicating success of the operation

Parameters:

valueAssertion: Type (`set` or `avoid`), attributeName and set(range) of values
timeslot: all necessary attribute settings are done in <timeslot> ms after this function has been called
See function ObjectSet for the parameters baseObject, level, and filter

Comments:

The function considers each object from ObjectSet (baseObject, level, filter) and applies to it the value assertion. The managed system has to ensure that the value of this attribute (providing this attribute is defined for the object) is set (`set`) or not set (`avoid`) to one of those values.

C.2 ASN.1 type definitions

The following ASN.1 (see ISO/IEC 8824 for ASN.1 specification) module contains all necessary abstract data definitions used in the attribute definitions in Annex B.

```
1900-4-Type-Definitions DEFINITIONS ::= BEGIN

            -----------------------------------------------------------
            -----------------------------------------------------------
            -- START Common Data Types

            -----------------------------------------------------------
            -- START Name Related Data Types

            NameType    ::=    CHOICE    {
                number    INTEGER,
                string    PrintableString
            }

            ObjectName    ::=    SEQUENCE OF NameType


            OptionalObjectName    ::=    CHOICE    {
                id      ObjectName,
                void    NULL
            }

            -- END Name Related Data Types
            -----------------------------------------------------------

            -----------------------------------------------------------
            -- START Radio Interface Related Data Types

            RadioInterfaceId    ::=    ENUMERATED    {
                umts, hsdpa, wimax, lte, wifi, gsm, ...
            }

            RadioInterface    ::= CHOICE    {
                id      RadioInterfaceId
                void    NULL
            }

            RadioInterfaces    ::=    SEQUENCE OF    RadioInterfaceId

            -- END Radio Interface Related Data Types
            -----------------------------------------------------------

            -----------------------------------------------------------
            -- START Channel Related Data Types

            ChannelIds    ::=    SEQUENCE OF    OptionalObjectName

            FrequencyRange    ::=    SEQUENCE    {
                centralFrequency    REAL,
                frequencyBand       REAL
            }

            -- END Channel Related Data Types
            -----------------------------------------------------------

            -----------------------------------------------------------
            -- START Location Related Data Types

            Location    ::= SEQUENCE    {
                latitude     REAL,
                longitude    REAL,
                height       REAL OPTIONAL
            }
```

```
LocationArea    ::=    SEQUENCE (SIZE(3 .. MAX)) OF SEQUENCE    {
    latitude     REAL,
    longitude    REAL
}

-- END Common Data Types
-----------------------------------------------------------
-----------------------------------------------------------


-----------------------------------------------------------
-----------------------------------------------------------
-- START Policy Related Data Types

-----------------------------------------------------------
-- START Policy Event Related Data Types

PolicyEvent    ::=    SEQUENCE    {
    eventType    ENUMERATED    {
        objectCreation, objectDeletion, stateChanged, attributeChanged,
        immediatelyOnce, scheduledTimer, ...
    }
    -- Additional data qualifying the event.
    -- Type is different and depends on eventType.
    eventQualifier    ANY
}

-- END Policy Event Related Data Types
-----------------------------------------------------------

-----------------------------------------------------------
-- START Policy Condition Related Data Types

ValidPolicyTime ::= CHOICE {
    validUntil     GeneralizedTime,
    unspecified    NULL
}

ValidPolicyLocation ::= CHOICE {
    validLocationArea    LocationArea,
    unspecified          NULL
}

LogicalOperator    ::=    ENUMERATED    {
    equal, non-equal, less, less-or-equal, greater, greater-or-equal
}

AttributeValueAssertion    ::=    SEQUENCE    {
    object            OptionalObjectName OPTIONAL,
    attributeName     PrintableString,
    fieldName         PrintableString OPTIONAL,
    attributeValue    ANY
}

ValueRange    ::=    CHOICE    {
    intRange    SEQUENCE{
        low     INTEGER,
        high    INTEGER
    }
    floatRange    SEQUENCE{
        low     REAL,
        high    REAL
    }
    stringRange    SEQUENCE{
        low     PrintableString,
        high    PrintableString
    }
}
```

```
ElementaryLogicalExpression    ::=    SEQUENCE    {
    attributeName    PrintableString,
    fieldname        PrintableString OPTIONAL,
    operation    CHOICE    {
        valueComparison    SEQUENCE    {
            operator    LogicalOperator,
            comparedWith    CHOICE    {
                definedValue           ANY,
                otherAttributeValue    AttributeValueAssertion
            },
            isContainedIn    SET OF CHOICE    {
                definedValue           ANY,
                definedRange           ValueRange,
                otherAttributeValue    AttributeValueAssertion
            }
        }
    }
}

Filter    ::=    CHOICE    {
    elementary       ElementaryLogicalExpression,
    andExpression    SEQUENCE    {
        operand1    Filter,
        operand2    Filter
    },
    orExpression    SEQUENCE    {
        operand1    Filter,
        operand2    Filter
    },
    negExpression    Filter
}

ObjectSetParameters    ::=    {
    -- This type fits to provide a call of the ObjectSet
    -- function (see C.1) with parameter values
    object    ObjectName,
    level     ENUMERATED    {
        self, directContained, contained, allSubtree
    },
    filter    Filter
}

EventCountParameters    ::=    SEQUENCE    {
    -- This type fits to provide a call of the EventCount
    -- function (see C.1) with parameter values
    objectSet    ObjectSetParameters,
    eventType    ENUMERATED    {
        objectCreation, objectDeletion, stateChnaged,
        attributeChanged, ...
    },
    timeslot     INTEGER
}

CountItem    ::=     CHOICE    {
    objects    ObjectSetParameters,
    events     EventCountParameters
}

PolicyCondition    ::=    SEQUENCE OF SEQUENCE    {
    -- Number of objects/events
    countItem       CountItem,
    operator        LogicalOperator,
    comparedWith    CHOICE    {
        definedValue      INTEGER,
        otherCountItem    CountItem
    }
}

-- END Policy Condition Related Data Types
```

```
-----------------------------------------------------------

-----------------------------------------------------------
-- START Policy Action Related Data Types

ObjectOperation    ::=    SEQUENCE    {
    baseObject    OptionalObjectName,
    level    ENUMERATED    {
        self, directContained, contained, allSubtree
    },
    filter       Filter,
    operation    CHOICE    {
        valueAssertion    SEQUENCE    {
            operationType    ENUMERATED    {
                guaranteeValues, avoidValues
            },
            attributeName    PrintableString OPTIONAL,
            valueList    SET OF CHOICE    {
                definedValue    ANY,
                definedRange    ValueRange
            }
        },
        objectPresence    SEQUENCE    {
            operationType    ENUMERATED    {
                mustExist, mustNotExist
            },
            className    PrintableString,
            attributeValueConstraints    SET OF SEQUENCE    {
                attributeName     PrintableString,
                fieldName         PrintableString OPTIONAL,
                attributeValue    ANY
            }
        }
    },
    timeSlot    INTEGER
}

ObjectOperations    ::=    SEQUENCE OF    ObjectOperation

-- END Policy Action Related Data Types
-----------------------------------------------------------

-- END Policy Related Data Types
-----------------------------------------------------------
-----------------------------------------------------------


-----------------------------------------------------------
-----------------------------------------------------------
-- START Terminal Related Data Types

-----------------------------------------------------------
-- START Services Related Data Types

Services    ::=    SEQUENCE OF SEQUENCE    {
    serviceName    PrintableString
    serviceCost    ANY
}

-- END Services Related Data Types
-----------------------------------------------------------

-----------------------------------------------------------
-- START Appliction Related Data Types

Direction    ::=    ENUMERATED    {
    downlink, uplink
}
```

```
TrafficClassId    ::=    ENUMERATED    {
    conversational, streaming, interactive, background, ...
}

TrafficClass    ::= CHOICE    {
    id      TrafficClassId
    void    NULL
}


QoSRequirementId    ::=    ENUMERATED    {
    maximumDelay, maximumDelayVariation, maximumPacketLoss,
    minimumBandwidth, preferredBandwidth, ...
}

QoSRequirements    ::=     SEQUENCE OF SEQUENCE    {
    qoSRequirementName     QoSRequirementId
    qoSRequirementValue    ANY
}


Links    ::=    SEQUENCE OF OptionalObjectName


ApplicationMeasurementId    ::=    ENUMERATED    {
    observedDelay, observedDelayVariation, observedPacketLoss,
    observedBandwidth, ...
}

ApplicationMeasurementIds    ::=    SEQUENCE OF {
    ApplicationMeasurementId
}

ApplicationMeasurements    ::=     SEQUENCE OF SEQUENCE    {
    applicationMeasurementName     ApplicationMeasurementId
    applicationMeasurementValue    ANY
}

-- END Application Related Data Types
-----------------------------------------------------------

-----------------------------------------------------------
-- START Device Related Data Types

DeviceOptionId    ::=    ENUMERATED    {
    maximumTxPower, maximumNumberOfRadioInterfaces, ...
}

DeviceOptions    ::=     SEQUENCE OF SEQUENCE    {
    deviceOptionName     DeviceOptionId
    deviceOptionValue    ANY
}


DeviceMeasurementId    ::=    ENUMERATED    {
    deviceLocation, batteryPower, ...
}

DeviceMeasurementIds    ::=    SEQUENCE OF {
    DeviceMeasurementId
}

DeviceMeasurements    ::=     SEQUENCE OF SEQUENCE    {
    deviceMeasurementName     DeviceMeasurementId
    deviceMeasurementValue    ANY
}


LinkMeasurementId    ::=    ENUMERATED    {
    receivedPower, receivedSINR, ...
```

```
}

LinkMeasurementIds    ::=    SEQUENCE OF {
    LinkMeasurementId
}

LinkMeasurements    ::=     SEQUENCE OF SEQUENCE    {
    linkMeasurementName     LinkMeasurementId
    linkMeasurementValue    ANY
}


ChannelMeasurementId    ::=    ENUMERATED    {
    channelInterference, channelLoad, ...
}

ChannelMeasurementIds    ::=    SEQUENCE OF {
    ChannelMeasurementId
}

ChannelMeasurements    ::=     SEQUENCE OF SEQUENCE    {
    channelMeasurementName     ChannelMeasurementId
    channelMeasurementValue    ANY
}

-- END Device Related Data Types
-----------------------------------------------------------

-- END Terminal Related Data Types
-----------------------------------------------------------
-----------------------------------------------------------

-----------------------------------------------------------
-----------------------------------------------------------
-- START CWN Related Data Types

-----------------------------------------------------------
-- START RAN Related Data Types

RANUsers    ::=    SEQUENCE OF SEQUENCE    {
    userId      OptionalObjectName
    userData    ANY
}

-- END RAN Related Data Types
-----------------------------------------------------------

-----------------------------------------------------------
-- START Transport Interface Related Data Types

TransportInterface    ::=    SEQUENCE    {
    transportTechnology     PrintableString,
    bandwidth               REAL,
    userPlaneBandwidth      REAL OPTIONAL,
    controlPlaneBadwidth    REAL OPTIONAL,
    omBandwidth             REAL OPTIONAL
}

TransportInterfaces    ::=    SEQUENCE OF TransportInterface

-- END Transport Interface Related Data Types
-----------------------------------------------------------

-----------------------------------------------------------
-- START Base Station Related Data Types

BaseStationOptionId    ::=    ENUMERATED    {
    maximumTxPower, maximumNumberOfRadioInterfaces, ...
}
```

```
BaseStationOptions    ::=     SEQUENCE OF SEQUENCE    {
    baseStationOptionName     BaseStationOptionId
    baseStationOptionValue    ANY
}


BaseStationMeasurementId    ::=    ENUMERATED    {
    transmitPower, transportLoad, processingLoad, ...
}

BaseStationMeasurementIds    ::=    SEQUENCE OF {
    BaseStationMeasurementId
}

BaseStationMeasurements    ::=     SEQUENCE OF SEQUENCE    {
    baseStationMeasurementName     BaseStationMeasurementId
    baseStationMeasurementValue    ANY
}

-- END Base Station Related Data Types
-----------------------------------------------------------

-----------------------------------------------------------
-- START Cell Related Data Types

CellOptionId    ::=    ENUMERATED    {
    maximumTxPower, ...
}

CellOptions    ::=     SEQUENCE OF SEQUENCE    {
    cellOptionName     CellOptionId
    cellOptionValue    ANY
}


CellMeasurementId    ::=    ENUMERATED    {
    transmitPower, cellLoad, trafficLoad, cellThroughput,
    cellInterference, ...
}

CellMeasurementIds    ::=    SEQUENCE OF {
    CellMeasurementId
}

CellMeasurements    ::=     SEQUENCE OF SEQUENCE    {
    cellMeasurementName     CellMeasurementId
    cellMeasurementValue    ANY
}


AntennaConfiguration    ::=    CHOICE    {
    omnidirectional    NULL,
    beamforming        REAL,
    ...
}

AntennaConfigurations    ::=     SEQUENCE OF    {
    AntennaConfiguration
}


Terminals    ::=     SEQUENCE OF    OptionalObjectName

-- END Cell Related Data Types
-----------------------------------------------------------

-- END CWN Related Data Types
-----------------------------------------------------------
-----------------------------------------------------------
END
```

Annex D

(informative)

Information model extensions and usage example

D.1 Functions for external management interface

For the purpose of describing the usage example, the NRM and TRM entities are considered to be managing and managed systems, respectively. Both systems share the knowledge on terminal and policy-related classes of the information model. It is assumed that the managed system provides at its external interface a set of primitives representing typical management functions. These functions are used by the managing system assuming that an underlying communication protocol between managing and managed systems is available. It is important to note that the functions are designed in a way where in one invocation they perform operations on a set of managed objects.

The description of the function is as follows.

Set attribute values in a set of objects:

```
SetAttributeValue (baseObject, level, filter
                        [, attributeName, value]+)
```

Sets the attributes from the list to their respective values in all objects selected by ObjectSet (baseObject, level, filter).

Get attribute values in a set of objects:

```
GetAttributeValue (baseObject, level, filter
                        [, attributeName]+)
```

Returns a list of triples (objectName, attributeName, value) with the values for all requested attributes in all objects selected by ObjectSet (baseObject, level, filter).

Create an object:

```
CreateObject (baseObject, level, filter, className
                        [, attributeName, value]+)
```

Creates an object of class given by className contained in all objects selected by ObjectSet (baseObject, level, filter). Initializes attribute values as given in the parameter list. Returns the name of the new object (relative to the baseObject's name).

Delete an object:

```
DeleteObject (baseObject, level, filter)
```

Deletes all objects selected by ObjectSet (baseObject, level, filter). In case one of these objects contains further objects, those are also deleted by the operation.

Finally, the managing system provides at its external interface a primitive representing the function of the managing system to receive a spontaneous event report from the managed system. The content of this message is defined in a corresponding class provided for each type of event report.

Send a report:

```
EventReport(reportType, reportingObject, reportData)
```

D.2 Additional utility classes

To facilitate efficient exchange of information, it is useful to incorporate common utility classes to apply the required statistical operations, filters (that is, selection criteria), trigger thresholds, and other mechanisms that can optimize the efficiency of information exchanges.

The following utility classes are used within this annex in order to control on behalf of their instances common functions in a managed system:

— **Threshold**
  An object of this class, once instantiated in the scope of another (its superior) object, provides the possibility to define threshold values for an attribute of the superior object. Attribute thresholds can be use to generate corresponding threshold crossed events.

— **Value characteristic**
  An object of this class, once instantiated in the scope of another (its superior) object, provides the possibility to provide for the values of an attribute of this superior a characteristic over a given interval of time (window) taking into account those values collected each time a smaller sampling interval time is elapsed. Characteristics are defined as one of min, max, mean value, standard deviation, or just the values sequence (for collecting the value history). The window may be fixed (meaning the next window starts after the previous window time completely elapsed) or sliding (meaning that after each sampling interval the window is shifted by the sampling interval time). Further, accuracy and precision may be defined for the measured values considered in the calculation of the characteristic. The latest available characteristic is provided as the 'current value' attribute of this class.

— **Scheduler**
  An object of this class causes the generation of timeout events in the managed system in a periodic fashion. By specifying a stop time, only a certain number (even one) of timeout events may be specified. Other objects may refer to a scheduler object with the purpose to get triggered by the timeout event for specific actions.
  A start time for a scheduler object may be specified (to allow starting timeout generation later than at creation time), an operational state is added in order to enable/disable the timeout generation from the managing system if needed.

— **Measurement reporter**
  An object of this class, once instantiated in the scope of another (its superior) object, provides the possibility to end a collection of current values of a specified set of attributes of the superior

or contained therein objects to the managing system in form of a measurement report. Therefore, it refers to a scheduler object in order to get triggers for sending the reports. The possibility to specify the managing system is foreseen; while currently only NRM is assumed as such managing system. Further, an attribute 'timeEllapsedSinceLastReport' is defined. This may especially be used to trigger an immediate measurement report (by setting its value to 0).

— **Report Generator**
An object of this class is used to dynamically control the event reports sent to a managing system. Generic event types are defined for the objects in the managed system, but it might be worth spontaneously reporting them to a managing system in some circumstances or at special time intervals. Therefore, a **Report Generator** must be created that specifies the event type, the target system, and (optionally) filter criteria for the data members of the report. Otherwise, no event reports are sent (except measurement reports that are not an accepted event type for **Report Generator** objects).

— **Event Report** (and derived specific reports)
Objects of these classes are just data objects that define what data are expected to be delivered to the managing system in case an event of a given type occurred and the managed system is requested to send an event report on that. Derived reports include the following:

  — **Object Creation Report**
  — **Object Deletion Report**
  — **Attribute Value Change Report**
  — **Measurement Report**
  — **High Threshold Crossed Report**
  — **Low Threshold Crossed Report**

Table D.1 through Table D.12 describe each utility class in detail.

The UML class diagram for utility classes is shown in Figure D.1.

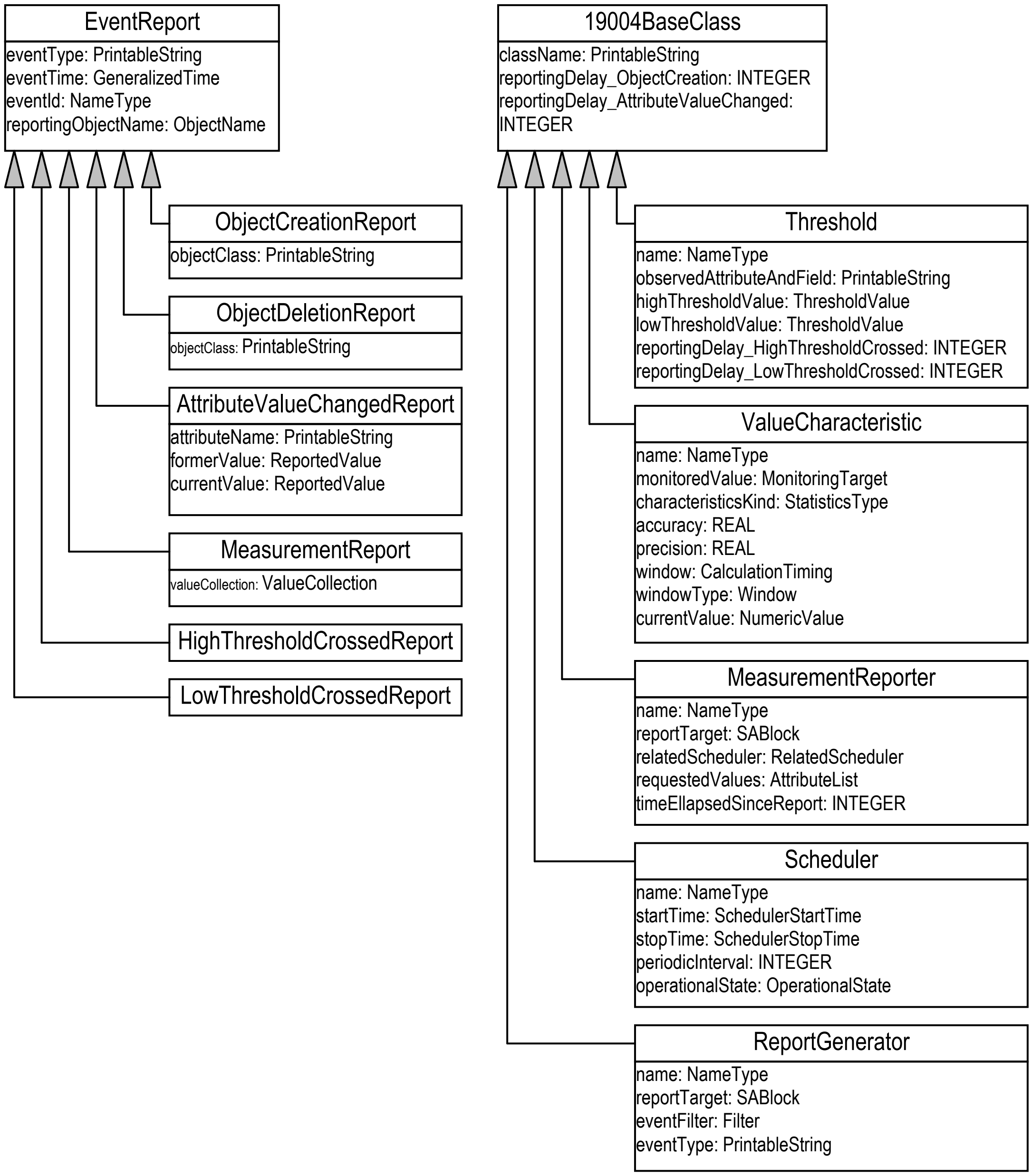


Figure D.1—UML class diagram for utility classes

Table D.1—EventReport class definition

Class **EventReport** *(abstract class)*			
This is the mandatory base class for all IEEE 1900.4 event reports. An event report object is created if a) an event of the specified event type has occurred, and b) an EventReporter object or an ECAPolicy object are requiring the creation of the event report.			
DERIVED FROM			
ATTRIBUTES			
eventType	*Value type:* PrintableString	*Possible access:* Read	*Default value:* *- not specified -*
This attribute defines the type of the reported event.			
eventTime	*Value type:* GeneralizedTime	*Possible access:* Read	*Default value:* *- not specified -*
This attribute defines the time at that the event occurred.			
eventId	*Value type:* NameType	*Possible access:* Read	*Default value:* *- not specified -*
This attribute contains string or number assigned to uniquely identify the event report object.			
reportingObjectName	*Value type:* ObjectName	*Possible access:* Read	*Default value:* *- not specified -*
This attribute defines the object that reported the occurring event.			
CONTAINED IN			
CONTAINS			
SUPPORTED EVENTS			

Table D.2—ObjectCreationReport class definition

Class **ObjectCreationReport**			
The ObjectCreationReport object is generated if a new object has been created (objectCreation event type). This may be of interest in case an object occurs during normal operation and a managing system needs to be aware of it.			
DERIVED FROM	**EventReport**		
ATTRIBUTES			
objectClass	*Value type:* PrintableString	*Possible access:* Read	*Default value:* *- not specified -*
This attribute reports the class of the created object.			
CONTAINED IN			
CONTAINS			
SUPPORTED EVENTS			

Table D.3—ObjectDeletionReport class definition

Class **ObjectDeletionReport**			
The ObjectDeletionReport is generated if a object has been deleted (objectDeletion event type). This may be of interest in case an object is deleted during normal operation and a managing system needs to be aware of it.			
DERIVED FROM	**EventReport**		
ATTRIBUTES			
objectClass	*Value type:* PrintableString	*Possible access:* Read	*Default value:* *- not specified -*
This attribute reports the class of the deleted object.			
CONTAINED IN			
CONTAINS			
SUPPORTED EVENTS			

Table D.4—AttributeValueChangedReport class definition

Class **AttributeValueChangedReport**			
The AttributeValueChangedReport is generated after the value of an attribute has been changed (attributeValueChanged event type). This may be of interest in case the attribute value is changed during normal operation and a managing system needs to be aware of this value change.			
DERIVED FROM	**EventReport**		
ATTRIBUTES			
attributeName	*Value type:* PrintableString	*Possible access:* Read	*Default value: - not specified -*
This attribute reports the name of the attribute that changed its value.			
formerValue	*Value type:* ReportedValue	*Possible access:* Read	*Default value: - not specified -*
This attribute reports the former attribute value.			
currentValue	*Value type:* ReportedValue	*Possible access:* Read	*Default value: - not specified -*
This attribute reports the new attribute value.			
CONTAINED IN			
CONTAINS			
SUPPORTED EVENTS			

Table D.5—MeasurementReport class definition

Class **MeasurementReport**			
The MeasurementReport is generated by a MeasurementReporter object after a measurement period is over (measurementReport event type). It contains all values defined in the MeasurementReporter object.			
DERIVED FROM	**EventReport**		
ATTRIBUTES			
valueCollection	*Value type:* ValueCollection	*Possible access:* Read	*Default value: - not specified -*
This attribute reports the list of measured values.			
CONTAINED IN			
CONTAINS			
SUPPORTED EVENTS			

Table D.6—HighThresholdCrossedReport class definition

Class **HighThresholdCrossedReport**	
The HighThresholdCrossedReport is generated if the attribute value of the monitored attribute in a Threshold object has exceeded the highThresholdValue (highThresholdCrossed event type).	
DERIVED FROM	**EventReport**
ATTRIBUTES	
CONTAINED IN	
CONTAINS	
SUPPORTED EVENTS	

Table D.7—LowThresholdCrossedReport class definition

Class **LowThresholdCrossedReport**	
The LowThresholdCrossedReport is generated if the attribute value of the monitored attribute in a Threshold object has gone below the lowThresholdValue (lowThresholdCrossed event type).	
DERIVED FROM	**EventReport**
ATTRIBUTES	
CONTAINED IN	
CONTAINS	
SUPPORTED EVENTS	

Table D.8—Threshold class definition

Class **Threshold**			
This class provides the possibility to define various threshold crossed event reports for an object.			
DERIVED FROM	**19004BaseClass**		
ATTRIBUTES			
name	*Value type:* NameType	*Possible access:* Read	*Default value:* *- not specified -*
This attribute contains string or number assigned to uniquely identify the Threshold object.			
observedAttributeAndField	*Value type:* PrintableString	*Possible access:* Read-Write	*Default value:* *- not specified -*
This attribute defines what attribute (and optionally, field) of the superior object has to be monitored. Its value must allow for lower/higher comparison.			
highThresholdValue	*Value type:* ThresholdValue	*Possible access:* Read-Write	*Default value:* *- not specified -*
This attribute defines the high threshold value. If NULL – no high threshold passed event is generated.			
lowThresholdValue	*Value type:* ThresholdValue	*Possible access:* Read-Write	*Default value:* *- not specified -*
This attribute defines the low threshold value. If NULL – no low threshold passed event is generated.			
reportingDelay_High ThresholdCrossed	*Value type:* INTEGER	*Possible access:* Read-Write	*Default value:* 0
This attribute defines a delay between internal event detection and event reporting (if the monitored value goes again below the threshold during this time – no report will be generated)			
reportingDelay_LowThreshold Crossed	*Value type:* INTEGER	*Possible access:* Read-Write	*Default value:* 0
This attribute defines the delay between internal event detection and event reporting (if the monitored value goes again above the threshold during this time – no report will be generated)			
CONTAINED IN	**ApplicationMeasurements, DeviceMeasurements, LinkMeasurements, ChannelMeasurements, BaseStationMeasurements, CellMeasurements**		
CONTAINS			
SUPPORTED EVENTS	highThresholdCrossed, lowThresholdCrossed		

Table D.9—ValueCharacteristic class definition

Class **ValueCharacteristic**			
This class allows for defining different statistics to be calculated for attribute values of a managed object. It may be contained (created) within an arbitrary object, depending on the operational needs.			
DERIVED FROM	**19004BaseClass**		
ATTRIBUTES			
name	*Value type:* NameType	*Possible access:*	*Default value:* *- not specified -*
This attribute contains string or number assigned to uniquely identify the ValueCharacteristic object.			
monitoredValue	*Value type:* MonitoringTarget	*Possible access:* Read-Write	*Default value:* *- not specified -*
This attribute defined what has to be monitored in the monitored object. This may be simply the value of an attribute, the number of occurrences of a given event type or the number of objects of a given class contained in the monitored object.			
characteristicsKind	*Value type:* StatisticsType	*Possible access:* Read-Write	*Default value:* *- not specified -*
This attribute defines what calculation shall be applied to the monitored values.			
accuracy	*Value type:* REAL	*Possible access:* Read-Write	*Default value:* 0
This optional attribute defines the accuracy applied for measuring. Default: not specified			
precision	*Value type:* REAL	*Possible access:* Read-Write	*Default value:* 0
This optional attribute defines the precision applied for for measuring. Default: not specified			
window	*Value type:* CalculationTiming	*Possible access:* Read-Write	*Default value:* *- not specified -*
This attribute defines the time interval (in ms) during that the calculation shall be applied. Further, the sampling period can be defined in this attribute.			
windowType	*Value type:* Window	*Possible access:* Read-Write	*Default value:* *- not specified -*
This attribute defines if the currentValue calculation is updated after each sampling time interval or after each window time interval.			
currentValue	*Value type:* NumericValue	*Possible access:* Read	*Default value:* *- not specified -*
This attribute provides the latest calculated value			
CONTAINED IN	**ApplicationMeasurements, DeviceMeasurements, LinkMeasurements, ChannelMeasurements, BaseStationMeasurements, CellMeasurements**		
CONTAINS			
SUPPORTED EVENTS			

Table D.10—MeasurementReporter class definition

Class **MeasurementReporter**			
This class providing a flexible way to specify various measurement reports containing a list of different attribute values observed at a certain time (usually these values represent measurement results). Further, this class includes the ability to send the reports to defined targets in accordance with a specified scheduler.			
DERIVED FROM	**19004BaseClass**		
ATTRIBUTES			
name	*Value type:* NameType	*Possible access:* Read	*Default value:* *- not specified -*
This attribute contains string or number assigned to uniquely identify the MeasurementReporter object.			
reportTarget	*Value type:* SABlock	*Possible access:* Read-Write	*Default value:* *- not specified -*
This attribute defines the target entity where the measurement reports send to.			
relatedScheduler	*Value type:* RelatedScheduler	*Possible access:* Read-Write	*Default value:* *- not specified -*
This attribute defines the scheduler that controls the sending of measurement reports. If NULL – no measurement reporting times are defined.			
requestedValues	*Value type:* AttributeList	*Possible access:* Read, Add-Remove	*Default value:* *- not specified -*
This attribute defines all values that must be included into in the measured report. For example, currentValue attributes of ValueCharacteristic objects can be included into this list.			
timeEllapsedSinceReport	*Value type:* INTEGER	*Possible access:* Read-Write	*Default value:* *- not specified -*
This attribute provides the time ellapsed since the last report. Measured unit is ms. If no report has been done yet - will be reported as –1. Setting to 0 causes immediate measurement report.			
CONTAINED IN	**BaseStationMeasurements, CellMeasurements, ApplicationMeasurements, DeviceConfiguration, DeviceMeasurements**		
CONTAINS			
SUPPORTED EVENTS	MeasurementReport		

Table D.11—Scheduler class definition

Class **Scheduler**			
This class provides the possibility to define various schedulers to schedule different operations in a common way.			
DERIVED FROM	**19004BaseClass**		
ATTRIBUTES			
name	*Value type:* NameType	*Possible access:* Read	*Default value:* *- not specified -*
This attribute contains string or number assigned to uniquely identify the Scheduler object.			
startTime	*Value type:* SchedulerStartTime	*Possible access:* Read-Write	*Default value:* *- not specified -*
This attribute defines the time when the scheduled object starts the periodic timer for the scheduler.			
stopTime	*Value type:* SchedulerStopTime	*Possible access:* Read-Write	*Default value:* *- not specified -*
This attribute defines the time when the scheduled object stops the periodic timer for the scheduler. On behalf of this a scheduler may be used as a single timer. If NULL - no stop time defined.			
periodicInterval	*Value type:* INTEGER	*Possible access:* Read-Write	*Default value:* *- not specified -*
This attribute defines the time (in ms) for one interval			
operationalState	*Value type:* OperationalState	*Possible access:* Read-Write	*Default value:* enabled
This attribute allows to stop or re-start the scheduler.			
CONTAINED IN	**BaseStationMeasurements, CellMeasurements, ApplicationMeasurements, DeviceConfiguration, DeviceMeasurements, …**		
CONTAINS			
SUPPORTED EVENTS			

Table D.12—ReportGenerator class definition

Class **ReportGenerator**			
This class provides the possibility to define various reports on reportable events of an object. Reports will be sent on event occurrence.			
DERIVED FROM	**19004BaseClass**		
ATTRIBUTES			
name	*Value type:* NameType	*Possible access:* Read	*Default value:* *- not specified -*
This attribute contains string or number assigned to uniquely identify the ReportGenerator object.			
reportTarget	*Value type:* SABlock	*Possible access:* Read-Write	*Default value:* *- not specified -*
This attribute defines the target where the reports send to.			
eventFilter	*Value type:* Filter	*Possible access:* Read-Write	*Default value:* *- not specified -*
This attribute defines the filter to be applied. It prevents too many event or undesired reports.			
eventType	*Value type:* PrintableString	*Possible access:* Read-Write	*Default value:* *- not specified -*
This attribute defines the event type to be reported. Values of all report parameters defined for this event type must be included into the report.			
CONTAINED IN	**Terminal, BaseStation, Cell**		
CONTAINS			
SUPPORTED EVENTS			

D.3 Additional ASN.1 type definitions for utility classes

```
1900-4-Utilities-Type-Definitions DEFINITIONS ::= BEGIN

               IMPORTS ObjectName, OptionalObjectName, Filter FROM 1900-4-Type-
Definitions;

               ReportedValue ::= ANY

               ThresholdValue ::= CHOICE {
                   Undefined           NULL,
                   wholeNumber         INTEGER,
                   fractionalNumber    REAL
               }

               MonitoringTarget ::= CHOICE {
                   attributeOrFieldValue SEQUENCE {
                       -- target of monitoring is an attribute or field value
                       attributeName    PrintableString,
                       fieldName        PrintableString OPTIONAL
                   },
                   eventName    PrintableString,
                   -- target of monitoring is the number of occurring events
                   className    PrintableString
                   -- target of monitoring is the number of objects
                   -- of this class contained
                   -- in the monitored object
               }

               StatisticsType ::= ENUMERATED {min, max, mean, standardDeviation, history}
               -- history: store the list of all measured values

               CalculationTiming ::= SEQUENCE {
                   Total             INTEGER,    -- in ms
                   sampleInterval    INTEGER OPTIONAL    -- in ms
               }

               Window ::= ENUMERATED {fix, sliding}

               NumericValue ::= CHOICE {
                   wholeNumber         INTEGER,
                   fractionalNumber    REAL
               }

               SABlock ::= ENUMERATED {nRM, tRM, oSM}

               RelatedScheduler ::= OptionalObjectName

               AttributeList ::= SEQUENCE OF SEQUENCE {
                   level    ENUMERATED    { -- level + filter define a set of objects
                       self, directContained, contained, allSubtree
                   },
                   filter    Filter,
                   attributeName    PrintableString,
                   fieldName    PrintableString OPTIONAL    -- pick up non-constructed
                                                            -- values from attributes
               }

               ValueCollection ::= SEQUENCE OF SEQUENCE {
                   Object    ObjectName,
                   attributeName    PrintableString,
                   fieldName    PrintableString OPTIONAL,    -- pick up non-constructed
                                                           -- values from attributes
                   value    ANY
               }
```

```
               SchedulerStartTime ::= CHOICE {
                   absoluteTime    GeneralizedTime,
                   relativeTime    INTEGER    -- relative to scheduler creation time
               }

               SchedulerStopTime ::= CHOICE {
                   Undefined       NULL,
                   absoluteTime    GeneralizedTime,
                   relativeTime    INTEGER    -- relative to start time
               }

               OperationalState ::= ENUMERATED {enabled, disabled}

END
```

D.4 Example for distributed radio resource usage optimization use case

To implement distributed radio resource usage optimization use case, it is necessary to obtain context information and to generate radio resource selection policies. This example gives illustration of how to apply the IEEE 1900.4 information model for these purposes.

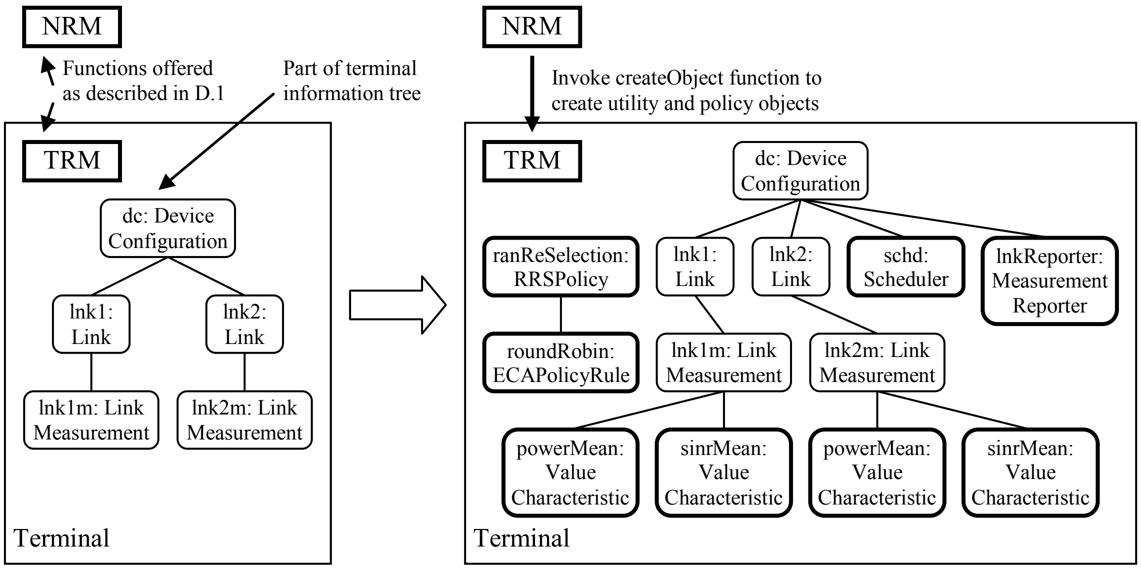
In particular, the following is described:

- Radio resource selection policy, generated by NRM, is received by TRM
- According to this radio resource selection policy TRM performs to following two actions:
  - Action 1: TRM is allowed to select periodically new RAN to which to connect.
  - Action 2: TRM periodically obtains link measurements from TMC and sends measurement report to NRM.

In this example the following is assumed:

- NRM and TRM offer the functions described in D.1
- NRM and TRM use the information model extensions given in D.1, D.2, and D.3

Figure D.2 describes this example. In Figure D.2, objects are described using legend <object name>:<class name>, where object names are selected arbitrarily, while class names are defined in the IEEE 1900.4 information model.



Information tree before applying radio resource selection policy in TRM

It is assumed that there are two active links in the Terminal at the time the radio resource selection policy is generated in NRM and received in TRM

Information tree after creation of necessary objects

(A) ValueCharacteristic object is created for each link measurement to store the value of this measurement
(B) MeasurementReporter object is created to collect the link statistics and to send them as a report to NRM
(C) Scheduler object is created to be used as trigger for applying the radio resource selection policy
Note: Steps (A) – (C) are required to prepare to perform actions specified by radio resource selection policy
(D) Radio resource selection policy object is created
(E) ECA policy rule is added to the radio resource election policy object
Note: During policy creation TRM arranges (on behalf of the specified attribute values) the associations between the radio resource selection policy, the scheduler, and the measurement reporter objects

Figure D.2—Description of example

Steps (A)–(E) in Figure D.2 can be performed on behalf of the function CreateObject, while for the reporting of measurement results the function EventReport can be used. Both functions are defined in D.1. Steps (A)–(E) are described in details in the following tables.

Some objects used for the description of steps (A)–(E) in Table D.13 through Table D.17 are not shown in Figure D.2 for simplicity (each of these cases is commented).

To keep the example short, some attribute values used in Table D.17 are described in a simplified manner compared to their ASN.1 description given in Annex C.

Table D.13—Description of step (A)

Function	Formal parameter	Actual parameter	Comment
CreateObject	baseObject	trm/dc	Device configuration object
	level	contained	Consider all objects contained in the base object
	filter	className == "LinkMeasurement"	This value representation is not an ASN1 value, it is used here for simplicity
	className	ValueCharacteristic	Include object of this class
	attributeName	name	
	value	sinrMean	
	attributeName	monitoredValue	
	value	*attributeOrFieldValue:* *attributeName:* listOfActiveLinkMeasurements *fieldName:* linkMeasurementValue [linkMeasurementId = receivedSINR]	From monitoring target the attributeOrFieldValue option is chosen. The field name uses the following syntax: From the list of active link measurements select that with linkMeasurementId is equal to receivedSINR and take the value from its linkMeasurementValue field.
	attributeName	characteristicsKind	
	value	mean	
	attributeName	window	
	value	*total:* 10000	
	attributeName	windowType	
	value	fix	
CreateObject	baseObject	trm/dc	The device configuration object
	level	contained	Consider all objects contained in the base object
	filter	className == "LinkMeasurement"	This value representation is not an ASN1 value, it is used here for simplicity
	className	ValueCharacteristic	Include object of this class
	attributeName	name	
	value	powerMean	
	attributeName	monitoredValue	
	value	*attributeOrFieldValue:* *attributeName:* listOfActiveLinkMeasurements *fieldName:* linkMeasurementValue [linkMeasurementId = receivedPower]	From monitoring target the attributeOrFieldValue option is chosen. The field name uses the following syntax: From the list of active link measurements select that with linkMeasurementId is equal to receivedPower and take the value from its linkMeasurementValue field.
	attributeName	characteristicsKind	
	value	mean	
	attributeName	window	
	value	*total:* 10000	
	attributeName	windowType	
	value	fix	

Table D.14—Description of step (B)

Function	Formal parameter	Actual parameter	Comment
CreateObject	baseObject	trm/dc	
	level	contained	Consider only the device configuration object
	filter		Empty filter
	className	MeasurementReporter	Create object of this class
	attributeName	name	
	value	linkReporter	
	attributeName	reportTarget	
	value	nRM	
	attributeName	relatedSceduler	
	value	NULL	
	attributeName	valueCollection	
	value	*level:* contained *filter:* className == "ValueCharacteristic" *attributeName:* curentValue	

Table D.15—Description of step (C)

Function	Formal parameter	Actual parameter	Comment
CreateObject	baseObject	trm/dc	
	level	contained	Consider only the device configuration object
	filter		Empty filter
	className	Scheduler	Create object of this class
	attributeName	name	
	value	schd	
	attributeName	startTime	
	value	*absoluteTime* 8:00:00 + random(10000)	Initializing scheduler
	attributeName	stopTime	
	value	undefined	
	attributeName	periodicInterval	
	value	10000	

Table D.16—Description of step (D)

Function	Formal parameter	Actual parameter	Comment
CreateObject	baseObject	trm/dc	
	level	contained	Consider only the device configuration object
	filter		Empty filter
	className	RRSPolicy	Create object of this class
	attributeName	policyRuleSetId	
	value	ranReSelection	

Table D.17—Description of step (E)

Function	Formal parameter	Actual parameter	Comment
CreateObject	baseObject	trm/ranReSelection	
	level	contained	Consider only the RRSPolicy object
	filter		Empty filter
	className	ECAPolicyRule	Create object of this class
	attributeName	policyRuleId	
	value	roundRobin	
	attributeName	validityTimeConstraint	
	value		
	attributeName	validitySpaceConstraint	
	value		
	attributeName	event	
	value	*eventType:* scheduledTimer *eventQualifier:* trm/dc/schd	
	attributeName	condition	
	value	true	
	attributeName	action	
	value	[*baseObject:* trm *level:* contained *filter:* className == "LinkProfile" *operationType*: guarateeValues *attributeName:* Cell(associatedCellId). CellConfiguration. activeRadioInterface *valueSet:* [lte, wifi] *timeslot:* 3], [*baseObject:* trm/dc/lnkReporter *level:* self *filter:* true *operationType*: guarateeValues *attributeName:* timeEllapsedSinceReport *valueSet:* [0] *timeslot:* 1]	Action 1: Selecting new RAN to connect to Note: LinkProfile object inside TRM is not shown in Figure D.2 for simplicity. Cell and CellConfiguraiton objects inside NRM are not shown in Figure D.2 for simplicity. Action 2: Sending measurement report

Annex E

(informative)

Deployment examples

E.1 Introduction

Referring to the system architecture in Clause 6, this standard specifies the following interfaces:

On the terminal side, interfaces between TRM and TMC for terminal context information collection and between TRM and TRC for reconfiguration management are specified.

Interface between NRM and TRM for policy-based management is specified.

On the network side, interfaces between OSM and NRM for policy-based management, between NRM and RMC for RAN context information collection, and between NRM and RRC for reconfiguration management are specified.

If there are several NRMs, interface between these NRMs for context information exchange and coordination of decision making is specified.

This annex presents examples of deployment of IEEE 1900.4 entities in heterogeneous wireless environment.

The following three scenarios, corresponding to three scenarios of dynamic spectrum assignment use case, are considered:

- Single operator scenario
- Multiple operator scenario 1 (NRM is inside operator)
- Multiple operator scenario 2 (NRM is outside operator)

For each of these scenarios, seven different variants of deployment of RMC and RRC entities are possible:

- In packet-based core network
- In RAN
- In BS
- In a combination of the above

NRM, RMC, and RRC may be implemented in one or several separate network nodes.

All these variants are shown for the single operator scenario. To avoid repetition, only a part of these variants is shown for two multiple operator scenarios. Deployment examples for the multiple operator scenario mainly show differences with the single operator scenario.

For each of the presented deployment examples, this annex clearly shows IEEE 1900.4 entities. It also clearly highlights interfaces specified in this standard and interfaces that are not specified.

E.2 Deployment examples for single operator scenario

Figure E.1 and Figure E.2 show deployment examples 1 and 2 for single operator scenario.

In both figures, RMC and RRC are deployed in packet-based core network only.

In example 1 (Figure E.1) NRM, RMC, and RRC are deployed in packet-based core network. They are implemented in one network node.

In example 2 (Figure E.2) NRM, RMC, and RRC are also deployed in packet-based core network. But they are implemented in three different network nodes.

In these two deployment examples, the following interfaces specified in this standard are used:

- Interface between OSM and NRM
- Interface between NRM and RMC
- Interface between NRM and RRC
- Interface between NRM and TRM
- Interface between TRM and TMC
- Interface between TRM and TRC

RMC deployed in packet-based core network obtains RAN context information via Reconfiguration and Measurement SAP (rCFG_MEDIA_SAP), provided by packet-based core network. TMC obtains terminal context information via Reconfiguration and Measurement SAP (rCFG_MEDIA_SAP), provided by Terminal.

RRC deployed in packet-based core network controls reconfiguration of RAN via Reconfiguration and Measurement SAP (rCFG_MEDIA_SAP), provided by packet-based core network. TRC controls reconfiguration of Terminal via Reconfiguration and Measurement SAP (rCFG_MEDIA_SAP), provided by Terminal.

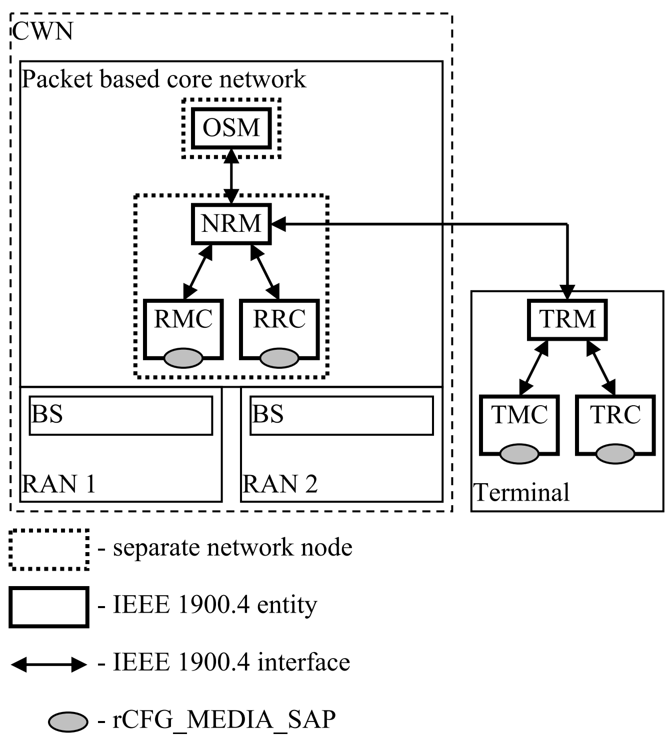


Figure E.1—Single operator scenario, deployment example 1

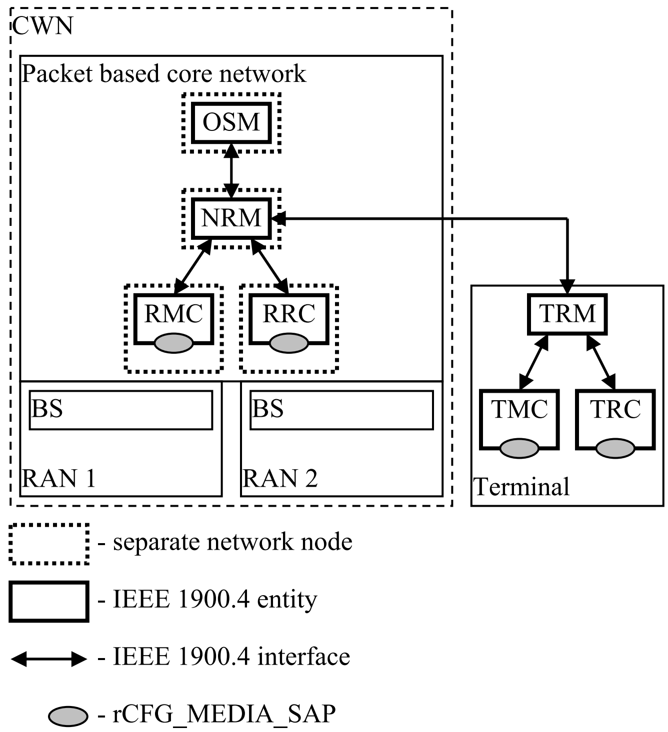


Figure E.2—Single operator scenario, deployment example 2

Figure E.3 shows deployment example 3 for single operator scenario.

In this example, RMC and RRC are deployed as follows:

— RMC and RRC are deployed in RAN only for RAN 1, while BSs of RAN 1 are legacy BSs
— RMC and RRC are deployed in BSs only for RAN 2
— RMC and RRC are deployed in both RAN and BSs for RAN 3

In RAN 1, RMC and RRC are implemented in one network node.

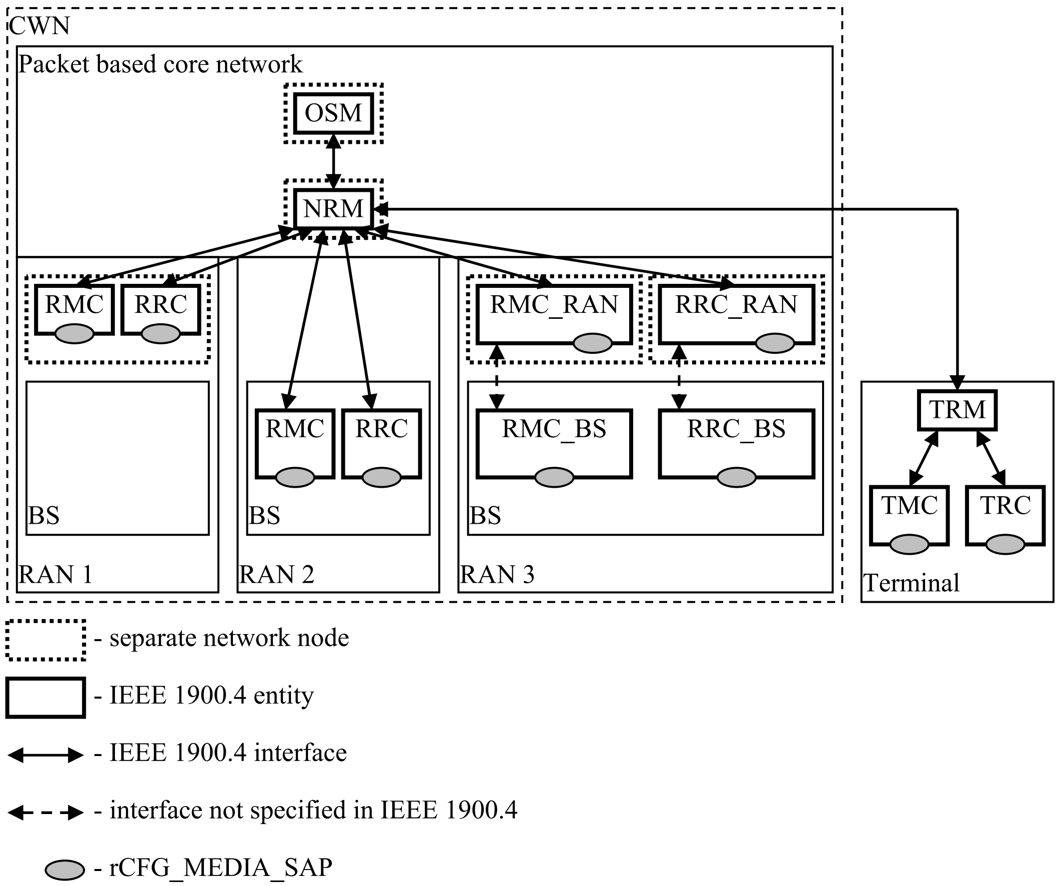
In RAN 3, RAN part of RMC and RRC are implemented in two different network nodes.

In this deployment example, the following interfaces specified in this standard are used:

— Interface between OSM and NRM
— Interface between NRM and RMC
— Interface between NRM and RRC
— Interface between NRM and TRM
— Interface between TRM and TMC
— Interface between TRM and TRC

RMCs deployed in RAN and BSs obtain RAN context information via Reconfiguration and Measurement SAP (rCFG_MEDIA_SAP), provided by RAN and BSs. TMC obtains terminal context information via Reconfiguration and Measurement SAP (rCFG_MEDIA_SAP), provided by Terminal.

RRCs deployed in RAN and BSs control reconfiguration of RAN and BSs via Reconfiguration and Measurement SAP (rCFG_MEDIA_SAP), provided by RAN and BSs. TRC controls reconfiguration of Terminal via Reconfiguration and Measurement SAP (rCFG_MEDIA_SAP), provided by Terminal.



RMC_RAN – part of RMC deployed in RAN
RRC_RAN – part of RRC deployed in RAN
RMC_BS – part of RMC deployed in BS
RRC_BS – part of RRC deployed in BS

Figure E.3—Single operator scenario, deployment example 3

Figure E.4 shows deployment example 4 for single operator scenario.

In this example RMC and RRC are deployed as follows:

- Part of RRC and RMC are deployed in packet-based core network
- Part of RRC and RMC are deployed in RAN for RAN 1
- Part of RRC and RMC are deployed in BSs for RAN 2
- Part of RRC and RMC are deployed in RAN and BSs for RAN 3

Packet-based core network part of RMC and RRC is implemented in one network node.

In RAN 1, RAN part of RMC and RRC is implemented in one network node.

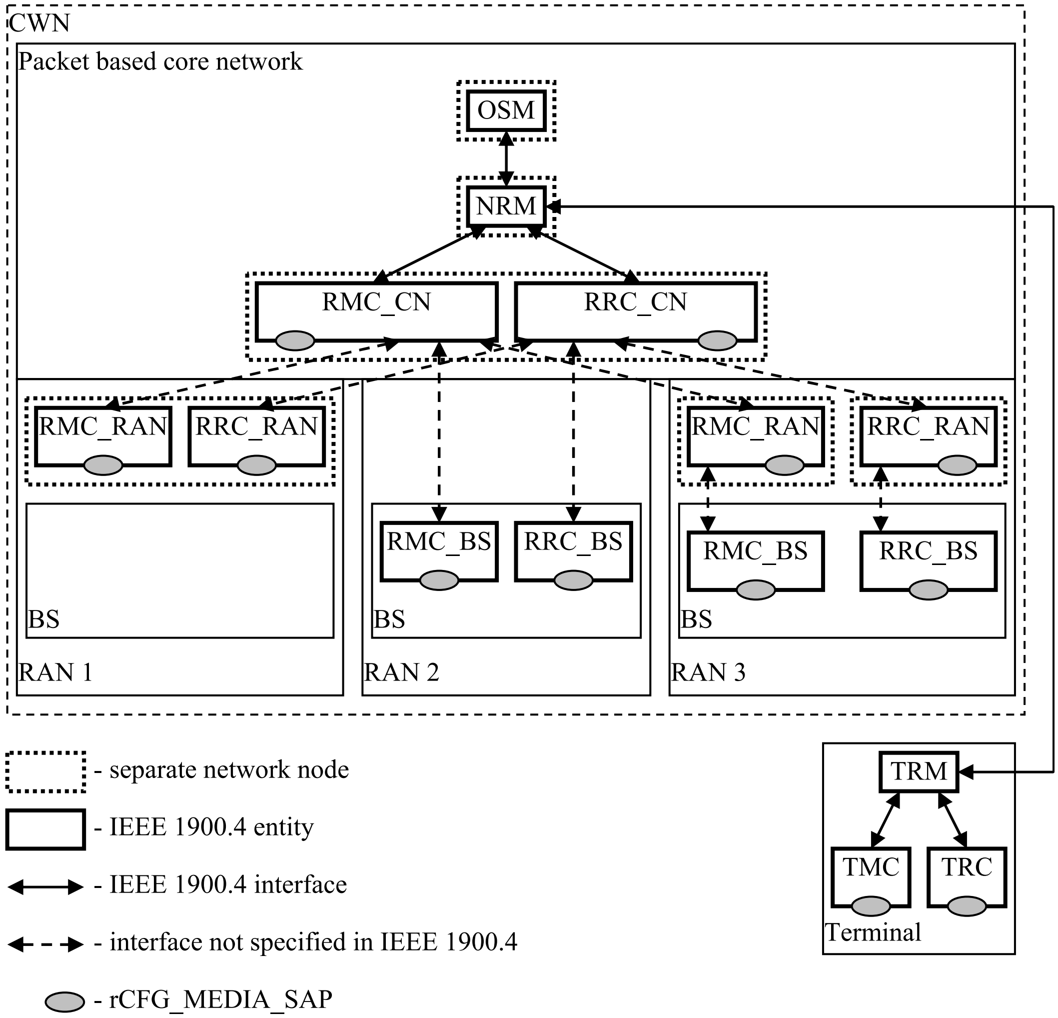
In RAN 3, RAN part of RMC and RRC is implemented in two different network nodes.

In this deployment example, the following interfaces defined in this standard are used:

- Interface between OSM and NRM
- Interface between NRM and RMC
- Interface between NRM and RRC
- Interface between NRM and TRM
- Interface between TRM and TMC
- Interface between TRM and TRC

Parts of RMC deployed in packet-based core network, RAN, and BSs obtain RAN context information via Reconfiguration and Measurement SAP (rCFG_MEDIA_SAP), provided by packet-based core network, RAN, and BSs. TMC obtains terminal context information via Reconfiguration and Measurement SAP (rCFG_MEDIA_SAP), provided by Terminal.

Parts of RRCs deployed in packet-based core network, RAN, and BSs control reconfiguration of RAN and BSs via Reconfiguration and Measurement SAP (rCFG_MEDIA_SAP), provided by packet-based core network, RAN, and BSs. TRC controls reconfiguration of Terminal via Reconfiguration and Measurement SAP (rCFG_MEDIA_SAP), provided by Terminal.



RMC_CN – part of RMC deployed in packet based core network
RRC_CN – part of RRC deployed in packet based core network
RMC_RAN – part of RMC deployed in RAN
RRC_RAN – part of RRC deployed in RAN
RMC_BS – part of RMC deployed in BS
RRC_BS – part of RRC deployed in BS

Figure E.4—Single operator scenario, deployment example 4

E.3 Multiple operator scenario 1 (NRM is inside operator)

Figure E.5 shows deployment example for multiple operator scenario 1, where NRMs are inside operators.

To avoid repetition, in multiple operator scenario 1 deployment example RMC and RRC are deployed in packet-based core network. However, they can be also deployed in a distributed manner as described in deployment examples for the single operator scenario.

In this deployment example, the following interfaces specified in this standard are used:

- Interface between OSM and NRM
- Interface between NRMs of different operators
- Interface between NRM and RMC
- Interface between NRM and RRC
- Interface between NRM and TRM
- Interface between TRM and TMC
- Interface between TRM and TRC

RMC deployed in packet-based core network obtains RAN context information via Reconfiguration and Measurement SAP (rCFG_MEDIA_SAP), provided by packet-based core network. TMC obtains terminal context information via Reconfiguration and Measurement SAP (rCFG_MEDIA_SAP), provided by the Terminal.

RRC deployed in packet-based core network controls reconfiguration of RAN via Reconfiguration and Measurement SAP (rCFG_MEDIA_SAP), provided by packet-based core network. TRC controls reconfiguration of Terminal via Reconfiguration and Measurement SAP (rCFG_MEDIA_SAP), provided by the Terminal.

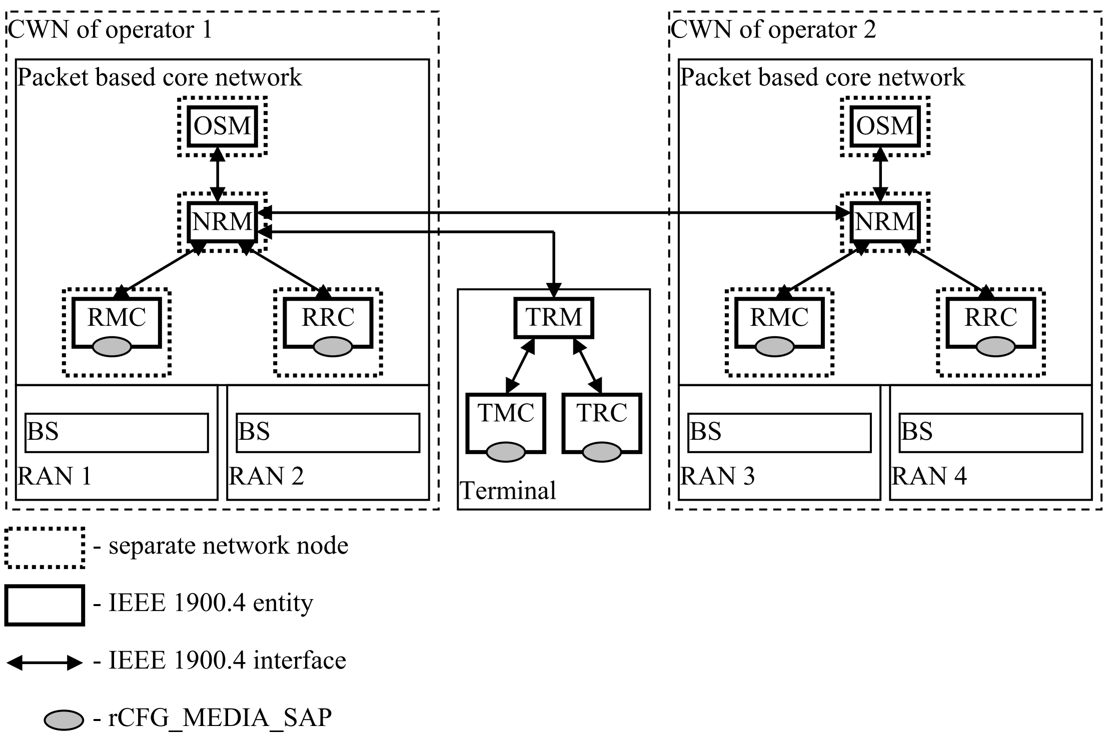


Figure E.5—Deployment example for multiple operator scenario 1

E.4 Multiple operator scenario 2 (NRM is outside operator)

Figure E.6 shows deployment example for multiple operator scenario 2, where NRM is outside operators.

To avoid repetition, in multiple operator scenario 2 deployment example RMC and RRC are deployed in packet-based core network. However, they can be also deployed in a distributed manner as described in deployment examples for single operator scenario.

In this deployment example, the following interfaces specified in this standard are used:

— Interface between OSM and NRM
— Interface between NRM and RMC
— Interface between NRM and RRC
— Interface between NRM and TRM
— Interface between TRM and TMC
— Interface between TRM and TRC

RMC deployed in packet-based core network obtains RAN context information via Reconfiguration and Measurement SAP (rCFG_MEDIA_SAP), provided by packet-based core network. TMC obtains terminal context information via Reconfiguration and Measurement SAP (rCFG_MEDIA_SAP), provided by the Terminal.

RRC deployed in packet-based core network controls reconfiguration of RAN via Reconfiguration and Measurement SAP (rCFG_MEDIA_SAP), provided by packet-based core network. TRC controls reconfiguration of Terminal via Reconfiguration and Measurement SAP (rCFG_MEDIA_SAP), provided by the Terminal.

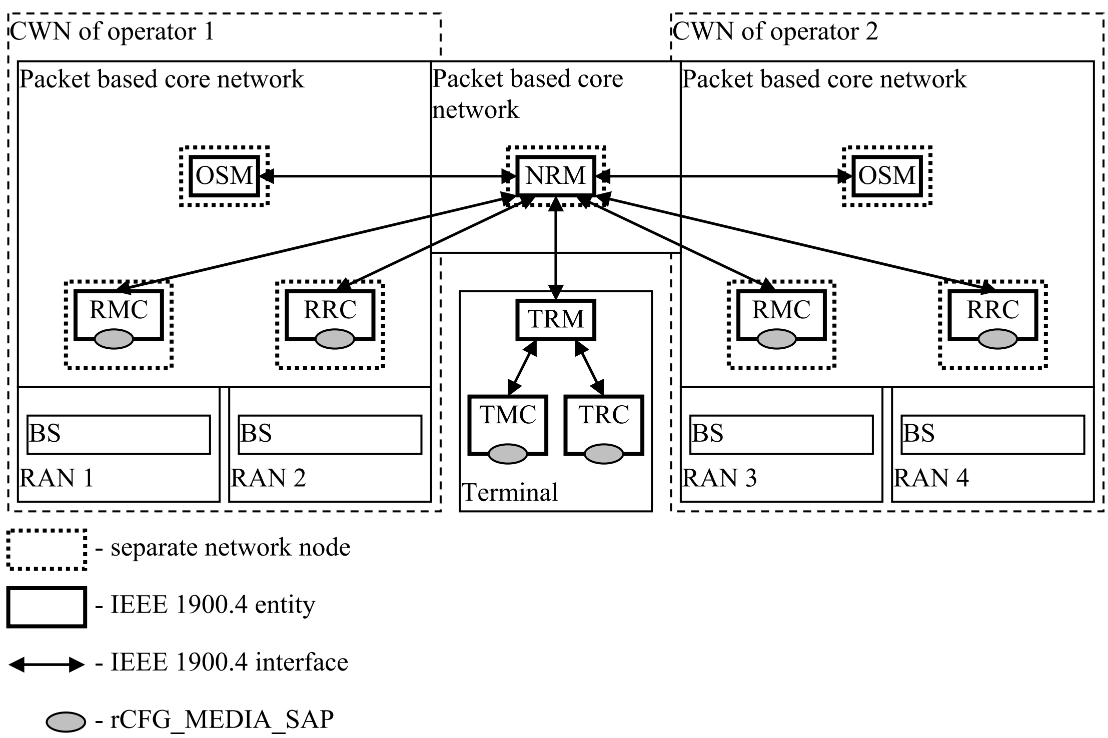


Figure E.6—Deployment example for multiple operator scenario 2

Annex F

(informative)

Bibliography

[B1] 3GPP TS 23.251 V6.6.0, Network Sharing; Architecture and functional description (Release 6), Mar. 2006.

[B2] BBN Technologies, "The XG Vision," Version 2.0, 2004.[12]

[B3] Bourse, D., *et al* "FP7 E3 Project: Introducing Cognitive Wireless Systems in the B3G World," *ICT Mobile Summit 2008*, June 2008.

[B4] Buljore, S., *et al* "IEEE P1900.4 Standard: Reconfiguration of Multi-Radio Systems," *IEEE SIBIRCON 2008*, July. 2008.

[B5] Buljore, S., *et al*, "IEEE P1900.4 System Overview on Architecture and Enablers for Optimised Radio and Spectrum resource usage," *IEEE DySPAN 2008*, Oct. 2008.

[B6] Buljore, S., and Martigne, P., "Proposed System Concept," P1900.4 Working Group, doc: P1900.4-07-04-2007, Apr. 2007.

[B7] Buljore, S., *et al* "Introduction to IEEE P1900.4 Activities," *IEICE Transactions on Communications*, vol. E91-B, no. 1, pp. 2–9, Jan. 2008.

[B8] Buljore, S., and Martigne, P., "SCC41 Plenary Meeting Working Group 4 Overview and Report," IEEE SCC41 Plenary, Apr. 2007.

[B9] Cordeiro, C., Challapali, K., and Birru, D., "IEEE 802.22: An Introduction to the First Wireless Standard based on Cognitive Radios," *Journal of Communications*, vol. 1, no. 1, pp. 38–47, Apr. 2006.

[B10] European Parliament Resolution, "Towards a European policy on radio spectrum," ITRE/6/37236, Feb. 2007.

[B11] FCC ET Docket No. 04-186, "Unlicensed Operation in the TV Broadcast Bands," 2004.

[B12] Filin, S., *et al* "Dynamic Spectrum Assignment and Access Scenarios, System Architecture, and Procedures for IEEE P1900.4 Management System," *Third International Conference on Cognitive Radio Oriented Wireless Networks and Communications (CrownCom 2008)*, May 2008.

[B13] Filin, S., Harada, H., Hasegawa, M., and Kato, S., "QoS-Guaranteed Load-Balancing Dynamic Spectrum Access Algorithm," *IEEE PIMRC 2008*, Sept. 2008.

[B14] Guenin, J., "IEEE Standards Coordinating Committee 41 on Dynamic Spectrum Access," *ITU-R WP5A SDR/CR Seminar*, Feb. 2008.

[B15] Hanaoka, S., Yamamoto, J., and Yano, M., "Platform for Load Balancing and Throughput Enhancement with Cognitive Radio," *IEICE Transactions on Communications*, vol. E91-B, no. 8, pp. 2501–2508, Aug. 2008.

---

[12] This publication is accessible from http://www.ir.bbn.com/projects/xmac/rfc/rfc-vision.pdf.

[B16] Hanaoka, S., Yano, M., and Hirata,T., "Testbed System of Inter-Radio System Switching for Cognitive Radio," *IEICE Transactions on Communications*, vol. E91-B, no. 1 pp. 14–21, Jan. 2008.

[B17] Harada, H., "Software defined radio prototype toward Cognitive Radio Communication Systems," *IEEE DySPAN 2005*, vol. 1, pp. 539–547, Nov. 2005.

[B18] Harada, H., *et al* "A Software Defined Cognitive Radio System," *IEEE Globecom 2007*, pp. 294–299, Nov. 2007.

[B19] Hase, Y., Okada, K., and Wu, G., "A Novel Mobile Basic Access System Using Mobile Access Signaling Card On Telecommunication Systems (MASCOT)," *IPSJ SIG Notes. MBL*, vol. 97, no. 72, pp. 37–42, July 1997.

[B20] Holland, O., Attar, A., Olaziregi, N., Sattari, N., and Aghvami, A. H., "A Universal Resource Awareness Channel for Cognitive Radio," *IEEE PIMRC 2006*, Sept. 2006.

[B21] Holland, O. *et al* "Development of a Radio Enabler for Reconfiguration Management within the IEEE P1900.4 Working Group," *IEEE DySPAN 2007*, pp. 232–239, Apr. 2007.

[B22] IEEE Std 802.21™-2008, IEEE Standard for Local and Metropolitan Area Networks: Media Independent Handover Services.

[B23] IEEE P802.11y™/D11.0, June 2008, Draft 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—Amendment 3: 3650–3700 MHz Operation in USA.[13]

[B24] IEEE P802.22™/D6.0, April 2006, 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.

[B25] Inoue, M., Mahmud, K., Murakami, H., Hasegawa, M., and Morikawa, "Seamless Handover Using Out-Of-Band Signaling in Wireless Overlay Networks," *WPMC 2003*, vol. 1, pp. 186–190, Oct. 2003.

[B26] Inoue, M., Mahmud, K., Murakami, H., Hasegawa, M., and Morikawa, "Novel Out-Of-Band Signaling for Seamless Interworking between Heterogeneous Networks," *IEEE Wireless Communication Magazine*, vol. 11, no. 2, pp. 56–63, Apr. 2004.

[B27] Inoue, M., Mahmud, K., Murakami, H., Hasegawa, M., and Morikawa, H., "Design and Implementation of Out-Of-Band Signaling for Seamless Handover in Wireless Overlay Networks," *IEEE ICC 2004*, pp. 3932–3936, June 2004.

[B28] Ishizu, K., *et al* "Design and Implementation of Cognitive Wireless Network based on IEEE P1900.4," *Third IEEE Workshop on Networking Technologies for Software Defined Radio (SDR) Networks 2008*, June 2008.

[B29] Muck, M., *et al* "IEEE P1900.B: Coexistence Support for Reconfigurable, Heterogeneous Air Interfaces," *IEEE DySPAN 2007*, pp. 381–389, Apr. 2007.

[B30] Muck, M., *et al* "End-to-End Reconfigurability in Heterogeneous Wireless Systems—Software and Cognitive Radio Solutions," *IST Mobile and Wireless Summit 2007*, July 2007.

[13] Numbers preceded by P are IEEE authorized standards projects that were not approved by the IEEE-SA Standards Board at the time this publication went to press. For information about obtaining drafts, contact the IEEE.

[B31] Murakami, H., Mahmud, K., Hasegawa, M., and Inoue, M., "On the Methods of Provisioning Basic Access Signaling for MIRAI," *IEICE Society Conference*, B-5-125, p. 422, Sept. 2002.

[B32] Ofcom, "Spectrum Usage Rights: Technology and Usage Neutral Access to the Spectrum" Consultation, 2006.[14]

[B33] Prasad, R. V., Pawelczak, P., Hoffmeyer, J. A., and Berger, H. S., "Cognitive functionality in next generation wireless networks: standardization efforts," *IEEE Communications Magazine*, vol. 46, no. 4, pp. 72–78, Apr. 2008.

[B34] Seelig, F., "A Description of the August 2006 XG Demonstrations at Fort A.P. Hill," *IEEE DySPAN 2007*, pp. 1–12, Apr. 2007.

[B35] Sherman, M., Mody, A., Martinez, R., Rodriguez, C., and Reddy, R., "IEEE Standards Supporting Cognitive Radio and Networks, Dynamic Spectrum Access, and Coexistence," *IEEE Communications Magazine*, vol. 46, no. 7, pp. 72–79, July 2008.

[B36] Sherman, M., Mody, A., Martinez, R., Reddy, R., and Kiernan, T., "A Survey Of IEEE Standards Supporting Cognitive Radio And Dynamic Spectrum Access," *IEEE MILCOM 2008*, Nov. 2008.

[B37] Strassner, J., "Policy-based network management: solutions for the next generation," Morgan Kaufmann (series in networking), 2005.

[B38] Wu, G., Havinga, P. J. M., and Mizuno, M., "Architecture of Multimedia Integrated network by Radio Access Innovation (MIRAI)," *Technical report of IEICE RCS*, vol. 100, no. 664, pp. 111–119, Mar. 2001.

[B39] Wu, G., Mizuno, M., and Havinga, P. J. M., "MIRAI Architecture for Heterogeneous Network," *IEEE Communications Magazine*, vol. 40, no. 2, pp. 126–134, Feb. 2002.

[14] This publication can be accessed at http://www.ofcom.org.uk/consult/condocs/sur/.