IEEE P802.19
Wireless Coexistence

|  |
| --- |
| Profile O data types |
| Date: 2013-05-16 |
| Author(s): |
| Name | Company | Address | Phone | email |
| Mika Kasslin | Nokia |  |  | mika.kasslin@nokia.com |
| Jari Junell | Nokia |  |  | jari.junell@nokia.com |

Abstract

This document is a submission to IEEE 802.19 TG1 that contains an updated definition of profile O data types.

**Notice:** This document has been prepared to assist IEEE 802.19. It is offered as a basis for discussion and is not binding on the contributing individual(s) or organization(s). The material in this document is subject to change in form and content after further study. The contributor(s) reserve(s) the right to add, amend or withdraw material contained herein.

# Annex A (normative) Data types

# Profile O

IEEE802191ProtocolDataType DEFINITIONS AUTOMATIC TAGS ::= BEGIN

*-- Data type definitions used in messages*EXPORTS SubscribedService, OperationCode, NetworkTechnology, NetworkType, DiscoveryInformation, AvailableChannelsInformation, ListOfAvailableFrequencies, ListOfSupportedFrequencies, ListOfSupportedChNumber , ListOfOperatingFrequencies, ListOfOperatingChNumber, InterferenceDirection, RadioEnvironmentInformation, NetworkGeometryClass, DeviceLocation, RequiredResource, ListOfAllowedTVWSChNumber, ConstOfChUses, ChClassInfo, FrequencyRange, ReqInfoDescr, ReqInfoValue, MeasurementDescription, MeasurementResult, NegotiationInformation, ListOfWinnerCMID, ListOfSlotTimePosition, ReconfigurationParameters, FailedParameters, EventParams, ListOfGeolocation, DatabaseInformation;

CxID ::= ENUMERATED {

 cE\_ID,

 cM\_ID,

 cDIS\_ID,

 tVWSDB\_ID

}

OperationCode ::= ENUMERATED {

 new,

 modify,

 remove

}

SubscribedService ::= ENUMERATED {

 information,

 management,

 interCMCoexistenceSetElements,

 allCoexistenceSetElements

}

NetworkID ::= ENUMERATED {

 bSSID,

 …

}

NetworkTechnology ::= ENUMERATED {

 iEEE802.11af,

 iEEE802.22,

 eCMA392,

 oneSeg,

 …

}

NetworkType ::= ENUMERATED {

 fixed,

 mode1,

 mode2,

 …

}

AvailableChannelsInformation ::= CHOICE {

 listOfAvailableFrequencies AvailableChannelList,

 listOfAvailableChNumber AvailableChNumberList,

 databaseInformation DatabaseInformation

}

AvailableChannelList ::= SEQUENCE OF AvailableChannelElement

AvailableChannelElement ::= SEQUENCE {

 frequencyRange FrequencyRange,

 txPowerLimit REAL, /\* unit [W] \*/

 availableStartTime GeneralizedTime,

 availableDuration REAL /\* unit [s] \*/

}

FrequencyRange ::= SEQUENCE {

 startFreq INTEGER, /\* unit [Hz] \*/

 stopFreq INTEGER /\* unit [Hz] \*/

}

AvailableChNumberList ::= SEQUENCE {

 listOfAllowedTVWSChNumber ListOfAllowedTVWSChNumber,

 listOfAvailableStartTime SEQUENCE OF GeneralizedTime,

 listOfAvailableDuration SEQUENCE OF REAL, /\* unit [s] \*/

 listOfConstOfChUses SEQUENCE OF ConstOfChUses

}

ListOfAllowedTVWSChNumber ::= SEQUENCE OF INTEGER

ConstOfChUses ::= SEQUENCE OF ConstOfChUse

ConstOfChUse ::= SEQUENCE {

 constOfChUseID ConstOfChUseID,

 constOfChUseValue ConstOfChUseValue

}

ConstOfChUseID ::= ENUMERATED {

 regulationMaxTxPower,

 regulationMaxAntGain,

 regulationMaxAntHeight,

 regulationTVDBUpdateTime,

 outOfBandEmissionLimit,

 …

}

ConstOfChUseValue ::= CHOICE {

 regulationMaxTxPower REAL, /\* absolute value in [W] \*/

 regulationMaxAntGain REAL, /\* absolute value \*/

 regulationAntMaxHeight REAL, /\* in [m] \*/

 regulationTVDBUpdateTime REAL, /\* in [s] \*/

 outOfBandEmissionLimit REAL, /\* in [W] \*/

 …

}

DatabaseInformation ::= SEQUENCE {

 dbIpAddress IA5String,

 dbPortNumber INTEGER,

 parameters ParametersForDBAccess

}

ParametersForDBAccess ::= CHOICE {

 fcc FccParam,

 ofcom OfcomParam

}

FccParam ::= SEQUENCE {

 deviceFCCID OCTET STRING,

 deviceSN OCTET STRING

}

OfcomParam ::= SEQUENCE {

 deviceID OCTET STRING,

 deviceSN OCTET STRING,

 modulation Modulation,

 listOfAntenna ListOfAntenna,

 listOfSpectrumMask ListOfSpectrumMask

}

DiscoveryInformation ::= SEQUENCE {

 geolocation Geolocation,

 maxTxPowerPerHz REAL, /\* unit [dBm/Hz] \*/

 rxSensitivity REAL, /\* unit [dBm] \*/

 rXAntennaGain AntennaGain,

 tXAntennaGain AntennaGain,

 tXSignalBW Integer, /\* unit [Hz], resolution 1 Hz \*/

 minReqSNR REAL, /\* absolute value \*/

 TolerableInterferenceLevel REAL, /\* unit [dBm] \*/

}

Geolocation ::= SEQUENCE {

 latitude Coordinate,

 longitude Coordinate,

 altitude Altitude,

 antloc AntennaLocationType

}

Coordinate ::= SEQUENCE {

 uncertainty REAL, /\* unit [deg, min,s] \*/

 coordinate REAL /\* unit [deg,min,s] \*/

}

Altitude ::= SEQUENCE {

 uncertainty REAL, /\* unit [m] \*/

 coordinate REAL, /\* unit [m] \*/

 type AltitudeType

}

AltitudeType ::= ENUMERATED {

 meters,

}

AntennaLocationType ::= ENUMERATED {

 indoors,

 outdoors,

 unknown

}

AntennaGain ::= SEQUENCE OF SEQUENCE {

 freq\_ant INTEGER, /\* unit [Hz], resolution 1 Hz \*/

 antParams AntennaParameters

}

AntennaParameters ::= SEQUENCE OF SEQUENCE {

 theta REAL, /\* unit [deg] \*/

 phi REAL, /\* unit [deg] \*/

 gain REAL /\* absolute value \*/

 }

ListOfSupportedFrequencies ::= SEQUENCE OF SEQUENCE {

 *-- The frequency borders of each possible sub band or channel*

 supportedRXFrequency FrequencyRange,

 supportedTXFrequency FrequencyRange,

 *-- Extra channel configuration (subchannelization or channel aggregation)*

 *--supported or not*

 extrachannelizationIsSupported BOOLEAN,

 *-- Extra channel configuration description*

 rXextrachannelizationDescription ExtraChannelizationDescription OPTIONAL,

 tXextrachannelizationDescription ExtraChannelizationDescription OPTIONAL,

*}*

extraChannelizationDescription ::= SEQUENCE{

 *-- Maximum number of non-contiguous channels supported in channel*

 *--aggregation*

 maxNuNonconCH INTEGER,

 *-- Maximum number of contiguous channels supported in channel bonding*

 maxNuConCH INTEGER,

 *--Min channel raster for fine tuning of frequency*

 minChRasterINTEGER *OPTIONAL,* /\* unit [Hz], resolution 1 Hz \*/

 *-- Maximum supported bandwidth per channel*

 maxCHBW INTEGER, /\* unit [Hz], resolution 1 Hz \*/

 *-- Minimum supported bandwidth per channel*

 minCHBW INTEGER, /\* unit [Hz], resolution 1 Hz \*/

 *-- Resolution for additional channel bandwidth between minCHBW and*

 *--maxCHBW*

 resolutionSBW INTEGER, /\* unit [Hz], resolution 1 Hz \*/

 *-- Minimum bandwidth within either maxCHBW or minCHBW. Any number or*

 *--location, which fits within either maxCHBW or minCHBW is allowed.*

 minUnderlayBW INTEGER, /\* unit [Hz], resolution 1 Hz \*/

 *-- Offset of the start frequency in the case of maxCHBW*

 offsetFreqMaxCHBW INTEGER, /\* unit [Hz], resolution 1 Hz \*/

 *-- Offset of the start frequency in the case of minCHBW*

 offsetFreqMinCHBW INTEGER, /\* unit [Hz], resolution 1 Hz \*/

 *-- Offset always based on the Primary Channelization or not*

 OffsetPerPrimaryChannelization BOOLEAN

}

ListOfOperatingFrequencies ::= SEQUENCE OF SEQUENCE {

 operFreq FrequencyRange,

 occupancy REAL,

 totalOccupancy REAL OPTIONAL

}

RequiredResource ::= SEQUENCE OF SEQUENCE {

 requiredBandwidth INTEGER, /\* unit [Hz], resolution 1 Hz \*/

 requiredOccupancyShare REAL /\* absolute value \*/

}

MeasurementCapability ::= ENUMERATED {

 sinr,

 fer,

 ipnf,

 signalDistribution,

 spectrum,

 ownNetworkChannelLoad,

 totalChannelLoad,

 otherUsers

}

InterferenceDirection ::= ENUMERATED {

 mutual,

 source,

 victim

}

InterfererData ::= SEQUENCE OF SEQUENCE {

 estimationFrequency INTEGER, /\* unit [Hz], resolution 1 Hz \*/

 interfererBW INTEGER, /\* unit [Hz], resolution 1 Hz \*/

 -- at master/device location of interfered part

 Imdata InterferenceValues,

 -- at the worst location of interfered part

 Iwdata InterferenceValues

 }

InterferenceValues ::= SEQUENCE {

 -- power spectral density at interfered location

 pSD REAL, /\* unit [dBm/Hz] \*/

 -- Direction of arrival

 dOA REAL /\* unit [deg], (N=0, W=90, S=180, E=270) \*/

 }

RadioEnvironmentInformation ::= SEQUENCE OF SEQUENCE {

 --Identifier of the CE for which information is given

 ceID CX\_ID,

 CHOICE {

 --information of the own network

 ownNet OwnNetwork,

 --802.19.1 type networks

 coexNets CoexSetNetworks,

 --Other networks outside 802.19.1

 otherNets OtherNetworks,

 --unused frequency locations

 unusedFreqs UnusedFrequencies,

 --Unknown frequency usage

 unknownUsage UnknownFrequencyUsage

 }

}

OwnNetwork ::= SEQUENCE {

 -- CM identifier for a source CE which CSEs are reported

 cMID CxID,

 -- CM IP address

 cMIPAddress OCTET STRING,

 -- CM port number

 cMPortNum OCTET STRING,

 -- Network technology, e.g., 802.11af, 802.22

 networkTechnology NetworkTechnology,

 -- Coexistence service subscription

 subscribedServ SubscribedService,

 -- List of supported resources: channel numbers or frequencies

 -- List of allowed channels for the source CE

 listOfAllowedTVWSChNum ListOfAllowedTVWSChNumber,

 --Frequencies supported by the source CE

 listOfSupportedResources CHOICE {

 -- List of supported channel numbers

 listOfSuppChNumber ListOfSupportedChNumber,

 -- List of supported frequencies

 listOfSuppFreq ListOfSupportedFrequencies

 },

 -- Indicates whether scheduled transmission is supported

 txScheduleSupported BOOLEAN,

 -- Indicates whether the coexistence set element releases resources

 --Coexistence value of the network

 coexistenceValue REAL, /\* absolute value \*/

 operResources ListOfOperatingResourceValues

}

ListOfOperatingResourceValue ::= SEQUENCE {

 CHOICE {

 -- Operating channel number

 operChNumber INTEGER,

 -- Operating frequency location

 operFreq FrequencyRange },

 occupancy REAL OPTIONAL, /\* absolute value \*/

 txSch TxSchedule OPTIONAL,

 totalOccupancy REAL OPTIONAL, /\* absolute value \*/

 channelIsShared BOOLEAN OPTIONAL,

 txPowerLimit REAL OPTIONAL /\* unit [W] \*/

 }

ListOfOperatingResourceValues ::= SEQUENCE OF listOfOperatingResourceValue

TxSchedule ::= SEQUENCE {

 scheduleStartTime REAL, /\* GeneralizedTime \*/

 scheduleDuration REAL, /\* unit [s] \*/

 numberOfScheduleRepetitions INTEGER,

 transmissionStartTime REAL, /\* GeneralizedTime \*/

 transmissionDuration REAL /\* unit [s] \*/

}

CoexSetNetworks ::= SEQUENCE OF SEQUENCE {

 -- CE identifier of the related WSO

 ceID CxID,

 -- CM identifier for CE

 cMID CxID,

 -- CM IP address

 cMIPAddress OCTET STRING,

 -- CM port number

 cMPortNum OCTET STRING,

 -- Network identifier, e.g., BSS ID

 networkID OCTET STRING,

 -- Network technology, e.g., 802.11af, 802.22

 networkTechnology NetworkTechnology,

 operResources listOfOperatingResourceValues

}

OtherNetworks ::= SEQUENCE OF SEQUENCE {

 -- Network identifier, e.g., BSS ID

 networkID OCTET STRING OPTIONAL,

 -- Network technology, e.g., 802.11af, 802.22

 networkTechnology NetworkTechnology OPTIONAL,

 CHOICE {

 -- Operating channel number

 operChNumber INTEGER, /\* channel number of incumbent system \*/

 -- Operating frequency location

 operFreq FrequencyRange },

 occupancy REAL OPTIONAL /\* absolute value \*/

}

UnusedFrequencies ::= SEQUENCE OF SEQUENCE {

 CHOICE {

 -- Operating channel number

 operChNumber INTEGER,

 -- Operating frequency location

 operFreq FrequencyRange }

}

UnknownFrequencyUsage ::= SEQUENCE OF SEQUENCE {

 CHOICE {

 -- Operating channel number

 operChNumber INTEGER,

 -- Operating frequency location

 operFreq FrequencyRange }

}

ReqInfoDescr ::= SEQUENCE OF ENUMERATED{

 sinr,

 desiredBandwidth,

 desiredOccupancy,

 desiredCoverage,

 channelNumber,

 subscribedService,

 interferenceLevel,

 coexistenceValue,

 threshold,

 …

}

ReqInfoValue ::= SEQUENCE OF SEQUENCE{

 reqInfoDescr ReqInfoDescr,

 reqInfoValue CHOICE {

 sinrValue REAL, /\* absolute value \*/

 desiredBandwidthValue INTEGER, /\* unit [Hz], resolution 1 Hz \*/

 desiredOccupancyValue REAL, /\* absolute value \*/

 desiredCoverageValue REAL, /\* unit [m] \*/
 channelNumberValue INTEGER,

 subscribedService SubscribedService

 interferenceLevelValue REAL, /\* unit [dBm] \*/

 coexistenceValue REAL, /\* absolute value \*/

 thresholdValue REAL /\* ? \*/

 }

}

MeasurementDescription ::= SEQUENCE {

 measType MeasurementType,

 measSchedule MeasurementSchedule,

 measFreq MeasurementFreq

}

MeasurementType ::= ENUMERATED {

 sinr,

 fer,

 ipnf,

 signalDistribution,

 spectrum,

 ownNetworkChannelLoad,

 totalChannelLoad,

 otherUsers

}

MeasurementSchedule ::= SEQUENCE {

 measStartTime REAL, /\* GeneralizedTime \*/

 numberOfMeasurements INTEGER,

 timeBetweenMeasurements REAL /\* unit [s] \*/

}

MeasurementFreq ::= CHOICE {

 startStopFreq FrequencyIndication,

 channelNumber ChannelIndication

}

FrequencyIndication ::= SEQUENCE OF FrequencyRange

ChannelIndication ::= SEQUENCE OF INTEGER

MeasurementReport ::= CHOICE {

 sinrReport SinrReport,

 ferReport FerReport,

 ipnfReport IpnfReport,

 signalDistributionReport SignalDistributionReport,

 spectrumReport SpectrumReport,

 ownNetworkChannelLoadReport ChannelLoadReport,

 totalChannelLoadReport ChannelLoadReport,

 otherUsersReport OtherUsersReport

}

SinrReport ::= SEQUENCE {

 measBW INTEGER, /\* unit [Hz], resolution 1 Hz \*/

 sinr REAL /\* absolute value \*/

}

FerReport ::= SEQUENCE {

 fer REAL /\* absolute value \*/

}

IpnfReport ::= SEQUENCE {

 measBW INTEGER, /\* unit [Hz], resolution 1 Hz \*/

 ipnf REAL

}

SignalDistributionReport ::= SEQUENCE {

 measBW INTEGER, /\* unit [Hz], resolution 1 Hz \*/

 lowEndOfSignalLevelRange INTEGER, /\* unit [dBm] \*/

 numberOfSignalLevelRange INTEGER,

 widthOfSignalLevelRange INTEGER, /\* unit [dB] \*/

 signalProportionPerRange SEQUENCE OF REAL

}

SpectrumReport ::= SEQUENCE {

 measBW INTEGER, /\* unit [Hz], resolution 1 Hz \*/

 lowEndOfReportedBandwidth INTEGER, /\* unit [Hz], resolution 1 Hz \*/

 numberOfSubchannels INTEGER,

 widthOfSubchannel INTEGER, /\* unit [Hz], resolution 1 Hz \*/

 signalStrengthPerSubchannel SEQUENCE OF REAL

}

ChannelLoadReport ::= SEQUENCE {

 load REAL

}

OtherUsersReport ::= SEQUENCE {

 technologyType NetworkTechnology

}

ReconfigurationParameters ::= SEQUENCE OF SEQUENCE {

 operResources ListOfOperatingResourceValue

}

FailedParameter ::= SEQUENCE {

 failedParameterID FailedParameterID,

 failedParameterValue FailedParameterValue

}

FailedParameters ::= SEQUENCE OF FailedParameter

FailedParameterID ::= ENUMERATED {

 operatingFrequency,

 listOfoperatingChNumber,

 txPowerLimit,

 channelIsShared,

 txSchedule

}

FailedParameterValue ::= CHOICE {

 operatingFrequency SEQUENCE {startFreq REAL, stopFreq REAL},

 listOfoperatingChNumber SEQUENCE OF INTEGER,

 txPowerLimit REAL, /\* unit [W] \*/

 channelIsShared BOOLEAN,

 txSchedule SEQUENCE OF TxSchedule OPTIONAL

}

EventParams ::= SEQUENCE {

 eventDescr EventDescr,

 addInfo AddInfo OPTIONAL

}

EventDescr ::= ENUMERATED {

 sinrThresholdReached,

 qosDegradation,

 temporaryResourceRelease,

 temporaryResourceReclaim,

 …

}

AddInfo ::= CHOICE {

 operatingParameters ListOfOperatingResourceValues,

 …

}

EntityStatus ::= ENUMERATED {

 alive,

 terminating,

 outOfService

}

END