IEEE P802.19
Wireless Coexistence

|  |
| --- |
| Annex A Data types |
| Date: 2013-07-16 |
| Author(s): |
| Name | Company | Address | Phone | email |
| Stanislav Filin | NICT |  |  | sfilin@nict.go.jp |
| Hiroshi Harada | NICT |  |  |  |
| Hyunduk Kang | ETRI |  |  |  |

Abstract

This document is a submission to IEEE 802.19 TG1 proposing update to Annex A 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.

# Proposed update

*It is proposed to substitute Annex A Data types with the text below.*

IEEE802191DataType DEFINITIONS AUTOMATIC TAGS ::= BEGIN

**-----------------------------------------------------------**

**--Exported data types**

**-----------------------------------------------------------**

--Exported data types

EXPORTS

 --Coexistence protocol entity ID

 CxID,

 --Status

 Status,

 --CxMedia status

 CxMediaStatus,

 --Coexistence service

 CoexistenceService,

 --Subscribed service

 SubscribedService,

 --Network technology

 NetworkTechnology,

 --Network type

 NetworkType,

 --Discovery information

 DiscoveryInformation,

 --Geolocation

 Geolocation,

 --Coverage area

 CoverageArea,

 --Installation parameters

 InstallationParameters,

 --Frequency range

 FrequencyRange,

 --List of available frequencies

 ListOfAvailableFrequencies,

 --List of operating frequencies

 ListOfOperatingFrequencies,

 --List of available channel numbers

 ListOfAvailableChNumbers,

 --List of operating channel numbers

 ListOfOperatingChNumbers,

 --List of supported frequencies

ListOfSupportedFrequencies,

 --Required resource RequiredResource,

 --Operation code for registration

 OperationCode,

 --Measurement capability

 MeasurementCapability,

 --List of available frequencies of the subject WSO

 ListOfSubjectWSOAvailableFrequencies,

 --Transmission schedule

 TxSchedule,

 --CM registration

 CMRegistration,

 --CE registration

 CERegistration,

 --Coexistence report

 CoexistenceReport,

 --Channel priority

 ChannelPriority,

 --List of subject CEs

 ListOfSubjectCEs,

 --List of neighbor CMs transport information

 ListOfNeighborCMsTransport,

 --List of neighbor CM

 ListOfNeighborCM,

 --List of neighbor CM WSOs

 ListOfNeighborCMWSOs,

 --List of CEs for reconfiguration

 ReconfigListOfCEs,

 --Channel classification information

 ChClassInfo,

 --Channel classification information list

 ChClassInfoList,

 --Failed parameters

 FailedParameters,

 --Event parameters

 EventParams,

 --Required information description

 ReqInfoDescr,

 -- Requested information value

 ReqInfoValue,

 -- Negotiation status

 NegotiationStatus,

 -- Negotiation information

 NegotiationInformation

 -- Winner CM ID list

 ListOfWinnerCMID,

 -- Slot time position list

 ListOfSlotTimePosition

 --Measurement description

 MeasurementDescription,

 --Measurement result

 MeasurementResult,

 --Mobility Information

 MobilityInformation;

**-----------------------------------------------------------**

**--Coexistence protocol entity ID**

**-----------------------------------------------------------**

--Coexistence protocol entity type

CxType ::= ENUMERATED {

 --Coexistence enabler

 ce,

 --Coexistence manager

 cm,

 --Coexistence discovery and information server

 cdis}

--Coexistence protocol entity ID

CxID ::= SEQUENCE {

 --Entity type

 type CxType,

 --Entity ID

 id OCTET STRING}

**-----------------------------------------------------------**

**--Status**

**-----------------------------------------------------------**

--Status

Status ::= ENUMERATED {

 --Primitive/message is successfully processed

 noError,

 --Primitive/message is rejected due to security reasons

 rejected,

 --Primitive/message cannot be successfully processed because according to the current entity status different primitive/message is expected

 invalidEntityStatus,

 --Primitive/message cannot be successfully processed because of invalid values of parameters

 invalidArgument,

 --Primitive/message cannot be successfully processed because of the process error in the receiving entity

 processFailure,

 --Primitive/message cannot be successfully processed because of the connection error

 networkFailure}

--CxMedia status

CxMediaStatus ::= ENUMERATED {

 noErrorAccepted,

 noErrorRejected,

 errorInvalidEntityStatus,

 errorInvalidArgument,

 errorProcessFailure,

 errorNetworkFailure,

 errorUnknown

}

**-----------------------------------------------------------**

**--Coexistence service**

**-----------------------------------------------------------**

--Coexistence service

CoexistenceService ::= ENUMERATED {

 --Information service

 information,

 --Management service

 management,

 --No service

 noService}

SubscribedService::= ENUMERATED {

 information,

 management,

 interCMCoexistenceSetElementsNeighbors,

 allCoexistenceSetElementsNeighbors

}

**-----------------------------------------------------------**

**--Network technology**

**-----------------------------------------------------------**

NetworkTechnology ::= ENUMERATED {

 --IEEE 802.11af

 ieee802dot11af,

 --IEEE 802.22

 ieee802dot22,

 --Radio microphone

 radioMic,

 --Area broadcast

 areaBroadcast,

 --ECMA 392

 ecma392}

**-----------------------------------------------------------**

**--Network type**

**-----------------------------------------------------------**

NetworkType ::= ENUMERATED {

 fixed,

 mode1,

 mode2,

 …

}

**-----------------------------------------------------------**

**--Discovery information**

**-----------------------------------------------------------**

DiscoveryInformation ::= SEQUENCE {

 coordinateX REAL,

 coordinateY REAL,

 coordinateZ REAL,

 maxTxPower REAL,

 rxSensitivity REAL,

 antennaGain REAL,

 minReqSNR REAL,

 antennaHeight REAL,

 …

}

**-----------------------------------------------------------**

**--Location**

**-----------------------------------------------------------**

--Location

Geolocation ::= CHOICE {

 --Place name or ID

 placeID OCTET STRING,

 --Coordinates of the master station

 coordinates SEQUENCE {

 --Latitude

 latitude REAL,

 --Longitude

 longitude REAL,

 --Altitude

 altitude REAL OPTIONAL}}

**-----------------------------------------------------------**

**--Coverage area**

**-----------------------------------------------------------**

--Coverage area

CoverageArea ::= SEQUENCE {

 --Coverage radius

 radius REAL,

 --Reference central frequency

 refFrequency REAL,

 --Reference height of master station

 refMasterHeight REAL,

 --Reference height of slave station

 refSlaveHeight REAL,

 --Reference transmission power

 refTxPower REAL}

**-----------------------------------------------------------**

**--Installation parameters**

**-----------------------------------------------------------**

--Installation parameters

InstallationParameters ::= SEQUENCE {

 --Operating height of master station

 opMasterHeight REAL OPTIONAL,

 --Operating height of slave station

 opSlaveHeight REAL OPTIONAL,

 --Operating transmission power

 opTxPower REAL OPTIONAL}

**-----------------------------------------------------------**

**--Frequency range related data types**

**-----------------------------------------------------------**

--Frequency range

FrequencyRange ::= SEQUENCE {

 --Start frequency

 startFreq REAL,

 --Stop frequency

 stopFreq REAL}

--List of available frequencies

ListOfAvailableFrequencies ::= SEQUENCE OF SEQUENCE {

 --Frequency range

 frequencyRange FrequencyRange,

 --Transmission power limit

 txPowerLimit REAL OPTIONAL,

 --Start time when this frequency range is available

 availableStartTime GeneralizedTime OPTIONAL,

 --Duration during which this frequency range is available

 availableDuration REAL OPTIONAL}

--List of supported frequencies

ListOfSupportedFrequencies ::= SEQUENCE OF SEQUENCE {

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

 supportedFrequency FrequencyRange,

 -- Extra channel configuration (subchannelization or channel aggregation) supported or not

 extrachannelizationIsSupported BOOLEAN,

 -- Extra channel configuration description

 extrachannelizationDescription 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

minChRaster REAL OPTIONAL,

 -- Maximum supported bandwidth per channel

 maxCHBW REAL,

 -- Minimum supported bandwidth per channel

 minCHBW REAL,

 -- Resolution for additional channel bandwidth between minCHBW and maxCHBW

 resolutionSBW REAL,

 -- Minimum bandwidth within either maxCHBW or minCHBW. Any number or location,

 -- which fits within either maxCHBW or minCHBW is allowed.

 minUnderlayBW REAL,

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

offsetFreqMaxCHBW REAL,

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

offsetFreqMinCHBW REAL,

 -- Offset always based on the Primary Channelization or not

 OffsetPerPrimaryChannelization BOOLEAN

}

--List of operating frequencies

ListOfOperatingFrequencies ::= SEQUENCE OF SEQUENCE {

 --Frequency range

 frequencyRange FrequencyRange,

 --Occupancy if known

 occupancy REAL OPTIONAL}

**-----------------------------------------------------------**

**--Available channel numbers**

**-----------------------------------------------------------**

ConstOfChUseID :: = ENUMERATED {

 regulationMaxTxPower,

 regulationMaxAntGain,

 regulationMaxAntHeight,

 regulationTVDBUpdateTime,

 outOfBandEmissionLimit,

 …

}

ConstOfChUseValue ::= CHOICE {

 regulationMaxTxPower REAL,

 regulationMaxAntMaxGain REAL,

 regulationAntMaxHeight REAL,

 regulationTVDBUpdateTime REAL,

 outOfBandEmissionLimit REAL,

 …

}

ConstOfChUses ::= SEQUENCE OF SEQUENCE {

 constOfChUseID ConstOfChUseID,

 constOfChUseValue ConstOfChUseValue

}

ListOfAvailableChNumbers ::= SEQUENCE OF SEQUENCE {

 chNumber INTEGER,

 availableStartTime GeneralizedTime,

 availableDuration REAL,

 constOfChUses ConstOfChUses

}

**-----------------------------------------------------------**

**--Operating channel numbers**

**-----------------------------------------------------------**

ListOfOperatingChNumbers ::= SEQUENCE OF SEQUENCE {

 chNumber INTEGER,

 occupancy REAL

}

**-----------------------------------------------------------**

**--Required resource**

**-----------------------------------------------------------**

--Required resource

RequiredResource ::= SEQUENCE OF SEQUENCE {

 --Required bandwidth

 requiredBandwidth REAL,

 --Expected occupancy if known

 occupancy REAL OPTIONAL}

**-----------------------------------------------------------**

**--Operation code for registration**

**-----------------------------------------------------------**

--Operation code for registration

OperationCode ::= ENUMERATED {

 --New registration

 new,

 --Update of registration information

 update,

 --Deregistration

 delete}

**-----------------------------------------------------------**

**--Measurement capability**

**-----------------------------------------------------------**

MeasurementCapability ::= ENUMERATED {

 energyDetection,

 featureDetection,

 …

}

**-----------------------------------------------------------**

**--Reconfiguration related data types**

**-----------------------------------------------------------**

--Transmission schedule

TxSchedule ::= SEQUENCE {

 --Schedule start time

 scheduleStartTime GeneralizedTime,

 --Schedule frame duration

 scheduleFrameDuration REAL,

 --Number of schedule frames

 numberOfFrames INTEGER,

 --Transmission start time within a schedule frame

 transmissionStartTime REAL,

 --Transmission duration within a schedule frame

 transmissionDuration REAL}

**-----------------------------------------------------------**

**--CM registration**

**-----------------------------------------------------------**

--CM registration

CMRegistration ::= SEQUENCE {

 --CM IP address

 ipAddress OCTET STRING,

 --CM port number

 portNumber INTEGER}

--List of WSO for registration

ListOfWSORegistrations ::= SEQUENCE OF SEQUENCE {

 --New registration, registration update or deregistration

 operationCode OperationCode,

 --WSO ID

 wsoID OCTET STRING,

 --Network technology

 networkTechnology NetworkTechnology OPTIONAL,

 --Location

 geolocation Geolocation OPTIONAL,

 --Coverage area

 coverageArea CoverageArea OPTIONAL,

 -- Mobility information

 mobilityInformation MobilityInformation OPTIONAL,

 --Installation parameters

 installationParameters InstallationParameters OPTIONAL,

 --List of available frequencies

 listOfAvailableFrequencies ListOfAvailableFrequencies OPTIONAL}

--CE registration

CERegistration ::= SEQUENCE OF SEQUENCE{

 --CE ID

 ceID CxID,

 -- List of WSO registration

 listOfWSORegistration ListOfWSORegistrations}

**-----------------------------------------------------------**

**--Coexistence report**

**-----------------------------------------------------------**

CoexistenceReport ::= SEQUENCE OF SEQUENCE {

 networkID OCTET STRING,

 networkTechnology NetworkTechnology,

 listOfOperatingChNumbers ListOfOperatingChNumbers

}

ChannelPriority ::= SEQUENCE OF SEQUENCE {

 channelNumber INTEGER,

 priority INTEGER

}

**-----------------------------------------------------------**

**--Coexistence set information related data types**

**-----------------------------------------------------------**

--Interference direction

InterferenceDirection ::= ENUMERATED {

 --Subject WSO creates interference to neighbor WSO

 source,

 --Neighbor WSO creates interference to subject WSO

 victim,

 --Both subject WSO and neighbor WSO create interference to each other

 mutual}

--List of neighbor WSOs

ListOfNeighborWSOs ::= SEQUENCE OF SEQUENCE {

 --Neighbor WSO ID

 wsoID OCTET STRING,

 --Neighbor WSO network technology

 networkTechnology NetworkTechnology,

 --Interference direction

 interferenceDirection InterferenceDirection,

 --Distance to subject WSO

 distance REAL

 --List of operating frequencies

 --Not used in CoexistenceSetInformationAnnouncement

 --Used in CoexistenceReportAnnouncement and CxMediaCoexistenceReportIndication

 listOfOperatingFrequencies ListOfOperatingFrequencies OPTIONAL}

--List of neighbor CEs

ListOfNeighborCEs ::= SEQUENCE OF SEQUENCE {

 --Neihgbor CE ID

 ceID CxID,

 --List of neighbor WSOs

 listOfNeighborWSOs ListOfNeighborWSOs}

--List of neighbor CMs

ListOfNeighborCMs ::= SEQUENCE OF SEQUENCE {

 --Neighbor CM ID

 cmID CxID,

 --List of neighbor CEs

 listOfNeighborCEs ListOfNeighborCEs}

--List of available frequencies of the subject WSO

ListOfSubjectWSOAvailableFrequencies ::= SEQUENCE OF SEQUENCE {

 --Frequency range

 frequencyRange FrequencyRange,

 --List of neighbor CMs

 listOfNeighborCMs ListOfNeighborCMs}

--List of subject WSOs

ListOfSubjectWSOs ::= SEQUENCE OF SEQUENCE {

 --Subject WSO ID

 wsoID OCTET STRING,

 --List of available frequencies of the subject WSO

 listOfSubjectWSOAvailableFrequencies ListOfSubjectWSOAvailableFrequencies}

--List of subject CEs

ListOfSubjectCEs ::= SEQUENCE OF SEQUENCE {

 --Subject CE ID

 ceID CxID,

 --List of subject WSOs

 listOfSubjectWSOs ListOfSubjectWSOs}

--List of neighbor CMs transport information

ListOfNeighborCMsTransport ::= SEQUENCE OF SEQUENCE {

 --Neighbor CM ID

 cmID CxID,

 -- Neighbor CM IP address

 ipAddress OCTET STRING,

 -- Neighbor CM port number

 portNumber INTEGER}

**-----------------------------------------------------------**

**--Coexistence set information**

**-----------------------------------------------------------**

ListOfCoexSetElement ::= SEQUENCE OF SEQUENCE {

 networkID OCTET STRING,

 networkTechnology NetworkTechnology

}

ListOfNeighborCM ::= SEQUENCE OF SEQUENCE {

 neighborCMID CxID,

 listOfCoexSetElement ListOfCoexSetElement

}

**-----------------------------------------------------------**

**--Coexistence set element information related data types**

**-----------------------------------------------------------**

--List of neighbor CM WSOs

ListOfNeighborCMWSOs ::= SEQUENCE OF SEQUENCE {

 --WSO ID

 wsoID OCTET STRING,

 --List of available frequencies

 listOfAvailableFrequencies ListOfAvailableFrequencies OPTIONAL,

 --List of operating frequencies

 listOfOperatingFrequencies ListOfOperatingFrequencies OPTIONAL

}

**-----------------------------------------------------------**

**--Coexistence set element reconfiguration related data types**

**-----------------------------------------------------------**

--List of WSOs

ReconfigListOfWSOs ::= SEQUENCE OF SEQUENCE {

 --WSO ID

 wsoID OCTET STRING,

 --Potential new operating frequency

 newOperatingFrequency FrequencyRange

}

--List of CEs

ReconfigListOfCEs ::= SEQUENCE OF SEQUENCE {

 --CE ID

 ceID CxID,

 --List of WSOs

 reconfigListOfWSOs ReconfigListOfWSOs

}

**-----------------------------------------------------------**

**--Channel classification**

**-----------------------------------------------------------**

OperatingChannelInfo ::= SEQUENCE {

 operatingChannelNumber INTEGER,

 listOfNetworkID SEQUENCE OF OCTET STRING,

 …

}

ChClassInfo ::= SEQUENCE {

 availableChannelList SEQUENCE OF INTEGER,

 restrictedChannelList SEQUENCE OF INTEGER,

 protectedChannelList SEQUENCE OF INTEGER,

 unclassifiedChannelList SEQUENCE OF INTEGER,

 operatingChannelList SEQUENCE OF OperatingChannelInfo,

 coexistenceChannelList SEQUENCE OF OperatingChannelInfo,

 …

}

ChClassInfoList ::= SEQUENCE OF SEQUENCE {

 networkID OCTET STRING,

 chClassInfo ChClassInfo

}

**-----------------------------------------------------------**

**--Failed parameters**

**-----------------------------------------------------------**

FailedParameterID ::= ENUMERATED {

 listOfoperatingChNumber,

 txPowerLimit,

 channelIsShared,

 txSchedule

}

FailedParameterValue ::= CHOICE {

 listOfoperatingChNumber SEQUENCE OF INTEGER,

 txPowerLimit REAL,

 channelIsShared BOOLEAN,

 txSchedule TxSchedule

}

FailedParameters ::= SEQUENCE OF SEQUENCE {

 failedParameterID FailedParameterID,

 failedParameterValue FailedParameterValue

}

**-----------------------------------------------------------**

**--Event indication**

**-----------------------------------------------------------**

EventDescr ::= ENUMERATED{

 sinrThresholdReached,

 qosDegradation,

 …

}

EventParams ::= SEQUENCE {

 eventDescr EventDescr

}

**-----------------------------------------------------------**

**--Information Acquiring**

**-----------------------------------------------------------**

ReqInfoDescr ::= SEQUENCE OF ENUMERATED {

 sinr,

 desiredBandwidth,

 desiredOccupancy,

 desiredQoS,

 desiredCoverage,

 channelNumber,

 subscribedService,

 interferenceLevel,

 fairness,

 threshold,

 mobilityInformation,

 ...

}

ReqInfoValue ::= SEQUENCE OF SEQUENCE {

 reqInfoDescr ReqInfoDescr,

 reqInfoValue CHOICE {

 sinrValue REAL,

 desiredBandwidthValue REAL,

 desiredOccupancyValue REAL,

 desiredQoSValue REAL,

 desiredCoverageValue REAL,

 channelNumberValue REAL,

 subscribedServiceValue SubscribedService

 interferenceLevelValue REAL,

 fairnessValue REAL,

 thresholdValue REAL,

mobilityInformation MobilityInformation,

 otherValue ANY

 }

}

**-----------------------------------------------------------**

**--Negotiation**

**-----------------------------------------------------------**

NegotiationStatus :: = SEQUENCE {

 negotiationSuccess BOOLEAN,

 negotiationFailure BOOLEAN,

 underNegotiation BOOLEAN,

 …

}

StartEndTime :: = SEQUENCE {

 startTime REAL,

 endTime REAL

}

TimeSharingUnitInfo ::= SEQUENCE {

 referenceTime REAL,

 windowTime StartEndTime,

 slotTime StartEndTime,

 …

}

NegotiationInformation :: = SEQUENCE {

 mode BOOLEAN,

 listOfChNumber SEQUENCE OF INTEGER,

 timeSharingUnitInfo TimeSharingUnitInfo,

 slotTimePosition StartEndTime,

 numberOfSlots INTEGER,

 disallowedSlotTimePosition StartEndTime,

 listOfContentionNumbers SEQUENCE OF REAL,

 …

}

ListOfWinnerCMID ::= SEQUENCE OF CxID

ListOfSlotTimePosition ::= SEQUENCE OF REAL

**-----------------------------------------------------------**

**--Measurement**

**-----------------------------------------------------------**

MeasurementSchedule ::= SEQUENCE {

 measStartTime REAL,

 numberOfMeasurements INTEGER,

 timeBetweenMeasurements REAL

}

MeasurementFreq ::= SEQUENCE OF INTEGER

MeasurementType ::= ENUMERATED {

 interferenceLevel

}

MeasurementDescription ::= SEQUENCE {

 measType MeasurementType,

 measSchedule MeasurementSchedule,

 measFreq MeasurementFreq

}

MeasurementReport ::= CHOICE {

 interferenceLevelValue REAL,

 …

}

MeasurementResult ::= SEQUENCE OF SEQUENCE {

 measurementDescription MeasurementDescription,

 measurementReport MeasurementReport

}

-----------------------------------------------------------

--Mobility Information

-----------------------------------------------------------

MobilityInformation :: = CHOICE {

maxSpeed REAL, /\*unit [km/h] \*/

speedInformation SpeedInformation,

routeInformation RouteInformation

}

SpeedInformation ::= SEQUENCE {

WSOSpeed REAL, /\* unit [km/h] \*/

WSODirection REAL, /\* unit [radian] \*/

}

RouteInformation ::= SEQUENCE {

 PlannedRoute SEQUENCE of Geolocation,

 PlannedTime SEQUENCE of GeneralizedTime

}

END