IEEE P802.19
Wireless Coexistence

|  |
| --- |
| Proposed resolution to comments 114 and 119 |
| Date: 2011-09-20 |
| Author(s): |
| Name | Company | Address | Phone | email |
| Stanislav Filin | NICT |  |  | sfilin@nict.go.jp |
| Junyi Wang | NICT |  |  |  |
| Hiroshi Harada | NICT |  |  |  |

Abstract

This document is a submission to IEEE 802.19 TG1 proposing resolution to comments 114 and 119.

**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.

# Introduction

Subclause 4.3 provides data type definition for Reference Model primitives. Subclause 5.3 provides data type definition for protocol messages. There is big overlapping between data type definitions in subclauses 4.3 and 5.3. It is proposed to combine these two subclauses and move the result to a normative annex. The next clause proposes text for the annex.

# Proposed annex

*Annex title:*

Data type definition

*Annex text:*

TransportPref ::= ENUMERATED{

 TCP,

 UDP,

 HTTP,

 SNMP,

 …

}

SubscribedService::= ENUMERATED{

 information,

 management,

 interCMNeighbors,

 allNeighbors

}

NetworkID::= ENUMERATED{

 BSSID,

 …

}

NetworkTechnology ::= ENUMERATED{

 IEEE802.11af,

 IEEE802.22,

 ECMA392,

 …

}

NetworkType ::= ENUMERATED{

 fixed,

 mode2,

 …

}

DiscoveryInformation ::= SEQUENCE{

 coordinateX REAL,

 coordinateY REAL,

 coordinateZ REAL,

 maxTxPower REAL,

 rxSensitivity REAL,

 antennaGain REAL,

 minReqSNR REAL,

 TolerableInterferenceLevel REAL,

 antennaHeight REAL,

 …

}

ListOfSupportedChNumber ::= SEQUENCE OF INTEGER

ListOfOperatingChNumber ::= SEQUENCE OF INTEGER

ListOfSupportedFrequencies ::= SEQUENCE OF SEQUENCE{

 startFreq REAL,

 stopFreq REAL

}

ListOfOperatingFrequencies ::= SEQUENCE OF SEQUENCE{

 startFreq REAL,

 stopFreq REAL,

 occupancy REAL,

 totalOccupancy REAL OPTIONAL

}

InterferenceDirection ::= ENUMERATED{mutual, source, victim}

FreqDescription ::= SEQUENCE{

 networkID NetworkID OPTIONAL,

 networkTechnology NetworkTechnology OPTIONAL,

 coexType ENUMERATED{known, unknown},

 interferenceDirection InterferenceDirection,

 occupancy REAL OPTIONAL,

 totalOccupancy REAL OPTIONAL

}

RadioEnvironmentInformation ::= SEQUENCE OF SEQUENCE{

 startFreq REAL,

 stopFreq REAL,

 state ENUMERATED{free, occupiedKnown, occupiedUnknown, notMeasured},

 freqDescription FreqDescription OPTIONAL

}

NetworkGeometryClass ::= CHOICE{Class#1, Class#2, Class#3, Class#4}

DeviceLocation ::= SEQUENCE{

 coordinateX REAL,

 coordinateY REAL,

 coordinateZ REAL

}

RequiredResource ::= SEQUENCE OF SEQUENCE{

 requiredBandwidth REAL,

 expectedLoad REAL

}

ListOfNeighborCEID ::= SEQUENCE OF CX\_ID

NeighborReport ::= SEQUENCE OF SEQUENCE{

 networkID NetworkID,

 networkTechnology NetworkTechnology,

 interferenceDirection InterferenceDirection,

 interferenceLevelFromNeighbor REAL,

 interferenceLevelToNeighbor REAL,

 listOfOperatingChannelNumber SEQUENCE OF INTEGER OPTIONAL,

 listOfOperatingFrequencies ListOfOperatingFrequencies OPTIONAL,

 radioEnvironmentInformation RadioEnvironmentInformation OPTIONAL,

 networkGeometryClass NetworkGeometryClass

}

AggregatedInterferferenceControlParameters :: = SEQUENCE{

 ReferencePointID INTEGER,

 Geolocation ReferencePointGeolocation,

 ACS REAL,

 Antenna height REAL,

 Antenna gain REAL,

 Protection ratio REAL,

 …

}

ReferencePointGeolocation :: = ENUMERATED {

 Latitude REAL,

 Longitude REAL,

 Altitude REAL,

 …

}

AvailableChannelList::= SEQUENCE OF SEQUENCE{

 startFreq REAL,

 stopFreq REAL,

 txPowerLimit REAL,

 aggregatedInterferferenceControlParameters AggregatedInterferferenceControlParameters

}

ListOfAllowedTVWSChNumber ::= SEQUENCE OF INTEGER

ConstOfChUseID :: = ENUMERATED{

 regulationMaxTxPower,

 regulationMaxAntGain,

 regulationMaxAntHeight,

 regulationTVDBUpdateTime,

 OutOfBandEmissionLimit,

 …

}

ConstOfChUseValue :: = CHOICE{

 regulationMaxTxPower REAL,

 regulationMaxAntMaxGain REAL,

 regulationAntMaxHeight REAL,

 regulationTVDBUpdateTime REAL,

 OutOfBandEmissionLimit REAL,

 …

}

ConstOfChUse : : = SEQUENCE{

 constOfChUseID ConstOfChUseID,

 constOfChUseValue ConstOfChUseValue

}

ConstOfChUses : : = SEQUENCE OF ConstOfChUse

OperatingChannelInfo :: = SEQUENCE {

 operatingChannelNumber INTEGER,

 listOfNetworkID SEQUENCE OF NetworkID,

 …

}

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 NetworkID,

 chClassInfo ChClassInfo

}

ReqInfoDescr ::= SEQUENCE OF ENUMERATED{

 SINR,

….desiredBandwidth,

 desiredOccupancy,

 desiredQoS,

 desiredCoverage,

 channelNumber,

 …

}

ReqInfoValue ::= SEQUENCE OF SEQUENCE{

 reqInfoDescr ReqInfoDescr,

 reqInfoValue CHOICE{SINRValue REAL, desiredBandwidthValue REAL,

 desiredOccupancyValue REAL, desiredQoSValue REAL,

 desiredCoverageValue REAL, channelNumberValue REAL,

 otherValue ANY}

}

MeasSchedule ::= SEQUENCE {

 measStartTime REAL,

 numberOfMeasurements INTEGER,

 timeBetweenMeasurements REAL

}

MeasFreq ::= SEQUENCE{

 measStartAFreq REAL OPTIONAL,

 measEndFreq REAL OPTIONAL,

 listOfChNumber SEQUENCE OF INTEGER OPTIONAL

}

MeasurementDescription ::= SEQUENCE OF SEQUENCE{

 measDescr ENUMERATED{SINR, BER, SensingLevel, PrimaryDetection, TVBDDetection,

 ChannelLoadMeasurement, …},

 measSchedule MeasSchedule,

 measFreq MeasFreq

}

MeasurementResult ::= SEQUENCE OF SEQUENCE{

 reqInfoDescr ReqInfoDescr,

 reqInfoValue CHOICE{SINRValue REAL, BERValue REAL,

 SensingLevelValue REAL, PrimaryDetectionValue BOOLEAN,

 TVBDDetectionValue BOOLEAN, ChannelLoadMeasurementValue REAL,

 otherValue ANY}

}

StartEndTime :: = SEQUENCE {

 startTime REAL,

 endTime REAL,

}

NegotiationStatus :: = SEQUENCE {

 negotiationSuccess BOOLEAN,

 negotiationFailure BOOLEAN,

 underNegotiation BOOLEAN,.

 …

}

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 CX\_ID

ListOfSlotTimePosition ::= SEQUENCE OF REAL

TxSchedule ::= SEQUENCE {

 scheduleStartTime REAL,

 scheduleDuration REAL,

 numberOfScheduleRepetitions INTEGER,

 transmissionStartTime REAL,

 transmissionDuration REAL

}

ReconfigurationRequest ::= SEQUENCE OF SEQUENCE {

 operatingFrequency SEQUENCE{startFeq REAL, stopFreq REAL} OPTIONAL,

 listOfOperatingChNumber SEQUENCE OF INTEGER OPTIONAL,

 txPowerLimit REAL OPTIONAL,

 channelIsShared BOOLEAN,

 txSchedule SEQUENCE OF TxSchedule OPTIONAL,

 networkTechnology NetworkTechnology,

}

FailedParameterID : : = ENUMERATED {

 operatingFrequency,

 listOfoperatingChNumber,

 txPowerLimit,

 channelIsShared,

 txSchedule,

}

FailedParameterValue : : = CHOICE{

 operatingFrequency SEQUENCE{startFeq REAL, stopFreq REAL},

 listOfoperatingChNumber SEQUENCE OF INTEGER,

 txPowerLimit REAL,

 channelIsShared BOOLEAN,

 txSchedule SEQUENCE OF TxSchedule OPTIONAL

}

FailedParameter : : = SEQUENCE{

 failedParameterID FailedParameterID,

 failedParameterValue FailedParameterValue

}

FailedParameters : : = SEQUENCE OF FailedParameter

EventDescr ::= ENUMERATED{

 SINRThresholdReached,

 QoSDegradation,

 MisLocatedTVBDDetected,

 …

}

MisLocatedTVBDDetectedInfo ::= SEQUENCE{

 networkID NetworkID,

 listOfoperatingFrequency SEQUENCE OF SEQUENCE{startFeq REAL, stopFreq REAL}

 OPTIONAL,

 listOfChannelNumber SEQUENCE OF INTEGER OPTIONAL

}

AddInfo ::= CHOICE{

 misLocatedTVBDDetectedInfo MisLocatedTVBDDetectedInfo,

 …

}

EventParams ::= SEQUENCE{

 eventDescr EventDescr,

 addInfo AddInfo OPTIONAL

}

GuranteedQoSOfWiredConnection:: = ENUMERATED{

 CHOICE{xDSL, OpticalFibre, Others},

 GuranteedMinimumBitRates,

 GuranteedMaximumLatency OPTIONAL,

 …

}

CX\_ID ::= ENUMERATED{

 CE\_ID,

 CM\_ID,

 CDIS\_ID,

 TVWSDB\_ID

}

OperationCode ::= ENUMERATED{

 New,

 Add,

 Modify,

 Remove

}