IEEE P802.19  
Wireless Coexistence

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Proposed resolution to comment to ASN.1 data type definition for COEX\_MEDIA\_SAP | | | | |
| Date: 2011-11-08 | | | | |
| 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 comment to ASN1 data type definition for COEX\_MEDIA\_SAP.

**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 resolution

*It is proposed to modify the current text in 4.3.2 COEX\_MEDIA\_SAP as shown below:*

IEEE802191MSAPDataType DEFINITIONS AUTOMATIC TAGS::= BEGIN

SubscribedService ::= ENUMERATED{

information,

management

}

NetworkTechnology ::= ENUMERATED{

ieee802dot11af5MHz,

ieee802dot11af10MHz,

ieee802dot11af20MHz,

ieee802dot22

…

}

NetworkType ::= ENUMERATED{

fixed,

mode2,

…

}

DiscoveryInformation ::= SEQUENCE{

latitude REAL,

longitude REAL,

altitude REAL,

maxTxPower REAL,

rxSensitivity REAL,

antennaGain REAL,

minReqSNR REAL,

antennaHeight REAL,

…

}

FrequencyRange ::= SEQUENCE{

startFreq REAL,

stopFreq REAL

}

ListOfSupportedFrequencies ::= SEQUENCE OF FrequencyRange

OperatingFrequency ::= SEQUENCE {

frequencyRange FrequencyRange,

occupancy REAL,

totalOccupancy REAL OPTIONAL

}

ListOfOperatingFrequencies ::= SEQUENCE OF OperatingFrequency

InterferenceDirection ::= ENUMERATED {

mutual,

source,

victim

}

CoexType ::= ENUMERATED {

known,

unknown

}

FreqDescription ::= SEQUENCE{

networkID OCTET STRING OPTIONAL,

networkTechnology NetworkTechnology OPTIONAL,

coexType CoexType,

interferenceDirection InterferenceDirection,

occupancy REAL OPTIONAL,

totalOccupancy REAL OPTIONAL

}

REState ::= ENUMERATED {

free,

occupiedKnown,

occupiedUnknown,

notMeasured

}

REInfoElement ::= SEQUENCE {

frequencyRange FrequencyRange,

state REState,

freqDescription FreqDescription OPTIONAL

}

RadioEnvironmentInformation ::= SEQUENCE OF REInfoElement

NetworkGeometryClass ::= ENUMERATED {

class1,

class2,

class3,

class4

}

NeighborReportElement ::= SEQUENCE {

networkID OCTET STRING,

networkTechnology NetworkTechnology,

interferenceDirection InterferenceDirection,

interferenceLevelFromNeighbor REAL,

interferenceLevelToNeighbor REAL,

listOfOperatingFrequencies ListOfOperatingFrequencies OPTIONAL,

listOfOperatingChannelNumbers SEQUENCE OF INTEGER OPTIONAL,

radioEnvironmentInformation RadioEnvironmentInformation OPTIONAL,

networkGeometryClass NetworkGeometryClass OPTIONAL

}

NeighborReport ::= SEQUENCE OF NeighborReportElement

ReferencePointGeolocation :: = SEQUENCE {

latitude REAL,

longitude REAL,

altitude REAL

}

AggrIntCntrParams ::= SEQUENCE {

referencePointID INTEGER,

geolocation ReferencePointGeolocation,

acs REAL,

antennaHeight REAL,

antennaGain REAL,

protectionRatio REAL,

…

}

AvailableChannelElement ::= SEQUENCE {

frequencyRange FrequencyRange,

txPowerLimit REAL,

availableStartTime GeneralizedTime,

availableDuration REAL,

aggrIntCntrParams AggrIntCntrParams OPTIONAL

}

AvailableChannelList::= SEQUENCE OF AvailableChannelElement

RequiredResourceElement ::= SEQUENCE {

requiredBandwidth REAL,

expectedLoad REAL

}

RequiredResource ::= SEQUENCE OF RequiredResourceElement

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 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

}

ReqInfoDescrElement ::= ENUMERATED{

sinr,

desiredBandwidth,

desiredOccupancy,

desiredQoS,

desiredCoverage,

channelNumber,

...

}

ReqInfoDescr ::= SEQUENCE OF ReqInfoDescrElement

ReqInfoValueType ::= CHOICE {

sinrValue REAL,

desiredBandwidthValue REAL,

desiredOccupancyValue REAL,

desiredQoSValue REAL,

desiredCoverageValue REAL,

channelNumberValue REAL,

otherValue ANY

}

ReqInfoValueElement ::= SEQUENCE {

reqInfoDescr ReqInfoDescr,

reqInfoValue ReqInfoValueType

}

ReqInfoValue ::= SEQUENCE OF ReqInfoValueElement

MeasSchedule ::= SEQUENCE {

measStartTime REAL,

numberOfMeasurements INTEGER,

timeBetweenMeasurements REAL

}

MeasFreq ::= SEQUENCE {

measStartAFreq REAL OPTIONAL,

measEndFreq REAL OPTIONAL,

listOfChNumber SEQUENCE OF INTEGER OPTIONAL

}

MeasDescr ::= ENUMERATED {

sinr,

fer,

sensingLevel,

primaryDetection,

tvdbDetection,

channelLoadMeasurement,

...

}

MeasurementDescription ::= SEQUENCE OF SEQUENCE {

measDescr MeasDescr,

measSchedule MeasSchedule,

measFreq MeasFreq

}

ReqInfoValue ::= CHOICE {

SINRValue REAL,

FERValue REAL,

SensingLevelValue REAL,

PrimaryDetectionValue BOOLEAN,

TVBDDetectionValue BOOLEAN,

ChannelLoadMeasurementValue REAL,

otherValue ANY

}

MeasurementResult ::= SEQUENCE OF SEQUENCE {

reqInfoDescr ReqInfoDescr,

reqInfoValue ReqInfoValue

}

TxScheduleElement ::= SEQUENCE {

scheduleStartTime GeneralizedTime,

scheduleDuration REAL,

numberOfScheduleRepetitions INTEGER,

transmissionStartTime REAL,

transmissionDuration REAL

}

TxSchedule ::= SEQUENCE OF TxScheduleElement

ReconfigurationRequestElement ::= SEQUENCE {

operatingFrequency FrequencyRange OPTIONAL,

listOfOperatingChNumber SEQUENCE OF INTEGER OPTIONAL

txPowerLimit REAL OPTIONAL,

channelIsShared BOOLEAN,

txSchedule TxSchedule OPTIONAL,

networkTechnology NetworkTechnology OPTIONAL

}

ReconfigurationRequest ::= SEQUENCE OF ReconfigurationRequestElement

FailedParameterID ::= ENUMERATED {

operatingFrequency,

listOfoperatingChNumber,

txPowerLimit,

channelIsShared,

txSchedule,

}

FailedParameterValue ::= CHOICE {

operatingFrequency FrequencyRange,

listOfoperatingChNumber SEQUENCE OF INTEGER,

txPowerLimit REAL,

txSchedule TxSchedule

}

FailedParameter ::= SEQUENCE {

failedParameterID FailedParameterID,

failedParameterValue FailedParameterValue

}

FailedParameters ::= SEQUENCE OF FailedParameter

EventDescr ::= ENUMERATED{

sinrThresholdReached,

qosDegradation,

misLocatedTVBDDetected,

…

}

MisLocatedTVBDDetectedInfo ::= SEQUENCE {

networkID OCTET STRING,

listOfoperatingFrequency SEQUENCE OF FrequencyRange OPTIONAL,

listOfChannelNumber SEQUENCE OF INTEGER OPTIONAL

}

AddInfo ::= CHOICE {

misLocatedTVBDDetectedInfo MisLocatedTVBDDetectedInfo,

…

}

EventParams ::= SEQUENCE {

eventDescr EventDescr,

addInfo AddInfo OPTIONAL

}

MediaType ::= ENUMERATED {

xDSL,

opticalFiber,

other

}

GuranteedQoSOfWiredConnection:: = ENUMERATED {

mediaType MediaType,

guranteedMinimumBitRate REAL,

guranteedMaximumLatency REAL OPTIONAL,

…

}

END