IEEE P802.19
Wireless Coexistence

|  |
| --- |
| CM to CM communication |
| Date: 2013-09-17 |
| Author(s): |
| Name | Company | Address | Phone | email |
| Stanislav Filin | NICT |  |  | sfilin@nict.go.jp |
| Hiroshi Harada | NICT |  |  |  |
| Hyunduk Kang | ETRI | 138 Gajeong-Ro, Yuseong-Gu, Daejeon, 305-700, South Korea |  | henry@etri.re.kr |
| Gwangzeen Ko | ETRI |  |  |  |
| Myung-Sun Song | ETRI |  |  |  |
| Jae-Ick Choi | ETRI |  |  |  |

Abstract

This document is a submission to IEEE 802.19 TG1 proposing CM to CM communication to resolve comments CID 3, CID 5, CID 58, and CID 82.

**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 add the text below at the end of the clause 5.2 Procedures.*

* + 1. Inter-CM information exchange procedures

These procedures are used to exchange information between CMs operating according to different profiles.

* + - 1. Providing inter-CM information procedure

A CM shall perform the providing inter-CM information procedure when coexistence set information has been changed for one or several WSOs which the CM serves and which have in their coexistence set WSOs served by one or several other CMs operating according to a different profile. An illustrative example of this procedure is shown in Figure 1. The ***InterCMInformationAnnouncement*** and ***InterCMInformationConfirm*** messages are defined in reference.



Figure 1. Providing inter-CM information

After coexistence set information has changed for one or several WSOs which this CM serves and which have in their coexistence set WSOs served by one or several other CMs operating according to a different profile, the CM shall generate one or several ***InterCMInformationAnnouncement*** messages. Number of the generated ***InterCMInformationAnnouncement*** messages is equal to the number of the CMs operating according to a different profile and serving WSOs that are in the coexistence set of the WSOs which coexistence set information has changed. After the ***InterCMInformationAnnouncement*** messages has been generated, the CM shall send these ***InterCMInformationAnnouncement*** messages to the CMs operating according to a different profile and serving WSOs that are in the coexistence set of the WSOs which coexistence set information has changed. CM operations related to generating and sending a ***InterCMInformationAnnouncement*** message are specified in clause reference.

After the other CM has received the ***InterCMInformationAnnouncement*** message, the other CM shall process the ***InterCMInformationAnnouncement*** message, shall generate a ***InterCMInformationConfirm*** message, and shall send this ***InterCMInformationConfirm*** message to the CM from which it received the ***InterCMInformationAnnouncement*** message. CM operations related to processing a ***InterCMInformationAnnouncement*** message, generating and sending a ***InterCMInformationConfirm*** message are specified in clause reference.

* + - 1. Obtaining inter-CM information procedure

A CM shall perform this procedure to obtain coexistence set information from another CM operating according to a different profile. An illustrative example of this procedure is shown in Figure 2. ***InterCMInformationRequest*** and ***InterCMInformationResponse*** messages are defined in Annex C.



Figure 2. Obtaining inter-CM information

In order to start the procedure a CM shall generate a ***InterCMInformationRequest*** message and send this message to another CM. CM operations related to generating and sending a ***InterCMInformationRequest*** message are specified in clause reference.

After the other CM has received a ***InterCMInformationRequest*** message from the CM, the other CM shall obtain coexistence set element information, shall generate a ***InterCMInformationResponse*** message, and shall send this ***InterCMInformationResponse*** message to the CM. CM operations related to obtaining coexistence set element information and generating and sending a ***InterCMInformationResponse*** message are specified in clause reference.

After the CM has received a ***InterCMInformationResponse*** message from the other CM, the CM shall process the ***InterCMInformationResponse*** message. CM operations related to processing a ***InterCMInformationResponse*** message are specified in clause reference.

*It is proposed to modify Annex C Messages as shown in the text below.*

# (normative) Messages

IEEE802191Message DEFINITIONS AUTOMATIC TAGS ::= BEGIN

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

**--Imported data types**

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

--Imported data types

IMPORTS

 --Coexistence protocol entity ID

 CxID,

 --Status

 Status,

 --CxMedia status

 CxMediaStatus,

 --Coexistence service

 CoexistenceService,

 --Subscribed service

 SubscribedService,

 --Network technology

 NetworkTechnology,

 --Network type

 NetworkType,

 --Location

 Geolocation,

 --Discovery information

 DiscoveryInformation,

 --Coverage area

 CoverageArea,

 --Installation parameters

 InstallationParameters,

 --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,

 --List of available frequencies of the subject WSO

 ListOfSubjectWSOAvailableFrequencies,

 --Transmission schedule

 TxSchedule,

 --Frequency range

 FrequencyRange,

 --CM registration

 CMRegistration,

 --CE registration

 CERegistration,

 --List of subject CEs

 ListOfSubjectCEs,

 --List of neighbor CMs transport information

 ListOfNeighborCMsTransport,

 --List of neighbor CM WSOs

 ListOfNeighborCMWSOs,

 --List of CEs for reconfiguration

 ReconfigListOfCEs

 --Coexistence report

 CoexistenceReport,

 --Channel priority

 ChannelPriority,

 --Measurement capability

 MeasurementCapability,

 --List of neighbor CM

 ListOfNeighborCM,

 --Failed parameters

 FailedParameters,

 --Channel classification information

 ChClassInfo,

 --Required information description

 ReqInfoDescr,

 --Requested information value

 ReqInfoValue,

 --Event parameters

 EventParams,

 --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,

 --Entity profile

 EntityProfile

FROM IEEE802191DataType;

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

**--Message structure, header structure, and payload types**

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

--Message structure

CxMessage ::= SEQUENCE {

 -- Message header

 header CxHeader,

 -- Message payload

 payload CxPayload}

--Header

CxHeader ::= CHOICE {

 --For announcement

 none NULL,

 --For request or single response

 requestID INTEGER (0..2147483647),

 --For multiple responses

 multipleResponse SEQUENCE {

 --Original request ID

 requestID INTEGER (0..2147483647),

 --Response number

 sequenceNumber INTEGER (0..2147483647),

 --True is the response is last

 isLastResponse BOOLEAN}}

--Payload types

CxPayload ::= CHOICE {

 --WSO subscription request

 subscriptionRequest SubscriptionRequest,

 --WSO subscription response

 subscriptionResponse SubscriptionResponse,

 --WSO subscription change request

 subscriptionChangeRequest SubscriptionChangeRequest,

 --WSO subscription change response

 subscriptionChangeResponse SubscriptionChangeResponse,

 --CE registration request

 ceRegistrationRequest CERegistrationRequest,

 --Registration response

 registrationResponse RegistrationResponse,

 --Reconfiguration request

 reconfigurationRequest ReconfigurationRequest,

 --Reconfiguration response

 reconfigurationResponse ReconfigurationResponse,

 --Stop operation announcement

 stopOperationAnnouncement StopOperationAnnouncement,

 --Stop operation confirm

 stopOperationConfirm StopOperationConfirm,

 --Coexistence report announcement

 coexistenceReportAnnouncement CoexistenceReportAnnouncement,

 --Coexistence report confirm

 coexistenceReportConfirm CoexistenceReportConfirm,

 --Coexistence report request

 coexistenceReportRequest CoexistenceReportRequest,

 --Coexistence report response

 coexistenceReportResponse CoexistenceReportResponse,

 --CM registration request

 cmRegistrationRequest CMRegistrationRequest,

 --Coexistence set information announcement

 coexistenceSetInformationAnnouncement CoexistenceSetInformationAnnouncement,

 --Coexistence set information confirm

 coexistenceSetInformationConfirm CoexistenceSetInformationAnnouncement,

 --Coexistence set information request

 coexistenceSetInformationRequest CoexistenceSetInformationRequest,

 --Coexistence set information response

 coexistenceSetInformationResponse CoexistenceSetInformationResponse,

 --Coexistence set element information announcement

 coexistenceSetElementInformationAnnouncement CoexistenceSetElementInformationAnnouncement,

 --Coexistence set element information confirm

 coexistenceSetElementInformationConfirm CoexistenceSetElementInformationConfirm,

 --Coexistence set element information request

 coexistenceSetElementInformationRequest CoexistenceSetElementInformationRequest,

 --Coexistence set element information response

 coexistenceSetElementInformationResponse CoexistenceSetElementInformationResponse,

 --Coexistence set element reconfiguration request

 coexistenceSetElementReconfigurationRequest CoexistenceSetElementReconfigurationRequest,

 --Coexistence set element reconfiguration response

 coexistenceSetElementReconfigurationResponse CoexistenceSetElementReconfigurationResponse,

 --CM Reconfiguration request

 cmReconfigurationRequest CMReconfigurationRequest,

 --CM Reconfiguration response

 cmReconfigurationResponse CMReconfigurationResponse,

 --WSO channel classification request

 channelClassificationRequest ChannelClassificationRequest,

 -- WSO channel classification response

 channelClassificationResponse ChannelClassificationResponse,

 --CM channel classification request

 cmChannelClassificationRequest CMChannelClassificationRequest,

 -- CM channel classification response

 cmChannelClassificationResponse CMChannelClassificationResponse,

 -- WSO channel classification update

 channelClassificationAnnouncement ChannelClassificationAnnouncement,

 --Available channel list request from WSO

 availableChannelsRequest AvailableChannelsRequest,

 -- Available channel list response from WSO

 availableChannelsResponse AvailableChannelsResponse,

 --Information acquiring request

 infoAcquiringRequest InforAcquiringRequest,

 --Information acquiring response

 infoAcquiringResponse InforAcquiringResponse,

 -- Event indication

 eventIndication EventIndication,

 -- Event confirm

 eventConfirm EventConfirm,

 --WSO measurement request

 measurementRequest MeasurementRequest,

 --WSO measurement response

 measurementResponse MeasurementResponse,

 --WSO measurement confirm

 measurementConfirm MeasurementConfirm,

 --Master/Slave CM request

 masterCMRequest MasterCMRequest,

 --Master/Slave CM response

 masterCMResponse MasterCMResponse,

 --Master/Slave CM configuration request

 masterSlaveCMconfigurationRequest MasterSlaveCMconfigurationRequest,

 --Master/Slave CM configuration response

 masterSlaveCMconfigurationResponse MasterSlaveCMconfigurationResponse,

 --Negotiation request

 negotiationRequest NegotiationRequest,

 --Negotiation announcement

 negotiationAnnouncement NegotiationAnnouncement,

 --Deregistration request

 wsoDeregistrationRequest WsoDeregistrationRequest,

 -- Deregistration response

 wsoDeregistrationResponse WsoDeregistrationResponse,

 --Inter-CM information announcement

 interCMInformationAnnouncement InterCMInformationAnnouncement,

 --Inter-CM information confirm

 interCMInformationCOnfirm InterCMInformationConfirm,

 --Inter-CM information request

 interCMInformationRequest InterCMInformationRequest,

 --Inter-CM information response

 interCMInformationResponse InterCMInformationResponse

}

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

**--WSO and CM subscription**

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

--WSO subscription request

SubscriptionRequest ::= SEQUENCE {

 --WSO subscription ID

 clientID IA5String OPTIONAL,

 --WSO subscription password

 clientPassword IA5String OPTIONAL,

 --Coexistence service to which WSO is subscribed

 coexistenceService CoexistenceService OPTIONAL,

 --Coexistence service to which WSO is subscribed

 subscribedService SubscribedService OPTIONAL}

--WSO subscription response

SubscriptionResponse ::= SEQUENCE {

 --CM subscription ID

 serverID IA5String OPTIONAL,

 --CM subscription password

 serverPassword IA5String OPTIONAL,

 --Subscription status

 --noError means subscription is confirmed

 --rejected means subscription is not confirmed

 status Status OPTIONAL,

 --Status

 status CxMediaStatus OPTIONAL}

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

**--WSO subscription change**

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

--Request to change subscription

SubscriptionChangeRequest ::= SEQUENCE {

 --Coexistence service to which WSO is subscribed

 subscribedService SubscribedService}

--Response for subscription change

SubscriptionChangeResponse :: = SEQUENCE {

 --Status

 status CxMediaStatus}

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

**--WSO registration**

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

CERegistrationRequest ::= SEQUENCE OF SEQUENCE {

 --New registration, registration update or deregistration

 operationCode OperationCode,

 --WSO ID

 wsoID OCTET STRING,

 --Network ID

 networkID OCTET STRING OPTIONAL,

 --Network technology

 networkTechnology NetworkTechnology OPTIONAL,

 --Network type

 networkType NetworkType OPTIONAL,

 --Location

 geolocation Geolocation OPTIONAL,

 --Discovery information

 discoveryInformation DiscoveryInformation OPTIONAL,

 --Coverage area

 coverageArea CoverageArea OPTIONAL,

 -- Mobility information

 mobilityInformation MobilityInformation OPTIONAL,

 --Installation parameters

 installationParameters InstallationParameters OPTIONAL,

 --List of available frequencies

 listOfAvailableFrequencies ListOfAvailableFrequencies OPTIONAL,

 --Transmission schedule is supported or not

 txScheduleSupported BOOLEAN OPTIONAL,

 --List of operating frequencies

 listOfOperatingFrequencies ListOfOperatingFrequencies OPTIONAL,

 --List of available channel number

 listOfAvailableChNumbers ListOfAvailableChNumbers OPTIONAL,

 --List of supported channel number

 listOfSupportedChNumbers SEQUENCE OF INTEGER OPTIONAL,

 -- List of supported frequencies

 listOfSuppFrequencies ListOfSupportedFrequencies OPTIONAL,

 --List of operating channel number

 listOfOperatingChNumbers ListOfOperatingChNumbers OPTIONAL,

 --Required resource

 requiredResource RequiredResource OPTIONAL,

 --Measurement capability

 measurementCapability MeasurementCapability OPTIONAL}

--Registration response

RegistrationResponse ::= SEQUENCE {

 --Registration status

 status Status OPTIONAL}

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

**--WSO reconfiguration**

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

--Reconfiguration request

ReconfigurationRequest ::= SEQUENCE OF SEQUENCE {

 --WSO ID

 wsoID OCTET STRING OPTIONAL,

 --Operating frequency

 operatingFrequency FrequencyRange OPTIONAL,

 --List of operating channel number

 listOfOperatingChNumber SEQUENCE OF INTEGER OPTIONAL,

 --Transmission power limit

 txPowerLimit REAL OPTIONAL,

 --Indication whether the channel is shared

 channelIsShared BOOLEAN OPTIONAL,

 --Transmission schedule

 txSchedule TxSchedule OPTIONAL,

 -- Channel classification information

 chClassInfo ChClassInfo OPTIONAL,

 -- Mobility information

 mobilityInformation MobilityInformation OPTIONAL

}

--Reconfiguration response

ReconfigurationResponse ::= SEQUENCE OF SEQUENCE {

 --WSO ID

 wsoID OCTET STRING OPTIONAL,

 --Reconfiguration status

 status Status OPTIONAL,

 --Status

 status BOOLEAN OPTIONAL,

 --Failed parameters

 failedParameters FailedParameters OPTIONAL}

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

**--Stop operation**

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

--Stop operation announcement

StopOperationAnnouncement ::= SEQUENCE {}

--Stop operation confirm

StopOperationConfirm ::= SEQUENCE {

 --Stop operation status

 status Status}

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

**--Coexistence report**

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

--Coexistence report announcement

CoexistenceReportAnnouncement ::= SEQUENCE OF SEQUENCE {

 --Subject WSO ID

 wsoID OCTET STRING,

 --List of subject WSO available frequencies

 listOfSubjectWSOAvailableFrequencies ListOfSubjectWSOAvailableFrequencies}

--Coexistence report confirm

CoexistenceReportConfirm ::= SEQUENCE {

 --Coexistence report status

 status Status}

--Request for coexistence report

CoexistenceReportRequest ::= SEQUENCE {}

--Response for coexistence report

CoexistenceReportResponse ::= SEQUENCE {

 --Coexistence report information

 coexistenceReport CoexistenceReport,

 --Channel priority information

 channelPriority ChannelPriority}

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

**--CM registration**

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

--CM registration request

CMRegistrationRequest ::= SEQUENCE {

 --CM profile

 cmProfile EntityProfile OPTIONAL,

 --CM registration

 cmRegistration CMRegistration OPTIONAL,

 --CE registration

 ceRegistration CERegistration OPTIONAL,

 --Operationg code

 operationCode OperationCode OPTIONAL,

 --CE ID

 ceID CxID OPTIONAL,

 --Network ID

 networkID OCTET STRING OPTIONAL,

 --Network technology

 networkTechnology NetworkTechnology OPTIONAL,

 --Network type

 networkType NetworkType OPTIONAL,

 --Discovery information

 discoveryInformation DiscoveryInformation OPTIONAL,

 --List of supported channel number

 listOfSupportedChNumbers SEQUENCE OF INTEGER OPTIONAL,

 -- List of supported frequencies

 listOfSuppFrequencies ListOfSupportedFrequencies OPTIONAL

}

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

**--Coexistence set information**

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

--Coexistence set information announcement

CoexistenceSetInformationAnnouncement ::= SEQUENCE {

 --List of subject CEs

 listOfSubjectCEs ListOfSubjectCEs,

 --List of neighbor CMs transport information mentioned inside the list of subject CEs

 listOfNeighborCMsTransport ListOfNeighborCMsTransport}

--Coexistence set information confirm

CoexistenceSetInformationConfirm ::= SEQUENCE {

 --Coexistence set information status

 status Status}

--Request for coexistence set information

CoexistenceSetInformationRequest ::= SEQUENCE {

 listOfNetworkID SEQUENCE OF OCTET STRING}

--Response for coexistence set information

CoexistenceSetInformationResponse ::= SEQUENCE {

 --Network ID

 coexistenceReport CoexistenceReport,

 --List of neighbor CM

 listOfneighborCM ListOfneighborCM}

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

**--Coexistence set element information**

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

--Coexistence set element information announcement

CoexistenceSetElementInformationAnnouncement ::= SEQUENCE OF SEQUENCE{

 --CE ID

 ceID CxID,

 --Coexistence service

 coexistenceService CoexistenceService OPTIONAL,

 --List of neighbor CM WSOs

 listOfNeighborCMWSOs ListOfNeighborCMWSOs OPTIONAL

}

--Coexistence set element information confirm

CoexistenceSetElementInformationConfirm ::= SEQUENCE {

 --Coexistence set element information status

 status Status}

--Coexistence set element information request

CoexistenceSetElementInformationRequest ::= SEQUENCE OF SEQUENCE{

 --CE ID

 ceID CxID,

 --List of neighbor CM WSOs

 listOfNeighborCMWSOs ListOfNeighborCMWSOs

}

--Coexistence set element information response

CoexistenceSetElementInformationResponse ::= SEQUENCE OF SEQUENCE{

 --CE ID

 ceID CxID,

 --Coexistence service

 coexistenceService CoexistenceService,

 --List of neighbor CM WSOs

 listOfNeighborCMWSOs ListOfNeighborCMWSOs

}

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

**--Coexistence set element reconfiguration**

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

--Coexistence set element reconfiguration request

CoexistenceSetElementReconfigurationRequest ::= SEQUENCE{

 --List of subject CEs

 reconfigListOfSubjectCEs ReconfigListOfCEs,

 --List of neighbor CEs

 reconfigListOfNeighborCEs ReconfigListOfCEs

}

--Coexistence set element reconfiguration response

CoexistenceSetElementReconfigurationResponse ::= SEQUENCE{

 --Indicates whether the suggested reconfiguration is accepted or not

 requestIsAccepted BOOLEAN

}

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

**--WSO reconfiguration for another CM**

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

--Reconfiguration request

CMReconfigurationRequest ::= SEQUENCE {

 --Indication for CE to be reconfigured

 reconfigTarget CxID,

 --List of operating channel number

 listOfOperatingChNumber SEQUENCE OF INTEGER,

 --Transmission power limitation

 txPowerLimit REAL,

 --Indication whether the channel is shared

 channelIsShared BOOLEAN,

 --Transmission schedule

 txSchedule TxSchedule ,

 -- Channel classification information

 chClassInfo ChClassInfo}

--Reconfiguration response

ReconfigurationResponse ::= SEQUENCE {

 --Indication for CE to be reconfigured

 reconfigTarget CxID,

 --Status

 status BOOLEAN

 --Failed parameters

 failedParameters FailedParameters}

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

**--Channel classification**

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

--Channel classification request

ChannelClassificationRequest ::= SEQUENCE {

 -- List of network ID

 listOfNetworkID SEQUENCE OF OCTET STRING}

-- Channel classification response

ChannelClassificationResponse ::= SEQUENCE OF SEQUENCE{

 -- Network ID

 networkID OCTET STRING

 --Channel classification information

 chClassInfo ChClassInfo}

--CM Channel classification request

CMChannelClassificationRequest ::= SEQUENCE {

 -- List of network ID

 listOfNetworkID SEQUENCE OF OCTET STRING}

-- CM Channel classification response

CMChannelClassificationResponse ::= SEQUENCE OF SEQUENCE{

 -- Network ID

 networkID OCTET STRING

 --Channel classification information

 chClassInfo ChClassInfo}

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

**--Channel classification update**

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

--Channel classification update

ChannelClassificationAnnouncement ::= SEQUENCE OF SEQUENCE {

 -- Network ID

 networkID OCTET STRING

 --Channel classification information

 chClassInfo ChClassInfo}

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

**--Information acquiring from another CM**

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

-- Information acquiring request

InfoAcquringRequest ::= SEQUENCE {

 ceID CxID,

 listOfReqInfoDescr SEQUENCE OF ReqInfoDescr}

-- Information acquiring response

InfoAcquringResponse ::= SEQUENCE {

 ceID CxID,

 reqInfoValue ReqInfoValue}

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

**--Available channel list from WSO**

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

-- Available channel request

AvailableChannelsRequest ::= SEQUENCE {}

-- Available channel response

AvailableChannelsResponse ::= SEQUENCE {

 --Available channel list information

 listOfAvailableChNumbers ListOfAvailableChNumbers}

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

**--Event indication**

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

-- Event indication

EventIndication ::= SEQUENCE {

 -- Event indication information

 eventParams EventParams}

-- Event confirm

EventConfirm ::= SEQUENCE {}

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

**--Measurement Request**

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

-- Measurement request

MeasurementRequest ::= SEQUENCE {

 -- Measurement request information

 measurementDescription MeasurementDescription}

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

**-- Measurement results**

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

-- Measurement response

MeasurementResponse ::= SEQUENCE OF SEQUENCE{

 -- Measurement results

 measurementResult MeasurementResult}

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

**-- Negotiation**

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

-- Negotiation request

NegotiationRequest ::= SEQUENCE {

 -- Negotiation status

 negotiationStatus NegotiationStatus,

 -- Negotiation information

 negotiationInformation NegotiationInformation}

-- Negotiation announcement

NegotiationAnnouncement ::= SEQUENCE {

 -- Winner CM ID list

 listOfWinnerCMID ListOfWinnerCMID,

 -- Slot time position list

 listOfSlotTimePosition ListOfSlotTimePosition}

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

**--Master/Slave CM selection**

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

--Master/Slave CM selection request

MasterCMRequest ::= SEQUENCE {

 --List of CEs managed by CM that intends to become slave CM

 listOfCEs SEQUENCE OF CxID}

-- Master/Slave CM selection response

MasterCMResponse ::= SEQUENCE {

 --Status

 status CxMediaStatus}

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

**--Master/Slave CM configuration**

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

--Master/Slave CM configuration request

MasterSlaveCMconfigurationRequest ::= SEQUENCE {

 -- List of CEs managed by CM

 listOfCEs SEQUENCE OF CxID}

--Master/Slave CM configuration response

MasterSlaveCMconfigurationResponse ::= SEQUENCE {

 --Operationg code

 operationCode OperationCode,

 --Slave CE ID

 slaveCeID CxID,

 --Network ID

 networkID OCTET STRING,

 --Network technology

 networkTechnology NetworkTechnology,

 --Network type

 networkType NetworkType

 --Discovery information

 discoveryInformation DiscoveryInformation,

 --Transmission schedule is supported or not

 txScheduleSupported BOOLEAN,

 --List of available channel number

 listOfAvailableChNumbers ListOfAvailableChNumbers,

 --List of supported channel number

 listOfSupportedChNumbers SEQUENCE OF INTEGER,

 --List of operating channel number

 listOfOperatingChNumbers ListOfOperatingChNumbers,

 --Required resource

 requiredResource RequiredResource

 --Measurement capability

 measurementCapability MeasurementCapability}

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

**--WSO deregistration**

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

--Deregistration request

WsoDeregistrationRequest ::= SEQUENCE {

 --Flag of wso deregistration

 wsoDeregistration BOOLEAN}

--Deregistration response

WsoDeregistrationResponse ::= SEQUENCE {

 --Status

 status CxMediaStatus}

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

**--Inter-CM information**

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

--Inter-CM information announcement

InterCMInformationAnnouncement ::= SEQUENCE OF SEQUENCE {

 --CE ID

 ceID CxID,

 --List of neighbor WSOs

 listOfNeighborWSOs SEQUENCE OF SEQUENCE {

 wsoID OCTET STRING,

 operatingFrequency ListOfOperatingFrequencies OPTIONAL,

 operatingChannel ListOfOperatingChNumbers OPTIONAL

 }

}

--Inter-CM information confirm

InterCMInformationConfirm ::= SEQUENCE {

 --Inter-CM information status

 status Status

}

--Inter-CM information request

InterCMInformationRequest ::= SEQUENCE OF SEQUENCE{

 --CE ID

 ceID CxID,

 --List of neighbor WSOs

 listOfNeighborWSOs SEQUENCE OF SEQUENCE {

 wsoID OCTET STRING

 }

}

--Inter-CM information response

InterCMInformationResponse ::= SEQUENCE OF SEQUENCE{

 --CE ID

 ceID CxID,

 --List of neighbor WSOs

 listOfNeighborWSOs SEQUENCE OF SEQUENCE {

 wsoID OCTET STRING,

 operatingFrequency ListOfOperatingFrequencies OPTIONAL,

 operatingChannel ListOfOperatingChNumbers OPTIONAL

 }

}

END

*It is proposed to* ***ListOfNeighborCMTransport*** *data type as shown in the text below.*

--List of neighbor CMs transport information

ListOfNeighborCMsTransport ::= SEQUENCE OF SEQUENCE {

 --Neighbor CM ID

 cmID CxID,

 --Neighbor CM profile

 cmProfile EntityProfile,

 -- Neighbor CM IP address

 ipAddress OCTET STRING,

 -- Neighbor CM port number

 portNumber INTEGER}

*It is proposed to* ***ListOfNeighborCM*** *data type as shown in the text below.*

ListOfNeighborCM ::= SEQUENCE OF SEQUENCE {

 neighborCMID CxID,

 --Neighbor CM profile

 cmProfile EntityProfile, listOfCoexSetElement ListOfCoexSetElement

}

*It is proposed to add the text below as 6.3.1 Communication between CMs.*

* + 1. Communication between CMs

When a CM operating according to profile 1 receives a ***CoexistenceSetInformationResponse*** message from the CDIS that serves this CM, it shall check the value of the parameter ***cmProfile*** for each neighbor CM.

If the value of the ***cmProfile*** parameter for a neighbor CM is equal to ***profile1***, the CM shall communicate with this neighbor CM as described in clause CM profile 1.

If the value of the ***cmProfile*** parameter for a neighbor CM is equal to ***profile2***, the CM shall do the following.

If operating channel is changed for a WSO served by the CM operating according to profile 1 and this WSO has neighbor WSOs served by one or several CMs operating according to profile 2, the CM shall perform the **providing inter-CM information procedure**, described in clause reference.

The CM shall generate and send one or several ***InterCMInformationAnnouncement*** messages to one or several neighbor CMs. The number of messages is equal to the number of CMs operating according to profile 2 that serve WSOs that are neighbors to the WSO which operating channel has been changed.

When generating a ***InterCMInformationAnnouncement*** message, the CM shall set the parameters of the ***CxMessage*** as shown in the table below.

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***header*** | ***CxHeader*** | ***requestID*** |
| ***payload*** | ***CxPayload*** | ***interCMInformationAnnouncement*** |

The CM shall set the parameters of the each element of the sequence of the ***interCMInformationAnnouncement*** payload as shown in the table below.

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***ceID*** | ***CxID*** | CE ID of the CE that serves WSO which operating channel is changed. |
| ***listOfNeighborWSOs*** | ***SEQUENCE OF SEQUENCE*** | See table below. |

The CM shall set the parameters of the each element of the sequence of the ***listOfNeighborWSOs*** payload as shown in the table below.

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***wsoID*** | ***OCTET STRING*** | ID of the WSO which operating channel is changed. |
| ***operatingFrequency*** | ***ListOfOperatingFrequencies*** | *This parameter is not used.* |
| ***operatingChannel*** | ***ListOfOperatingChNumbers*** | List of operating channel numbers of the WSO. |

After the CM has generated a ***InterCMInformationAnnouncement*** message, the CM shall send this message to a corresponding CM and shall wait for the ***InterCMInformationConfirm*** message.

Table below shows expected values of the parameters in the ***InterCMInformationConfirm*** message.

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***header*** | ***CxHeader*** | ***requestID*** |
| ***payload*** | ***CxPayload*** | ***interCMInformationConfirm*** |

Table below shows expected values of the parameters in the ***interCMInformationConfirm*** payload.

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***status*** | ***Status*** | ***noError*** |

If the CM operating according to profile 1 needs to obtain neighbor WSO operating channel information from a CM operating according to profile 2, the CM shall perform the **obtaining inter-CM information procedure**, described in clause reference.

The CM operating according to profile 1 shall generate and send the ***InterCMInformationRequest*** message to the neighbor CM operating according to profile 2.

When generating a ***InterCMInformationRequest*** message, the CM shall set the parameters of the ***CxMessage*** as shown in the table below.

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***header*** | ***CxHeader*** | ***requestID*** |
| ***payload*** | ***CxPayload*** | ***interCMInformationRequest*** |

The CM shall set the parameters of the each element of the sequence of the ***interCMInformationRequest*** payload as shown in the table below.

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***ceID*** | ***CxID*** | CE ID of the CE that serves WSO which operating channel information is requested. |
| ***listOfNeighborWSOs*** | ***SEQUENCE OF SEQUENCE*** | List of WSO IDs of the WSOs served by this CE which operating channel information is requested. |

After the CM has generated a ***InterCMInformationRequest*** message, the CM shall send this message to a corresponding CM and shall wait for the ***InterCMInformationResponse*** message.

Table below shows expected values of the parameters in the ***InterCMInformationResponse*** message.

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***header*** | ***CxHeader*** | ***requestID*** |
| ***payload*** | ***CxPayload*** | ***interCMInformationResponse*** |

Table below shows expected values of the parameters in the ***interCMInformationResponse*** payload.

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***wsoID*** | ***OCTET STRING*** | ID of the WSO which operating channel information is requested. |
| ***operatingFrequency*** | ***ListOfOperatingFrequencies*** | Operating frequency of the WSO. |
| ***operatingChannel*** | ***ListOfOperatingChNumbers*** | *This parameter is not used.* |

When a CM operating according to profile 2 receives a ***CoexistenceSetInformationAnnouncement*** message from the CDIS that serves this CM, it shall check the value of the parameter ***cmProfile*** for each neighbor CM.

If the value of the ***cmProfile*** parameter for a neighbor CM is equal to ***profile2***, the CM shall communicate with this neighbor CM as described in clause CM profile 2.

If the value of the ***cmProfile*** parameter for a neighbor CM is equal to ***profile1***, the CM shall do the following.

If operating channel is changed for a WSO served by the CM operating according to profile 2 and this WSO has neighbor WSOs served by one or several CMs operating according to profile 1, the CM shall perform the **providing inter-CM information procedure**, described in clause reference.

The CM shall generate and send one or several ***InterCMInformationAnnouncement*** messages to one or several neighbor CMs. The number of messages is equal to the number of CMs operating according to profile 1 that serve WSOs that are neighbors to the WSO which operating channel has been changed.

When generating a ***InterCMInformationAnnouncement*** message, the CM shall set the parameters of the ***CxMessage*** as shown in the table below.

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***header*** | ***CxHeader*** | ***requestID*** |
| ***payload*** | ***CxPayload*** | ***interCMInformationAnnouncement*** |

The CM shall set the parameters of the each element of the sequence of the ***interCMInformationAnnouncement*** payload as shown in the table below.

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***ceID*** | ***CxID*** | CE ID of the CE that serves WSO which operating channel is changed. |
| ***listOfNeighborWSOs*** | ***SEQUENCE OF SEQUENCE*** | See table below. |

The CM shall set the parameters of the each element of the sequence of the ***listOfNeighborWSOs*** payload as shown in the table below.

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***wsoID*** | ***OCTET STRING*** | ID of the WSO which operating channel is changed. |
| ***operatingFrequency*** | ***ListOfOperatingFrequencies*** | List of operating frequencies of the WSO. |
| ***operatingChannel*** | ***ListOfOperatingChNumbers*** | *This parameter is not used.* |

After the CM has generated a ***InterCMInformationAnnouncement*** message, the CM shall send this message to a corresponding CM and shall wait for the ***InterCMInformationConfirm*** message.

Table below shows expected values of the parameters in the ***InterCMInformationConfirm*** message.

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***header*** | ***CxHeader*** | ***requestID*** |
| ***payload*** | ***CxPayload*** | ***interCMInformationConfirm*** |

Table below shows expected values of the parameters in the ***interCMInformationConfirm*** payload.

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***status*** | ***Status*** | ***noError*** |

If the CM operating according to profile 2 needs to obtain neighbor WSO operating channel information from a CM operating according to profile 1, the CM shall perform the **obtaining inter-CM information procedure**, described in clause reference.

The CM operating according to profile 2 shall generate and send the ***InterCMInformationRequest*** message to the neighbor CM operating according to profile 1.

When generating a ***InterCMInformationRequest*** message, the CM shall set the parameters of the ***CxMessage*** as shown in the table below.

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***header*** | ***CxHeader*** | ***requestID*** |
| ***payload*** | ***CxPayload*** | ***interCMInformationRequest*** |

The CM shall set the parameters of the each element of the sequence of the ***interCMInformationRequest*** payload as shown in the table below.

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***ceID*** | ***CxID*** | CE ID of the CE that serves WSO which operating channel information is requested. |
| ***listOfNeighborWSOs*** | ***SEQUENCE OF SEQUENCE*** | List of WSO IDs of the WSOs served by this CE which operating channel information is requested. |

After the CM has generated a ***InterCMInformationRequest*** message, the CM shall send this message to a corresponding CM and shall wait for the ***InterCMInformationResponse*** message.

Table below shows expected values of the parameters in the ***InterCMInformationResponse*** message.

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***header*** | ***CxHeader*** | ***requestID*** |
| ***payload*** | ***CxPayload*** | ***interCMInformationResponse*** |

Table below shows expected values of the parameters in the ***interCMInformationResponse*** payload.

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***wsoID*** | ***OCTET STRING*** | ID of the WSO which operating channel information is requested. |
| ***operatingFrequency*** | ***ListOfOperatingFrequencies*** | *This parameter is not used.* |
| ***operatingChannel*** | ***ListOfOperatingChNumbers*** | Operating channel of the WSO. |