IEEE P802.19
Wireless Coexistence

|  |
| --- |
| Proposed CE operation |
| Date: 2013-03-18 |
| Author(s): |
| Name | Company | Address | Phone | email |
| Stanislav Filin | NICT |  |  | sfilin@nict.go.jp |
| Hiroshi Harada | NICT |  |  |  |

Abstract

This document is a submission to IEEE 802.19 TG1 proposing CE operation.

**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 create new section Entities Operation and add subsection CE Operation for Profile N using the text below.*

# Entities Operation

# CE Operation for Profile N

# General description

Figure below shows operation of a CE for IEEE 802.22, IEEE 802.11af, and RLSS.



After receiving a request to start operation, a CE shall perform the start-up sequence comprising two procedures: WSO subscription and WSO registration. After that, the CE shall switch to the waiting state, where it shall check for incoming messages or primitives or for request to stop operation.

If the WSO/RLSS is subscribed to management service and the CE has received a reconfiguration request, the CE shall perform WSO reconfiguration and switch back to the waiting state.

If the WSO/RLSS is subscribed to information service and the CE has received a coexistence report, the CE shall perform providing coexistence report and switch back to the waiting state.

If the WSO registration information has been changed, the CE shall perform WSO registration update and switch back to the waiting state.

If the WSO/RLSS subscription has been changed, the CE shall perform WSO subscription update and switch back to the waiting state.

If the CM serving this CE has stopped operation, the CE shall perform CM stop operation and shall check whether this has been the last available CM in the CE configuration file. If the CM has been the last available CM, the CE shall stop operation. If there is any available CM in the CE configuration file, the CE shall select one of the available CMs and initiate the start-up sequence with the selected CM.

If the CE stops operation, the CE shall perform CE stop operation and shall stop operation.

If the RLSS stops operation, the CE shall perform WSO/RLSS subscription update and shall stop operation.

If the WSO stops operation, the CE shall perform WSO registration update and shall stop operation.

# WSO subscription

During the **WSO subscription procedure**, described in (reference), the CE shall do the following.

First the CE shall generate and send the ***CxMediaSubscriptionRequest*** primitive to the WSO/RLSS it serves and shall wait for the ***CxMediaSubscriptionResponse*** primitive from the WSO.

Table below shows expected values of the parameters in the ***CxMediaSubscriptionResponse*** primitive.

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***clientID*** | ***IA5String*** | WSO/RLSS subscription ID. |
| ***clientPassword*** | ***IA5String*** | WSO/RLSS subscription password. |
| ***listOfCMs*** | ***SEQUENCE OF …*** | List of serving CMs (see table below). |
| ***coexistenceService*** | ***CoexistenceService*** | WSO/RLSS subscription to coexistence service (***management*** or ***information***). |

Table below shows expected values of the parameters of the each element of the sequence in the ***listOfCMs*** parameter. Number of elements in the sequence shall be at least one.

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***cmID*** | ***CxID*** | CM ID. |
| ***cmIPAddress*** | ***OCTET STRING*** | CM IP address. |
| ***cmPortNumber*** | ***INTEGER*** | CM port number. |
| ***serverID*** | ***IA5String*** | CM subscription ID. |
| ***serverPassword*** | ***IA5String*** | CM subscription password. |

After the CE has received the ***CxMediaSubscriptionResponse*** primitive from the WSO/RLSS, the CE shall select one of the CMs from the list and establish connection with this CM. Then the CE shall generate and send the ***SubscriptionRequest*** message to the selected CM and wait for the ***SubscriptionResponse*** message from the CM.

When generating the ***SubscriptionRequest*** message, the CE shall set the parameters of the ***CxMessage*** as shown in the table below.

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

The CE shall set the parameters of the ***subscriptionRequest*** payload as shown in the table below.

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***clientID*** | ***IA5String*** | The value of the ***clientID*** parameter from the received ***CxMediaSubscriptionResponse*** primitive. |
| ***clientPassword*** | ***IA5String*** | The value of the ***clientPassword*** parameter from the received ***CxMediaSubscriptionResponse*** primitive. |
| ***coexistenceService*** | ***CoexistenceService*** | The value of the ***coexistenceService*** parameter from the received ***CxMediaSubscriptionResponse*** primitive. |

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

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

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

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***serverID*** | ***IA5String*** | The value of the ***serverID*** parameter from the received ***CxMediaSubscriptionResponse*** primitive. |
| ***serverPassword*** | ***IA5String*** | The value of the ***serverPassword*** parameter from the received ***CxMediaSubscriptionResponse*** primitive. |
| ***status*** | ***Status*** | ***noError*** |

After the CE has received the ***SubscriptionResponse*** message from the CM, the CE shall generate and send the ***CxMediaSubscriptionConfirm*** primitive to the WSO/RLSS.

The CE shall set the parameters of the ***CxMediaSubscriptionConfirm*** primitive as shown in the table below.

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***status*** | ***Status*** | ***noError***, if subscription is confirmed |

# WSO registration

During the **WSO registration procedure**, described in (reference), the CE shall do the following.

First the CE shall generate and send the ***CxMediaRegistrationRequest*** primitive to the WSO/RLSS it serves and shall wait for the ***CxMediaRegistrationResponse*** primitive from the WSO/RLSS.

Table below shows expected values of the parameters in the each element of the sequence in the ***CxMediaRegistrationResponse*** primitive. The number of elements shall be equal to one for IEEE 802.22 CE and IEEE 802.11af CE. For the RLSS CE, the number of elements is equal to the number of WSOs that RLSS registers by this procedure.

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***wsoID*** | ***OCTET STRING*** | In case of the IEEE 802.22 CE and IEEE 802.11af CE, the value of this parameter shall be set to ***0***. In case of the RLSS CE, each WSO served by this RLSS shall be given unique ID. |
| ***networkID*** | ***OCTET STRING*** | Network ID. |
| ***networkTechnology*** | ***NetworkTechnology*** | This parameter shall be set to the WSO network technology. |
| ***geolocation*** | ***Geolocation*** | ***coordinate*** shall be selected and set to the location of the WSO. |
| ***coverageArea*** | ***CoverageArea*** | If coverage area of the WSO is not known, then this parameter is not used. If it is know, this parameter is used (see table below). |
| ***installationParameters*** | ***InstallationParameters*** | If installation parameters of the WSO are not known, then this parameter is not used. If they are know, this parameter is used (see table below). |
| ***listOfAvailableFrequencies*** | ***ListOfAvailableFrequencies*** | List of available frequencies from TVWS DB (see table below). |
| ***txScheduleSupported*** | ***BOOLEAN*** | ***TRUE*** if IEEE 802.19.1 transmission scheduling is supported, ***FALSE*** otherwise. |
| ***listOfOperatingFrequencies*** | ***ListOfOperatingFrequencies*** | If the WSO is not operating, this parameter is not used, otherwise it is set to the WSO operating frequency (see table below). |
| ***requiredResource*** | ***RequiredResource*** | Resource required for the WSO operation (see table below). |

Table below shows expected values of the parameters in the ***coverageArea*** parameter.

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***radius*** | ***REAL*** | Radius of the coverage area. |
| ***refFrequency*** | ***REAL*** | Reference frequency for which the radius is calculated. |
| ***refMasterHeight*** | ***REAL*** | Reference height of the master station for which the radius is calculated. |
| ***refSlaveHeight*** | ***REAL*** | Reference height of the slave station for which the radius is calculated. |
| ***refTxPower*** | ***REAL*** | Reference transmission power for which the radius is calculated. |

Table below shows expected values of the parameters in the ***installationParameters*** parameter.

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***opMasterHeight*** | ***REAL*** | Height of the master station. |
| ***opSlaveHeight*** | ***REAL*** | Height of the slave station. |
| ***opTxPower*** | ***REAL*** | Transmission power (minimum of the master and slave stations). |

Table below shows expected values of the parameters of the each element of the sequence in the ***listOfAvailableFrequencies*** parameter.

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***frequencyRange*** | ***FrequencyRange*** | Frequency range of the available frequency. |
| ***txPowerLimit*** | ***REAL*** | Power limit in this available frequency. |
| ***availableStartTime*** | ***GeneralizedTime*** | Available start time of the available frequency if applicable. |
| ***availableDuration*** | ***REAL*** | Available duration of the available frequency if applicable. |

Table below shows expected values of the parameters of the each element of the sequence in the ***listOfOperatingFrequencies*** parameter. The number of elements shall be one.

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***frequencyRange*** | ***FrequencyRange*** | Frequency range of the operating frequency. |
| ***occupancy*** | ***REAL*** | If occupancy is not known, this parameter is not used. If it is known, this parameter is set to its value. Range is from 0 to 1. |

Table below shows expected values of the parameters of the each element of the sequence in the ***requiredResource*** parameter. The number of elements shall be one.

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***requiredBandwidth*** | ***REAL*** | Required bandwidth for the WSO operation. |
| ***occupancy*** | ***REAL*** | If expected occupancy is not known, this parameter is not used. If it is known, this parameter is set to its value. Range is from 0 to 1. |

After the CE has received the ***CxMediaRegistrationResponse*** primitive from the WSO/RLSS, the CE shall generate and send the ***CERegistrationRequest*** message to the CM that serves this CE and wait for the ***RegistrationResponse*** message from the CM.

When generating the ***CERegistrationRequest*** message, the CE shall set the parameters of the ***CxMessage*** as shown in the table below.

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

The CE shall set the parameters of the each element of the sequence in the ***ceRegistrationRequest*** payload as shown in the table below. The number of elements of the sequence shall be equal to the number of elements in the received ***CxMediaRegistrationResponse*** primitive.

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***operationCode*** | ***OperationCode*** | ***new*** |
| ***wsoID*** | ***OCTET STRING*** | Value of ***wsoID*** parameter from the received ***CxMediaRegistrationResponse*** primitive. |
| ***networkID*** | ***OCTET STRING*** | Value of ***networkID*** parameter from the received ***CxMediaRegistrationResponse*** primitive. |
| ***networkTechnology*** | ***NetworkTechnology*** | Value of ***networkTechnology*** parameter from the received ***CxMediaRegistrationResponse*** primitive. |
| ***geolocation*** | ***Geolocation*** | Value of ***geolocation*** parameter from the received ***CxMediaRegistrationResponse*** primitive. |
| ***coverageArea*** | ***CoverageArea*** | Value of ***coverageArea*** parameter from the received ***CxMediaRegistrationResponse*** primitive. |
| ***installationParameters*** | ***InstallationParameters*** | Value of ***installationParameters*** parameter from the received ***CxMediaRegistrationResponse*** primitive. |
| ***listOfAvailableFrequencies*** | ***ListOfAvailableFrequencies*** | Value of ***listOfAvailableFrequencies*** parameter from the received ***CxMediaRegistrationResponse*** primitive. |
| ***txScheduleSupported*** | ***BOOLEAN*** | Value of ***txScheduleSupported*** parameter from the received ***CxMediaRegistrationResponse*** primitive. |
| ***listOfOperatingFrequencies*** | ***ListOfOperatingFrequencies*** | Value of ***listOfOperatingFrequencies*** parameter from the received ***CxMediaRegistrationResponse*** primitive. |
| ***requiredResource*** | ***RequiredResource*** | Value of ***requiredResource*** parameter from the received ***CxMediaRegistrationResponse*** primitive. |

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

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

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

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

After the CE has received the ***RegistrationResponse*** message from the CM, the CE shall generate and send the ***CxMediaRegistrationConfirm*** primitive to the WSO/RLSS.

The CE shall set the parameters of the ***CxMediaRegistrationConfirm*** primitive as shown in the table below.

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***status*** | ***Status*** | Value of ***status*** parameter from the received ***RegistrationResponse*** message. |

# WSO reconfiguration for management service

This procedure shall be performed only if the WSO/RLSS served by the CE is subscribed to management service.

After the CE has received the ***ReconfigurationRequest*** message from its serving CM, the CE shall perform the **WSO reconfiguration procedure**, described in (reference).

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

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

Table below shows expected values of the parameters of the each element of the sequence in the ***reconfigurationRequest*** payload. The number of elements shall be equal to one for IEEE 802.22 CE and IEEE 802.11af CE. For the RLSS CE, the number of elements is equal to the number of WSOs that needs to be reconfigured by this procedure.

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***wsoID*** | ***OCTET STRING*** | WSO ID. |
| ***operatingFrequency*** | ***FrequencyRange*** | Operating frequency for the WSO. |
| ***txPowerLimit*** | ***REAL*** | *This parameter is not used.* |
| ***channelIsShared*** | ***BOOLEAN*** | ***TRUE***, if the frequency will be shared with other WSOs, ***FALSE*** otherwise. |
| ***txSchedule*** | ***txSchedule*** | *This parameter is not used.* |

After the CE has received the ***ReconfigurationRequest*** message from the CM, the CE shall generate and send the ***CxMediaReconfigurationRequest*** primitive to the WSO/RLSS it serves and shall wait for the ***CxMediaReconfigurationResponse*** primitive from the WSO/RLSS.

The CE shall set the parameters of the each element of the sequence in the ***CxMediaReconfigurationRequest*** primitive as shown in the table below. The number of the elements shall be equal to the number of the elements in the received ***ReconfigurationRequest*** message.

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***wsoID*** | ***OCTET STRING*** | Value of ***wsoID*** parameter from the received ***ReconfigurationRequest*** message. |
| ***operatingFrequency*** | ***FrequencyRange*** | Value of ***operatingFrequency*** parameter from the received ***ReconfigurationRequest*** message. |
| ***txPowerLimit*** | ***REAL*** | Value of ***txPowerLimit*** parameter from the received ***ReconfigurationRequest*** message. |
| ***channelIsShared*** | ***BOOLEAN*** | Value of ***channelIsShared*** parameter from the received ***ReconfigurationRequest*** message. |
| ***txSchedule*** | ***txSchedule*** | Value of ***txSchedule*** parameter from the received ***ReconfigurationRequest*** message. |

Table below shows expected values of the parameters of the each element of the sequence in the ***CxMediaReconfigurationResponse*** primitive. The number of the elements shall be equal to the number of the elements in the sent ***CxMediaReconfigurationRequest*** primitive.

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***status*** | ***Status*** | ***noError***, if reconfiguration is successful. |

After the CE has received the ***CxMediaReconfigurationResponse*** primitive from the WSO/RLSS, the CE shall generate and send the ***ReconfigurationResponse*** message to its serving CM.

The CE shall set the parameters of the each element in the sequence in the ***ReconfigurationResponse*** message as shown in the table below. The number of the elements shall be equal to the number of the elements in the received ***CxMediaReconfigurationResponse*** primitive.

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***status*** | ***Status*** | Value of ***status*** parameter from the received ***CxMediaReconfigurationResponse*** primitive. |

# Coexistence report for information service

This procedure shall be performed only if the WSO served by the CE is subscribed to information service.

After the CE has received the ***CoexistenceReportAnnouncement*** message from its serving CM, the CE shall perform the **providing coexistence report procedure**, described in (reference).

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

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

Table below shows expected values of the parameters of the each element of the sequence in the ***coexistenceReportAnnouncement*** payload. The number of elements shall be equal to one for IEEE 802.22 CE and IEEE 802.11af CE. For the RLSS CE, the number of elements is equal to the number of WSOs which coexistence report has been changed.

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***wsoID*** | ***OCTET STRING*** | WSO ID. |
| ***listOfSubjectWSO******AvailableFrequencies*** | ***ListOfSubjectWSO******AvailableFrequencies*** | List of the available frequencies of the subject WSO (see table below). |

Table below shows expected value of each element in the sequence of the ***listOfSubjectWSOAvailableFrequencies*** parameter.

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***frequencyRange*** | ***FrequencyRange*** | Frequency range of the available frequency. |
| ***listOfNeighborCMs*** | ***ListOfNeighborCMs*** | List of neighbor CMs on this available frequency (see table below). |

Table below shows expected value of each element in the sequence of the ***listOfNeighborCMs*** parameter.

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***cmID*** | ***CxID*** | CM ID. |
| ***listOfNeighborCEs*** | ***ListOfNeighborCEs*** | List of neighbor CEs served by this neighbor CM (see table below). |

Table below shows expected value of each element in the sequence of the ***listOfNeighborCEs*** parameter.

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***ceID*** | ***CxID*** | CE ID. |
| ***listOfNeighborWSOs*** | ***ListOfNeighborWSOs*** | List of neighbor WSOs served by this neighbor CE (see table below). |

Table below shows expected value of each element in the sequence of the ***listOfNeighborWSOs*** parameter.

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***wsoID*** | ***OCTET STRING*** | WSO ID. |
| ***networkTechnology*** | ***NetworkTechnology*** | Network technology of the neighbor WSO. |
| ***interferenceDirection*** | ***InterferenceDirection*** | Interference direction between subject WSO and neighbor WSO. |
| ***distance*** | ***REAL*** | Distance between subject WSO and neighbor WSO used as a measure of interference. |
| ***listOfOperatingFrequencies*** | ***ListOfOperatingFrequencies*** | List of operating frequencies of the neighbor WSO (see table below). |

Table below show expected value of each element in the sequence of the ***listOfOperatingFrequencies*** parameter. The number of elements shall be one.

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***frequencyRange*** | ***FrequencyRange*** | Frequency range of the operating frequency. |
| ***Occupancy*** | ***REAL*** | *This parameter is not used.* |

After the CE has received the ***CoexistenceReportAnnouncement*** message from the CM, the CE shall generate and send the ***CxMediaCoexistenceReportIndication*** primitive to the WSO/RLSS it serves and shall wait for the ***CxMediaCoexistenceReportConfirm*** primitive from the WSO/RLSS.

The CE shall set the parameters of the each element of the sequence in the ***CxMediaCoexistenceReportIndication*** primitive as shown in the table below.

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***wsoID*** | ***OCTET STRING*** | Value of ***wsoID*** parameter from the received ***CoexistenceReportAnnouncement*** message. |
| ***listOfSubjectWSO******AvailableFrequencies*** | ***ListOfSubjectWSO******AvailableFrequencies*** | Value of ***listOfSubjectWSOAvailableFrequencies*** parameter from the received ***CoexistenceReportAnnouncement*** message. |

Table below shows expected values of the parameters in the ***CxMediaCoexistenceReportConfirm*** primitive.

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

After the CE has received the ***CxMediaCoexistenceReportConfirm*** primitive from the WSO, the CE shall generate and send the ***CoexistenceReportConfirm*** message to its serving CM.

The CE shall set the parameters of the ***CoexistenceReportConfirm*** message as shown in the table below.

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***status*** | ***Status*** | Value of ***status*** parameter from the received ***CxMediaCoexistenceReportConfirm*** primitive. |

# WSO registration update

After the CE has received the ***CxMediaRegistrationIndication*** primitive from its WSO/RLSS, the CE shall perform the **WSO registration update procedure**, described in (reference).

Table below shows expected values of the parameters of the each element of the sequence in the ***CxMediaRegistrationIndication*** primitive. The number of elements shall be equal to one for IEEE 802.22 CE and IEEE 802.11af CE. For the RLSS CE, the number of elements is equal to the number of WSOs that needs to do registration update by this procedure.

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***wsoID*** | ***OCTET STRING*** | WSO ID. |
| ***wsoStopOperation*** | ***BOOLEAN*** | ***FALSE*** |
| ***listOfAvailableFrequencies*** | ***ListOfAvailableFrequencies*** | List of available frequencies from TVWS DB (see table below). |
| ***listOfOperatingFrequencies*** | ***ListOfOperatingFrequencies*** | List of operating frequencies (see table below). |
| ***requiredResource*** | ***RequiredResource*** | *This parameter is not used.* |

Among the parameters described in the table below only the parameters that has been changed are included in the ***CxMediaRegistrationIndication*** primitive.

Table below shows expected values of the parameters of the each element of the sequence in the ***listOfAvailableFrequencies*** parameter.

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***frequencyRange*** | ***FrequencyRange*** | Frequency range of the available frequency. |
| ***txPowerLimit*** | ***REAL*** | Power limit in this available frequency. |
| ***availableStartTime*** | ***GeneralizedTime*** | Available start time of the available frequency if applicable. |
| ***availableDuration*** | ***REAL*** | Available duration of the available frequency if applicable. |

Table below shows expected values of the parameters of the each element of the sequence in the ***listOfOperatingFrequencies*** parameter. The number of elements shall be one.

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***frequencyRange*** | ***FrequencyRange*** | Frequency range of the operating frequency. |
| ***Occupancy*** | ***REAL*** | If occupancy is not known, this parameter is not used. If it is known, this parameter is set to its value. Range is from 0 to 1. |

After the CE has received the ***CxMediaRegistrationIndication*** primitive from the WSO/RLSS, the CE shall generate and send the ***CERegistrationRequest*** message to the CM that serves this CE and wait for the ***RegistrationResponse*** message from the CM.

When generating the ***CERegistrationRequest*** message, the CE shall set the parameters of the ***CxMessage*** as shown in the table below.

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

The CE shall set the parameters of the each element of the sequence of the ***ceRegistrationRequest*** payload as shown in the table below. The number of elements shall be equal to the number of elements in the received ***CxMediaRegistrationIndication*** primitive.

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***operationCode*** | ***OperationCode*** | ***update*** |
| ***wsoID*** | ***OCTET STRING*** | Value of ***wsoID*** parameter from the received ***CxMediaRegistrationIndication*** primitive. |
| ***networkID*** | ***OCTET STRING*** | *This parameter is not used.* |
| ***networkTechnology*** | ***NetworkTechnology*** | *This parameter is not used.* |
| ***geolocation*** | ***Geolocation*** | *This parameter is not used.* |
| ***coverageArea*** | ***CoverageArea*** | *This parameter is not used.* |
| ***installationParameters*** | ***InstallationParameters*** | *This parameter is not used.* |
| ***listOfAvailableFrequencies*** | ***ListOfAvailableFrequencies*** | Value of ***listOfAvailableFrequencies*** parameter from the received ***CxMediaRegistrationIndication*** primitive. |
| ***txScheduleSupported*** | ***BOOLEAN*** | *This parameter is not used.* |
| ***listOfOperatingFrequencies*** | ***ListOfOperatingFrequencies*** | Value of ***listOfOperatingFrequencies*** parameter from the received ***CxMediaRegistrationIndication*** primitive. |
| ***requiredResource*** | ***RequiredResource*** | *This parameter is not used.* |

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

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

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

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

After the CE has received the ***RegistrationResponse*** message from the CM, the CE shall generate and send the ***CxMediaRegistrationConfirm*** primitive to the WSO/RLSS.

The CE shall set the parameters of the ***CxMediaRegistrationConfirm*** primitive as shown in the table below.

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***status*** | ***Status*** | Value of ***status*** parameter from the received ***RegistrationResponse*** message. |

# WSO subscription update

After the CE has received the ***CxMediaSubscriptionIndication*** primitive from its WSO/RLSS indicating that WSO/RLSS is going to stop operation, the CE shall perform the **WSO subscription update procedure**, described in (reference).

Table below shows expected values of the parameters in the ***CxMediaSubscriptionIndication*** primitive.

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***coexistenceService*** | ***CoexistenceService*** | New subscribed coexistence service (***information*** or ***management***). |

After the CE has received the ***CxMediaSubscriptionIndication*** primitive from the WSO/RLSS, the CE shall generate and send the ***SubscriptionRequest*** message to the CM that serves this CE and wait for the ***SubscriptionResponse*** message from the CM.

When generating the ***SubscriptionRequest*** message, the CE shall set the parameters of the ***CxMessage*** as shown in the table below.

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

The CE shall set the parameters of the ***subscriptionRequest*** payload as shown in the table below.

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***clientID*** | ***IA5String*** | *This parameter is not used.* |
| ***clientPassword*** | ***IA5String*** | *This parameter is not used.* |
| ***coexistenceService*** | ***CoexistenceService*** | The value of the ***coexistenceService*** parameter from the received ***CxMediaSubscriptionIndication*** primitive. |

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

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

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

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***serverID*** | ***IA5String*** | *This parameter is not used.* |
| ***serverPassword*** | ***IA5String*** | *This parameter is not used.* |
| ***status*** | ***Status*** | ***noError*** |

After the CE has received the ***SubscriptionResponse*** message from the CM, the CE shall generate and send the ***CxMediaSubscriptionConfirm*** primitive to the WSO/RLSS.

The CE shall set the parameters of the ***CxMediaSubscriptionConfirm*** primitive as shown in the table below.

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***status*** | ***Status*** | The value of the ***status*** parameter from the received ***SubscriptionResponse*** message. |

# CM stop operation

After the CE has received the ***StopOperationAnnouncement*** message from its serving CM, the CE shall perform the **CM stop operation procedure**, described in (reference).

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

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

After the CE has received the ***StopOperationAnnouncement*** message from its serving CM, the CE shall generate and send the ***StopOperationConfirm*** message to the CM.

When generating the ***StopOperationConfirm*** message, the CE shall set the parameters of the ***CxMessage*** as shown in the table below.

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

The CE shall set the parameters of the ***stopOperationConfirm*** payload as shown in the table below.

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

After the CE has sent the ***StopOperationConfirm*** message to its serving CM, the CE shall check whether this has been the last available CM. If there is any available CM, the CE shall select one of the available CMs and initiate the start-up sequence with the selected CM.

If the CM has been the last available CM, the CE shall generate and send the ***CxMediaStopOperationIndication*** primitive to its WSO/RLSS and shall wait for the ***CxMediaStopOperationConfirm*** primitive from the WSO/RLSS.

Table below shows expected values of the parameters in the ***CxMediaStopOperationConfirm*** primitive.

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

After the CE has received the ***CxMediaStopOperationConfirm*** primitive from its WSO/RLSS, the CE shall stop operation.

# CE stop operation

After the CE has received a request to stop operation, the CE shall perform the **CE stop operation procedure**, described in (reference).

First the CE shall generate and send the ***CxMediaStopOperationIndication*** primitive to its WSO/RLSS and shall wait for the ***CxMediaStopOperationConfirm*** primitive from the WSO/RLSS.

Table below shows expected values of the parameters in the ***CxMediaStopOperationConfirm*** primitive.

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

After the CE has received the ***CxMediaStopOperationConfirm*** primitive from its WSO/RLSS, the CE shall generate and send the ***StopOperationAnnouncement*** message to the CM and shall wait for the ***StopOperationConfirm*** message from the CM.

When generating the ***StopOperationAnnouncement*** message, the CE shall set the parameters of the ***CxMessage*** as shown in the table below.

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

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

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

The CE shall set the parameters of the ***stopOperationConfirm*** payload as shown in the table below.

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

# RLSS stop operation

After the CE has received the ***CxMediaSubscriptionIndication*** primitive from its RLSS indicating that RLSS is going to stop operation, the CE shall perform the **WSO subscription update procedure**, described in (reference).

Table below shows expected values of the parameters in the ***CxMediaSubscriptionIndication*** primitive.

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***coexistenceService*** | ***CoexistenceService*** | ***noService*** |

After the CE has received the ***CxMediaSubscriptionIndication*** primitive from the RLSS, the CE shall generate and send the ***SubscriptionRequest*** message to the CM that serves this CE and wait for the ***SubscriptionResponse*** message from the CM.

When generating the ***SubscriptionRequest*** message, the CE shall set the parameters of the ***CxMessage*** as shown in the table below.

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

The CE shall set the parameters of the ***subscriptionRequest*** payload as shown in the table below.

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***clientID*** | ***IA5String*** | *This parameter is not used.* |
| ***clientPassword*** | ***IA5String*** | *This parameter is not used.* |
| ***coexistenceService*** | ***CoexistenceService*** | ***noService*** |

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

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

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

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***serverID*** | ***IA5String*** | *This parameter is not used.* |
| ***serverPassword*** | ***IA5String*** | *This parameter is not used.* |
| ***status*** | ***Status*** | ***noError*** |

After the CE has received the ***SubscriptionResponse*** message from the CM, the CE shall generate and send the ***CxMediaSubscriptionConfirm*** primitive to the WSO/RLSS.

The CE shall set the parameters of the ***CxMediaSubscriptionConfirm*** primitive as shown in the table below.

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***status*** | ***Status*** | The value of the ***status*** parameter from the received ***SubscriptionResponse*** message. |

# WSO stop operation

After the CE has received the ***CxMediaRegistrationIndication*** primitive from its WSO/RLSS indicating that this WSO or one or several WSOs served by this RLSS are stopping operation, the CE shall perform the **WSO registration update procedure**, described in (reference).

Table below shows expected values of the parameters of the each element of the sequence in the ***CxMediaRegistrationIndication*** primitive. The number of elements shall be equal to one for IEEE 802.22 CE and IEEE 802.11af CE. For the RLSS CE, the number of elements is equal to the number of WSOs that needs to stop operation by this procedure.

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***wsoID*** | ***OCTET STRING*** | WSO ID. |
| ***wsoStopOperation*** | ***BOOLEAN*** | ***TRUE*** |
| ***listOfAvailableFrequencies*** | ***ListOfAvailableFrequencies*** | *This parameter is not used.* |
| ***listOfOperatingFrequencies*** | ***ListOfOperatingFrequencies*** | *This parameter is not used.* |
| ***requiredResource*** | ***RequiredResource*** | *This parameter is not used.* |

After the CE has received the ***CxMediaRegistrationIndication*** primitive from the WSO/RLSS indicating that this WSO or one or several WSOs served by this RLSS are stopping operation, the CE shall generate and send the ***CERegistrationRequest*** message to the CM that serves this CE and wait for the ***RegistrationResponse*** message from the CM.

When generating the ***CERegistrationRequest*** message, the CE shall set the parameters of the ***CxMessage*** as shown in the table below.

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

The CE shall set the parameters of the each element of the sequence of the ***ceRegistrationRequest*** payload as shown in the table below. The number of elements shall be equal to the number of elements in the received ***CxMediaRegistrationIndication*** primitive.

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***operationCode*** | ***OperationCode*** | ***delete*** |
| ***wsoID*** | ***OCTET STRING*** | Value of ***wsoID*** parameter from the received ***CxMediaRegistrationIndication*** primitive. |
| ***networkID*** | ***OCTET STRING*** | *This parameter is not used.* |
| ***networkTechnology*** | ***NetworkTechnology*** | *This parameter is not used.* |
| ***geolocation*** | ***Geolocation*** | *This parameter is not used.* |
| ***coverageArea*** | ***CoverageArea*** | *This parameter is not used.* |
| ***installationParameters*** | ***InstallationParameters*** | *This parameter is not used.* |
| ***listOfAvailableFrequencies*** | ***ListOfAvailableFrequencies*** | *This parameter is not used.* |
| ***txScheduleSupported*** | ***BOOLEAN*** | *This parameter is not used.* |
| ***listOfOperatingFrequencies*** | ***ListOfOperatingFrequencies*** | *This parameter is not used.* |
| ***requiredResource*** | ***RequiredResource*** | *This parameter is not used.* |

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

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

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

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

After the CE has received the ***RegistrationResponse*** message from the CM, the CE shall generate and send the ***CxMediaRegistrationConfirm*** primitive to the WSO/RLSS.

The CE shall set the parameters of the ***CxMediaRegistrationConfirm*** primitive as shown in the table below.

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***status*** | ***Status*** | Value of ***status*** parameter from the received ***RegistrationResponse*** message. |