IEEE P802.19
Wireless Coexistence

|  |
| --- |
| Proposed entities common operation |
| Date: 2013-01-14 |
| 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 entities common 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 Entities Common Operation using the text below.*

# Entities Operation

# Common Operation

# Transport SAP operation

# General operation

The COEX\_TR\_SAP provides means for a CE, a CM, and a CDIS to communicate with each other and with external entities by using underlying transport service.

The transport service has the following functions:

* Address resolution
	+ Providing mapping between ***CxID*** parameter and IP address and port number
* Connection management
	+ Establishing connections to other entities and managing these connections
	+ Accepting connection requests from other entities and notifying them to the coexistence protocol
	+ Detecting disconnection and notify it to the coexistence protocol
	+ Apply security options
* Sending and receiving coexistence protocol messages
	+ Sending coexistence protocol messages to the entities specified by ***CxID*** parameter
	+ Receiving coexistence protocol messages from other entities and notifying them to the coexistence protocol.

Figure 19 shows general operation of the COEX\_TR\_SAP.



Figure 19. General operation of the COEX\_TR\_SAP.

# Transport service configuration

This procedure is not always used before the connection establishment procedure.

If the source entity is establishing connection with destination entity with which it has not had any prior communication and the source entity has destination entity coexistence protocol ID, IP address, and port number, then the transport service configuration procedure is used.

This refers to the following connection establishment cases:

* From CE to CM
* From CM to CDIS
* From CM to TVWS DB
* From CM to other CM
* From CDIS to other CDIS.

If the source entity is establishing connection with destination entity with which it has had prior communication and transport service of the source entity has destination entity coexistence protocol ID, IP address, and port number from prior communication, then the transport service configuration procedure is not used.

This refers to the following connection establishment cases:

* From CM to CE
* From CDIS to CM
* From TVWS DB to CM.

Figure 20 shows COEX\_TR\_SAP operation during transport service configuration.



Figure 20. COEX\_TR\_SAP operation during transport service configuration.

The source entity shall generate ***TrConfigurationRequest*** primitive and send it to the source entity transport service using COEX\_TR\_SAP. When generating the ***TrConfigurationRequest*** primitive, the source entity shall set its parameters as shown in the table below.

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***sourceID*** | ***CxID*** | Source entity ID. |
| ***destinationID*** | ***CxID*** | Destination entity ID. |
| ***destinationIPAddress*** | ***OCTET STRING*** | Destination entity IP address. |
| ***destinationPortNumber*** | ***INTEGER*** | Destination entity port number. |

After the source entity has sent the ***TrConfigurationRequest*** primitive to its transport service, the source entity shall wait for the ***TrConfigurationResponse*** primitive from the transport service.

Table below shows expected parameters of the ***TrConfigurationResponse*** primitive.

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***destinationID*** | ***CxID*** | Value of the ***destinationID*** parameter from the sent ***TrConfigurationRequest*** primitive. |
| ***status*** | ***Status*** | ***noError*** |

# Connection establishment

Figure 21 shows COEX\_TR\_SAP operation during connection establishment.



Figure 21. COEX\_TR\_SAP operation during connection establishment.

The source entity shall generate ***TrConnectionRequest*** primitive and send it to the source entity transport service using its COEX\_TR\_SAP. When generating the ***TrConnectionRequest*** primitive, the source entity shall set its parameters as shown in the table below.

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***sourceID*** | ***CxID*** | Source entity ID. |
| ***destinationID*** | ***CxID*** | Destination entity ID. |

The source entity transport service maps the destination entity ID to the destination entity IP address and port number, generates transport packet, and sends it to the destination entity via the underlying layers.

After the destination entity transport service has received the transport packet containing connection establishment request, it generates and sends ***TrConnectionIndication*** primitive to the destination entity.

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

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***sourceID*** | ***CxID*** | Source entity ID from the received connection establishment request. |

After the destination entity has received the ***TrConnectionIndication*** primitive, it shall generate the ***TrConnectionResponse*** primitive and shall send it to its transport service. When generating the ***TrConnectionIndication*** primitive, the destination entity shall set its parameters as shown in the table below.

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***destinationID*** | ***CxID*** | Value of the ***sourceID*** parameter from the received ***TrConnectionIndication*** primitive. |
| ***status*** | ***Status*** | ***noError*** |

The destination entity transport service maps the destination entity ID to the destination entity IP address and port number, generates transport packet, and sends it to the destination entity via the underlying layers.

After the source entity transport service has received the transport packet containing connection establishment response, it generates and sends ***TrConnectionConfirm*** primitive to the source entity.

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

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***sourceID*** | ***CxID*** | Destination entity ID from the received connection establishment response. |
| ***status*** | ***Status*** | ***noError*** |

# Sending and receiving coexistence protocol messages

Figure 22 shows COEX\_TR\_SAP operation during sending and receiving a coexistence protocol message.



Figure 22. COEX\_TR\_SAP operation during sending and receiving a coexistence protocol message.

When a source entity is sending a coexistence protocol message to a destination entity, the source entity and destination entity shall do the following.

The source entity shall generate ***TrSendRequest*** primitive and send it to the source entity transport services using its COEX\_TR\_SAP. When generating the ***TrSendRequest*** primitive, the source entity shall set its parameters as shown in the table below.

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***destinationID*** | ***CxID*** | Destination entity ID. |
| ***cxMessage*** | ***OCTET STRING*** | Encoded coexistence protocol message. |

The destination entity transport service maps the destination entity ID to the destination entity IP address and port number, generates transport packet, and sends it to the destination entity via the underlying layers.

After the source entity transport service has being confirmed on the transmission status, it generates and sends ***TrSendConfirm*** primitive to the source entity.

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

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***destinationID*** | ***CxID*** | Destination entity ID.Value of the ***destinationID*** parameter from the received ***TrSendRequest*** primitive. |
| ***status*** | ***Status*** | ***noError*** |

After the destination entity transport service has received the transport packet containing the coexistence protocol message, it generates and sends ***TrReceiveIndication*** primitive to the destination entity.

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

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***sourceID*** | ***CxID*** | Source entity ID. |
| ***cxMessage*** | ***OCTET STRING*** | Encoded coexistence protocol message. |

# Connection termination

Two cases of connection termination are distinguished:

* Connection termination initiated by one of the entities
* Connection termination due to network disconnection.

Figure 23 shows COEX\_TR\_SAP operation during connection termination initiated by one of the entities.



Figure 23. COEX\_TR\_SAP operation during connection termination initiated by one of the entities.

When a source entity is terminating connection with a destination entity, the source entity and destination entity shall do the following.

The source entity shall generate ***TrDisonnectionRequest*** primitive and send it to the source entity transport services using its COEX\_TR\_SAP. When generating the ***TrDisconnectionRequest*** primitive, the source entity shall set its parameters as shown in the table below.

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***destinationID*** | ***CxID*** | Destination entity ID. |

The destination entity transport service maps the destination entity ID to the destination entity IP address and port number, generates transport packet, and sends it to the destination entity via the underlying layers.

After the destination entity transport service has received the transport packet containing connection termination request, it generates and sends ***TrDisconnectionIndication*** primitive to the destination entity.

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

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***sourceID*** | ***CxID*** | Source entity ID from the received connection termination request. |

After the destination entity has received the ***TrDisconnectionIndication*** primitive, it becomes aware that the connection with the source entity is terminated. Usually this will happen after the coexistence protocols of the source entity and destination entity have finalized communication.

Figure 24 shows COEX\_TR\_SAP operation during connection termination due to network disconnection.



Figure 24. COEX\_TR\_SAP operation during connection termination due to network disconnection.

When the transport service of the source entity has detected that connection is lost with the destination entity, it generates ***TrDisconnectionIndication*** primitive and sends it to the source entity.

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

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***sourceID*** | ***CxID*** | Destination entity ID with which the connection is lost. |

When the transport service of the destination entity has detected that connection is lost with the source entity, it generates ***TrDisconnectionIndication*** primitive and sends it to the destination entity.

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

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Data type* | *Value* |
| ***sourceID*** | ***CxID*** | Source entity ID with which the connection is lost. |

After the transport services have indicated to their coexistence protocols that the connection is lost, they are expected to try to re-establish the connection.

# Processing CxMessage header

The ***header*** parameter in the ***CxMessage*** is choice of the following options (reference to message definition):

* ***none***
* ***requestID***.

Before sending a message, a sending entity shall set the values of the ***header*** parameter of the ***CxMessage*** as described below.

If the sending entity is sending an announcement message, the sending entity shall set ***header*** equal to ***none***.

If the sending entity is sending a request message, the sending entity shall set ***header*** equal to ***requestID***, where the value of the ***requestID*** is set arbitrary (implementation dependent).

If the sending entity is sending a response message to a previously received request message, the sending entity shall set ***header*** equal to ***requestID*** and shall set the value of the ***requestID*** to the same value as in the received request message.

# Processing status parameter

An entity sending, for example, request or announcement message or primitive may be waiting for response or confirm message or primitive. Response and confirm messages and primitives have ***status*** parameter. This parameter is used to inform the sending entity about the result of the receiving and processing of the message or primitive that has been sent.

The ***status*** parameter is can take one of the following values (reference to data types definition):

* ***noError***
* ***rejected***
* ***invalidEntityStatus***
* ***invalidArgument***
* ***processFailure***
* ***networkFailure***.

If the value of the ***status*** parameter is equal to ***noError***, then the message or primitive that has been sent is successfully received and processed.

If the value of the ***status*** parameter is equal to ***rejected***, then the message or primitive that has been sent is received and processed, but it is rejected because the sending entity is not authorized to communicate with the receiving entity.

If the value of the ***status*** parameter is equal to ***invalidEntityStatus***, then the message or primitive that has been sent is received, but it is not processed because the receiving entity is not expecting this message or primitive according to its current status of the coexistence protocol.

If the value of the ***status*** parameter is equal to ***invalidArgument***, then the message or primitive that has been sent is received, but it is not processed successfully because it has invalid set of the parameters or their values.

If the value of the ***status*** parameter is equal to ***processFailure***, then the message or primitive that has been sent is received, but it is not processed because the receiving entity does not operate properly.

If the value of the ***status*** parameter is equal to ***networkFailure***, then the message or primitive that has been sent is not received because there is problem with network connection.

The way an entity receiving a message or primitive having the ***status*** parameter shall act according to different values of this parameter is implementation depended.