IEEE P802.19
Wireless Coexistence

|  |
| --- |
| A conflict handling proposal for distributed decision making |
| Date: 2012-11-14 |
| Author(s): |
| Name | Company | Address | Phone | email |
| Jari Junell | Nokia  | Otaniementie 19, 02150 Espoo, Finland | +358-718036575 | jari.junell@nokia.com |
| Mika Kasslin | Nokia | Otaniementie 19, 02150 Espoo, Finland | +358-718036294 | mika.kasslin@nokia.com |
| Lauri Laitinen | Nokia | Otaniementie 19, 02150 Espoo, Finland |  | Lauri.laitinen@nokia.com |

Abstract

This document is a submission to IEEE 802.19 TG1 about conflict handling procedure description to support the section 10.4.7.5. This is comment resolution proposal related to comment with CID126.

**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.

*Editorial instructions: Add the following text in section 6.3.15 of the DF3.02.*

## 6.3.15 Conflict handling procedures

### 6.3.15.1 Obtaining state information from another CM procedure

A CM, which is in state of coexistence decision making for a WSO which is represented by a CE which this CM serves, shall perform this procedure in order to find out whether other CMs are in state of coexistence decision making for this same WSO. An illustrative example of this procedure is shown in Figure 52. ConflictInquiryRequest and ConflictInquiryResponse messages are defined in clause 6.4.



Figure 52: Requesting information about possible concurrent decision making by another CM

In order to start the procedure a CM shall generate a ConflictInquiryRequest message and shall send this message to all CMs that are allowed to calculate resource allocations to the WSO. CM operations related to generating and sending a ConflictInquiryRequest message are specified in clause 8.

After another CM has received a ConflictInquiryRequest message from the CM, another CM shall process the ConflictInquiryRequest message. CM operations related to processing a ConflictInquiryRequest message are specified in clause 8.

After the CM has received a ConflictInquiryResponse message from another CM, the CM shall process the ConflictInquiryResponse message. CM operations related to processing a ConflictInquiryResponse message are specified in clause 8.

### 6.3.15.2 Providing resource allocation ranking information to serving CM procedure

A CM shall perform this procedure in order to inform the serving CM about the ranking of other CMs resource allocations for the WSO. An illustrative example of this procedure is shown in Figure 53. ConflictResolutionRanking message is defined in clause 6.4.



Figure 53: Decision making ranking results to serving CM

In order to start the procedure a CM shall generate a ConflictResolutionRanking message and shall send this message to the serving CM. CM operations related to generating and sending a ConflictInquiryRequest message as well as serving CM operations related to processing a ConflictInquiryRequest message are specified in clause 8.

## 6.4 Messages

*Editorial instructions: Add the following messages in section 6.4 of the DF3.02. These messages are included also in our contribution (19-12/0199r0) about message and data type definitions.*

*--Conflict handling inquiry to another CM*

ConflictInquiryRequest ::= SEQUENCE {

 *--request source CM*

 cmID CxID,

 *-- request source CM IP address*

 sourceCMIPAddress OCTET STRING ,

 *-- request source CM port number*

 sourceCMPortNum OCTET STRING ,

 *--request destination CM of neighbor/extended neighbor WSO*

 cmID CxID,

 *-- CE identifier to which neighbor/extended neighbor WSO this message applies*

 neighborCeID CxID,

 *-- Network identifier of the neighbor/extended neighbor WSO, e.g., BSS ID*

 neighborNetworkID OCTET STRING,

 *-- CE identifier of target WSO to which suggested operating parameters apply*

 targetCeID CxID,

 *-- Network identifier of the target WSO, e.g., BSS ID to which suggested operating parameters apply*

 targetNetworkID OCTET STRING,

 *--Proposed operating parameters to the target WSO*

 operParamsTarget OperatingParameters

}

ConflictInquiryResponse ::= SEQUENCE {

 *--response source CM*

 cmID CxID,

 *-- response source CM IP address*

 sourceCMIPAddress OCTET STRING,

 *-- response source CM port number*

 sourceCMPortNum OCTET STRING,

 *-- CE identifier to which neighbor/extended neighbor WSO this message applies*

 neighborCeID CxID,

 *-- Network identifier of the neighbor/extended neighbor WSO, e.g., BSS ID*

 neighborNetworkID OCTET STRING,

 *--request source CM*

 cmID CxID,

 *-- CE identifier to which target WSO this message applies*

 targetCeID CxID,

 *-- Network identifier of the target WSO, e.g., BSS ID*

 targetNetworkID OCTET STRING,

 *--allocation status is “yes” if calculating resources to the target WSO, else “no”*

 allocationStatus BOOLEAN

}

ConflictResolutionRanking ::= SEQUENCE {

 *-- CE identifier to which target WSO this message applies*

 targetCeID CxID,

 *-- Network identifier of the target WSO, e.g., BSS ID*

 targetNetworkID OCTET STRING,

 *--Number of conflict responses to inquiry*

 conflictingResponses INTEGER,

 *--The ranking order of resource allocation proposals to target WSO as networkIDs from the best to worst and*

 *optionally also the ranking value included*

 rankOrder RankingOrder

}

## 6.5 Data types

*Editorial instructions: Add the following data type in section 6.5 of the DF3.02. These data types are included in our contribution related to finalizing section 6.5.*

RankingOrder ::= SEQUENCE {

 rankingNeighborNetworkID SEQUENCE OF OCTET STRING,

 rankingValue SEQUENCE OF INTEGER OPTIONAL

}