IEEE P802.19
Wireless Coexistence

|  |
| --- |
| A conflict handling proposal for distributed decision making |
| Date: 2012-04-24 |
| Author(s): |
| Name | Company | Address | Phone | email |
| Jari Junell | Nokia  | Itämerenkatu 9, 00180 Helsinki, Finland | +358-718036575 | jari.junell@nokia.com |
| Mika Kasslin | Nokia | Itämerenkatu 9, 00180 Helsinki, Finland | +358-718036294 | mika.kasslin@nokia.com |
| Päivi Ruuska | Nokia | Visiokatu 1, 33720 Tampere, Finland | +358-718035433 | paivi.m.ruuska@nokia.com |

Abstract

This document is a submission to IEEE 802.19 TG1 about conflict handling when two CMs apply distributed decision making and consequently there is a chance for the two CMs to conduct decision making concurrently.

This method resolves currently unsolved problem and is proposed to be added to DF 2.08.

**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.

The contribution shows a mechanism to solve a conflict situation of simultaneous resource allocations to the same WSO by at least two CMs that apply distributed decision making. The section contains some instructions to the technical editor to facilitate the editing work.

# 1 Text proposal for the candidate draft

*Editorial instruction: Have the following sub-section added to a correct location in sections 5 and 9 and update the sub-section numbering as appropriate.*

**9.X.X Conflict handling procedure**

**9.X.X.1 Introduction**

When two CMs, or multiple of them, apply distributed decision making they may concurrently be in process of allocating resources to a same WSO. This is a conflict in a resource allocation. It is a responsibility of a CM to check whether any other CM has calculated resources concurrently to any WSOs among a target WSO and its coexistence set. The following sub-sections define the procedure to handle this kind of conflict situation.

**9.X.X.2 Conflict handling**

Upon calculating a resource allocation proposal to a WSO a CM shall transmit a ConflictInquiryRequest to all other CMs that are allowed to calculate resource allocations to this WSO. The request shall contain an inquiry whether the other CM is calculating resources to the WSO. The request shall indicate also the CM’s own resource allocation proposal. The other CM shall send a ConflictInquiryResponse of either “yes” or “no”. If the response is “yes”, the responding CM becomes a conflict handling CM. Otherwise the responding CM shall not take part to the conflict handling process and it is not allowed to calculate resources to the WSO until a certain period has expired after the solution has been communicated.

The WSO for which the CM has calculated a resource allocation proposal may be a member of the first coexistence set and not under the CM itself. In that case there may be elements in the second coexistence set of that WSO that are not in the first coexistence set. The CM shall transmit a ConflictInquiryRequest via a serving CM of the WSO to those CMs managing the elements in the second coexistence set that address are unknown.

Upon transmitting a request to each of the CMs that are authorized to calculate resources to the WSO and upon receiving responses from all of them, the requesting CM knows, which are the conflict handling CMs. If there is no conflict handling CM, the requesting CM shall proceed to implement the resource allocation to the WSO either by ReconfigurationRequest if served by itself or CMReconfigurationrequest if served by another CM. The requesting CM shall communicate the allocation to all the relevant CMs by CoexistenceSetElement-InformationAnnouncement. Otherwise the requesting CM shall wait for conflict check requests from all the conflict handling CMs. The requesting CM shall reply to each of the requests with a “yes” response like the other CMs earlier indicated that they were in the process of calculating resources to the WSO.

Upon receiving a conflict check request from each other conflict handling CM with their resource allocation proposal, the requesting CM shall perform proposal ranking. In the proposal ranking the CM shall put the proposals from the other conflict handling CMs in order by giving the highest score (e.g. 5 in case of 5 proposals) to the “best” proposal, the second highest score (e.g. 4 in case of 5 proposals) and so on. The CM shall not consider its own proposal in the ranking but shall score only proposals from the other CMs. The CM shall transmit a ConflictResolutionRanking message with the ranking result (scores for all the proposals) to the CM that serves the WSO (i.e. serving CM) for which the allocation proposals have been ranked. The message shall indicate the number of conflict handling CMs to ensure that the service CM knows how many ranking messages to expect. Once the serving CM has received all the ranking result messages it shall determine which of the allocation proposals has received the highest total score. The serving CM may sum all the scores per proposal and use that as the total score of the proposal. The serving CM shall select the proposal that has the highest total score as the winning proposal. If there are multiple proposals with the highest total score, the CM shall pick up one of them as the winning proposal.

Upon determining the winning proposal, the serving CM shall communicate the selected allocation to the relevant CMs by CoexistenceSetElementInformationAnnouncement message and to the WSO itself by ReconfigurationRequest message in the manner it would do in case without conflicts. All the CMs having their WSOs in a coexistence set of the WSO and the serving CM itself shall reset a timer to specified value. They are not allowed to make any resource allocations to the WSO until the timer has reached zero. There is an exception for the serving CM to calculate a new resource allocation, i.e. the WSO is not able to operate anymore in the allocated frequency.

*Editorial instructions: Add the following text in section 5.2.X of the DF2.08.*

## 5.2.X Conflict handling procedures

### 5.2.X.1 Inquiry of concurrent decision making from another CM

This procedure, illustrated in Figure YY, is performed when a CM needs to know whether any other CM is in the process of making decisions on a WSO. The CM sends a ConflictInquiry\_Request to all the CMs that are allowed to calculate resource allocations to the WSO. Other CMs respond by transmitting a ConflictInquiry\_Response message.



Figure YY: Requesting information of possible concurrent decision making by another CM

### 5.2.X.2 Decision making ranking information to serving CM

A CM transmits a ConflictResolutionRanking message with the ranking result to the CM that serves the WSO (i.e. serving CM) for which the allocation proposals have been ranked.



Figure YY+1: Decision making ranking results of the target WSO to the CM which serves the WSO

## 5.3 Messages

*Editorial instructions: Add the following messages in section 5.3 of the DF2.08. These are also included in document 12/0051r0.*

*--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

}

## 5.4 Data types

*Editorial instructions: Add the following data type in section 5.4 of the DF2.08. These are also included in document 12/0051r0.*

RankingOrder ::= SEQUENCE {

 rankingNeighborNetworkID SEQUENCE OF OCTET STRING,

 RankingValue SEQUENCE OF INTEGER OPTIONAL

}