**3GPP TSG-SA5 Meeting #156S5-243545**

**19 - 23 August 2024, Maastricht, Netherlands**

**Source: Nokia**

**Title: Rel-19 pCR TR28.867 CCL conflicts resolution solution**

**Document for: Approval**

**Agenda Item: 6.19.4**

# 1 Decision/action requested

***The group is asked to discuss and agree on the proposal.***

# 2 References

[1] 3GPP TR 28.867: “Closed control loop management” v0.3.0

# 3 Rationale

The use cases on CCL conflicts resolution is a subset of CCL conflicts management. This pCR aggregates the two into a single use case on CCL conflicts management and provides a solution for resolution of direct parameter conflicts.

# 4 Detailed proposal

|  |
| --- |
| **First Change** |

5.6 Use case 6: CCL conflicts management

5.6.1.X Use case 9: CCL conflicts resolution

Multiple conflicts are possible among CCL or their instances. The CCL MnS producer should be able to interactively coordinate with MnS consumers to resolve the conflicts.

CCL Goal-conflicts resolution

The targets in the goals of Closed Control Loop should not contradict one another within that goal or contradict with other targets in goals of related CCLs, otherwise a goal conflict is observed. For such a goal conflict, goal coordination interactions are needed to resolve the conflict, i.e., to align goals (and related targets) that should be achieved by the various deployed Closed Control Loops. Given the potentially high number and diversity of Closed Control Loops, the process of setting and coordination goals for the Closed Control Loops should be accomplished using another CCL that consumes the CCL-related monitoring and governance services to coordinate the resolution of conflicts with the CCL.

The MnS producer for this CCL instance should inform the MnS consumer about a candidate goal conflict, e.g., about the values of the goal’s targets that are in conflict with the targets of another goal. In response, the MnS consumer could revise the goals of that CCL instance, terminate the execution of the CCL instance, delete the CCL instance,

CCL Trigger-time conflicts resolution

Typically, a CCL will be triggered to run at a specific time and terminate when certain conditions are met, to run when a certain performance threshold is crossed. If triggered independently, there may be conflicts among the CCLs. The triggers for different CCLs to be executed need to be coordinated to avoid conflicts among the CCLs. And in some instances, the conditions in the network may be such that it is not clear which CCL should be triggered, requiring to trigger multiple CCL in sequence. The triggering may be done by a coordination function that consumes the CCL-related monitoring and governance services to receive information with which to evaluate the conditions and determines which CCL to be triggered.

It may be the case that CCLs need to operate in a hierarchy with each CCL having an operational profile indicating the specific level of hierarchy. The operational profile describes characteristic sunder which the CCL operates, e.g., when or after which other CCLs, this CCL should be executed. For example, to ensure that handovers are always optimal, a CCL on handover optimization may need to be triggered every after a CCL on Energy saving has been executed to be sure that there are appropriate handover relations even when some cells may have been disabled. The MnS consumer that coordinates the execution times of the CCLs needs to configure the appropriate hierarchy for the CCLs. Using the operational profiles of the CCLs, , the MnS consumer evaluates the description of the third CCL against at least one of the profiles P1 and P2 and accordingly determines and configures the operational profile of the third CCL.

Note: A CCL may be involved in more than 1 hierarchies or within a single hierarchy, the CLL may relate to multiple other CCLS, which requires the hierarchies to be coordinated.

5.6.2 Potential Requirements

REQ-CCL-CONFLICT-1: The MnS Producer for CCL management should support a capability to detect a potential or actual conflict.

Note: A potential conflict is where some events are observed that indicate that there may be a conflict, but the CCL MnS Producer cannot conclude that it is a conflict. So, the CCL can indicate this so that some other entity e.g. the MnS consumer takes responsibility to confirm the conflict.

REQ-CCL-CONFLICT-2: The MnS Producer for CCL management should support a capability to inform an authorized MnS consumer about a potential conflict that has been detected.

REQ-CCL-CONFLICT-3: The MnS Producer for CCL management should support a capability to confirm a detected potential goal, action, indirect target, action execution time, scope conflict.

REQ-CCL-CONFLICT-4: The MnS Producer for CCL management should support a capability to resolve a goal, action, indirect target, action execution time, scope conflict that has been detected.

REQ-CCL-CONFLICT-5: The MnS Producer for CCL management should enable authorized MnS consumers to provide information that can be used to avoid the conflict.

REQ-CCL-CONFLICT-6: The MnS Producer for CCL management should enable authorized MnS consumers to provide information that can be used to resolve the conflict.

REQ-CCL-CONF\_RES-1: The MnS producer should support a capability to coordinate the resolution of conflicts on the CCLs goals.

REQ-CCL-CONF\_RES-2: The MnS producer should support a capability to coordinate the resolution of conflicts on the triggers for execution of the CCL instances.

REQ-CCL-CONF\_RES-3: The MnS producer should support a capability enabling an MnS consumer to define and coordinate the hierarchies of the CCL.

5.6.3 Potential Solutions

5.6.3.K Potential Solution K: bargaining as resolution of direct actions conflicts

Note: This solution focusses on the requirement on

* Avoidance of potential direct actions conflicts

5.6.3.K.1 Required capabilities and interactions.

The simplest way to resolve direct parameter conflicts is to separate the control spaces of the CCLs, i.e. to allocate each parameter to a specific CCL. However, this is not guaranteed to always be possible, i.e., in some cases two or more CCLs may want to set different values for the same parameter and the parameter cannot be assigned to only one CCL. In these cases, a coordinator functionality, e.g. a coordinator CCL should compute a compromise value for the parameter, a value which can be considered to be equally good for all the CCLs. However, since different CCLs have different goals, it is necessary for the coordinator CCL to understand the importance of the parameter to each CCL. For this purpose, the CCLs provide their usefulness for the parameter to the coordinator CCL. The usefulness provided by a CCL shows the relative goodness of different values of the parameter to the CCL in a pre-defined scale, e.g., [0:1]. Since all the CCLs used the same scale, when the CCL coordinator selects a parameter value, it can clearly understand how important this value is for each CCL. The CCL coordinator can then derive the compromise values which is then (provided to the CCLs to be) executed onto the managed object. An example way to compute the compromise is to use the Nash Social Welfare Function since it provides equal fairness to all competing entities.

A compromise based only on usefulness does not consider the relative (level of) interest of the CCLs in the parameter. To account for the interests, the CCLs should provide to the CCL coordinator their relative interest in the parameter, so that the computed compromise value accounts for the combined interests of the CCLs. The relative interest may be computed based on a fixed scale. For example, for a CCL on cell interference management on a scale of [0-10], a cell’s transmit power has a goodness of say 9 than the cells load which has a goodness of 3.

Note 1: the coordination CCL does not have to calculate the compromise value all the time as this requires information exchange among the CCLs and computational energy. It should be possible to configure the coordination CCL such that it calculates the compromise values only when certain conditions are met. The coordination CCL should be able to expose required services to the MnS consumer to configure such conditions.

Note 2: For a given CCL, the usefulness may be equivalent to the level of interest but it is not always the case. It is possible that a CCL has high interest in a parameter that has low usefulness.

5.6.3.K.2 Information objects to realize required capabilities and interactions

* Introduce for each control parameter of a CCL, an attribute presenting the usefullness of that parameter. The usefullness may be called parameterUsefullness , indicates the utility of different values of the parameter to the CCL.
	+ The parameterUsefullness should be notifiable, so that when the CCL sends its action plan, it can notify the coordinator CCL of the parameterUsefullness.
* Introduce for each control parameter of a CCL, an attribute representing the compromise computed by the coordinator CCL .
* Introduce for each control parameter of a CCL, an attribute presenting the degree of interest of the CCL in that parameter. The interest may be called parameterInterestLevel, indicates the CCL’s level on interest in the parameter. For a given CCL, the usefulness may be equivalent to the level of interest but it is not always the case. It is possible that a CCL has high interest in a parameter that has low usefulness.
	+ The parameterInterestLevel should be notifiable, so that when the CCL sends its action plan, it can notify the coordinator CCL of the parameterInterestLevel.

5.6.4 Evaluation of solutions

The potential solution described in clause 5.8.3.K is a fully NRM-based approach that extends the existing NRM to enable resolution of direct parameter conflicts through bargaining. The solution allows CCLs to provide to the coordination CCL their utility functions and degree of interest in the different parameters they want to control. The coordination CCL can then compute compromise values which are provided to the CCLs for execution on to the network. Therefore, the solution described in clause 5.8.3.K is a feasible solution for resolution of direct parameter conflicts.

# 6. Conclusions and Recommendations

It is recommended to move on to the normative specification development phase for the use case on

* bargaining for the resolution of direct parameter conflicts, the normative specification development should follow the solution outlined in clause 5.8.3.K