**3GPP TSG-SA5 Meeting #156S5-243542**

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

**Source: Nokia**

**Title: Rel-19 pCR TR28.867 solution for direct actions conflicts**

**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 management describes action conflicts as one of the conflicts that need to be managed. This pCR is to add a solution for managing action conflicts

# 4 Detailed proposal

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

5.6 Use case 6: CCL conflicts management

5.6.1 Description

Multiple CCLs could co-exist and concurrently act within the same environment. The CCLs can affect one another, in the worst cases leading to conflicts. The different kinds of conflicts are summarized by Table 5.6.1-1.

5.6.1-1: Types of potential conflicts among CCL instances for goals g1, g2 and g3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Conflict Type | Description | CCL-A | CCL-B | Comments |
| Target Conflict | For CCLs C1 and C2, when same at least 1 target of a goal is present in both CCL asking for different outcomes on that target on same controlled entity (ME1). | Control Scope: ME1Goal targets: * Load > 90% (to maximize resource utilization)
* latency < 10ms
 | Control Scope: ME1Goal target: * Load < 90% (to avoid congestion)
 | Conflict among the targets within the goals - due to different required target outcomes |
| Action Conflict | For CCLs C1 and C2, when both C1 and C2 is trying to configure the same characteristics of same target entity (gNB-g1) in contradiction. | **Example 1** | Conflict due to configuration actions at execution step because both CCL want different contradicting value for a particular characteristic of gNB-g1.Effect: even when executed at different times, the value may ping-pong continuously. |
| Goals target:* Throughput > 10gbps

Actions: * Target Entity: gNB-g1
* Target Change: scale-out virtual resource
 | Goals target:* EC is < 10KVA

Actions: * Target Entity: gNB-g1
* Target Change: scale-in virtual resource
 |
| **Example 2** |
| Goal target: * HO failure is < 2%

Actions: * Target Entity: gNB-g1
* Target Change: set CIO to a small **positive** value{to guarantee HOs with low chances of HO failure}
 | Goal target: * Load < 80%

Actions: * Target Entity: gNB-g1
* Target Change: set CIO to a small negative value [to advance HOs and move load to other cells]
 |
| Indirect target conflict | For CCLs C1 and C2, when C1 [optimize handover] and C2 [minimize interference] have different goals but the actions of C1 affect the goals of C2  | Goal target: * HO failure is < 2%

Actions: * Target Entity: gNB-g1
* Target Change: reduce CIO {to reduce chances of HO failure}
 | Goal target: * SINR > 10dB

Actions: * Target Entity: gNB-g1
* Target Change: lower antenna tilt
 | By reducing antenna tilt to minimize interference C2 affect the HO goal target of C1 |
| Action Execution Time Conflict | For CCLs C1 and C2, when both C1 and C2 are trying to configure the same characteristics of same target entity (gNB-g1) in contradiction. | Goals:* Throughput > 10gbps

Actions: * Target Entity: gNB-g1
* Target Change: scale-out
* Target Time: 04:00
 | Goals:* EC is < 10KVA

Actions: * Target Entity: gNB-g1
* Target Change: scale-in
* Target Time: 04:00
 | Conflict due to the time of executing the configuration actions at the execution step  |
| Scope conflict | For CCLs C1 and C2, C1 and C2 have different goals and actions but their scopes are overlapping – e.g. C1’s control scope (i.e. the controlled entities in the network) is part of C2’s measurement scope (i.e. the measured entities in the network) | Measurement scope: cells g1 Control Scope: g1Goal targets: * EC/bit is < 1WA

Actions: * Target Entity: gNB-g2
* Target Change: switch off g2
 | Measurement scope: cells g1, g2, g3, g4Control Scope: g2Goals: * Load < 80%

Actions: * Target Entity: gNB-g2
* Target Change: change CIO
 | By switching off g2, C1 affects the scope which C2 reads for its load distribution measurements |

The CCL may detect or observe events that identify the possibility of any one of the above conflicts. The conflict can be avoided using some information or the policies (e.g., priority) provided by the consumer. If the conflict actually occurs, the CCL MnS producer should support services to inform MnS consumers the confirmed detected conflicts. This may also include informing MnS consumer about the potential conflict.

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.

5.6.3 Potential SolutionsNote: This solution focusses on the requirement on

* detection and avoidance of potential direct actions conflicts
* detection and avoidance of actual direct actions conflicts

#### 6.6.3.A Potential Solution x: Detection and avoidance of actions conflicts

5.6.3.A.1 Required capabilities and interactions.

If two CCLs execute their actions within the same time period, the actions could cause undesirable effects, e.g., by conflicting for the same parameter on the managed entities. Before the interacting CCLs execute their actions, the Pre-execution coordination can be used to detect potential direct-actions conflicts.

To detect and avoid potential direct-actions conflicts:

* a CCL intending to take an action, sends its proposed configuration management changes to the coordination CCL prior to execution of those CM plans. The CM plans contain information of target resources and scheduled time for execution.
* The coordination CCL checks the submitted CM plans against other previous CM plans from other CCLs (that have been executed) to see if there are any potential conflicting actions based on the provided information. This ensures to check planned CM changes against actions that have already been executed.

To avoid potential direct-actions conflicts:

* The coordination CCL notifies the detected conflict(s) to the related CCLs
* The CCL may adjust its planned configurations to a new set that could have less conflicts

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

* Introduce a datatype, and related attribute on the CCL, representing the CCL desired changes, say called CMPlan. The CMPlan, , indicates the objects that are planned to be configured, the attributes on those objects that would be configured, the values to which they would be configured and the time at which those plans are expected to be executed. The CMPlan should be notifiable, the coordination CCL is notified by the CCL when the action has been drawn up
* Introduce an attribute on the CCL , say called detectedConflict, representing a conflict for a given action in the plannedAction. The detectedConflict may be a Boolean flag which is by default FALSE but is toggled to TRUE when a conflicts is detected. It may also be pair which adds information about the other actions to which the said action conflicts.

Note: After the potential conflict is detected, a different solution is needed to resolve those conflicts e.g. using priorities among CCLs.

#### 6.6.3.B Potential Solution x: Detecting actual conflicts based on counter-productiveness

5.6.3.B.1 Required capabilities and interactions.

In certain cases, two CCLs may work together on the same managed entity, maybe at different times scales or involving different aspects/sub functionalities of the managed entity. However, there may be some known or unknown interdependence between actions taken by the two CCLs, E.g., in the cases where the scopes of the two CCLs cannot be separated. Furthermore, two CCLs may change the same parameter one after the other. For multi-aspect optimization, such interdependence is often expected, and it should be tolerated by the system as long as it is not harmful in terms of the overall performance of the managed entity.

A way to detect actual conflicts and minimize their impacts is detect counter-productiveness., CCLs that operate on the same managed object monitor any counter-productiveness and if observed, maintain it within some tolerance limits.

For this a CCL instance A that is likely to be affected, needs to monitor a specific scope or context that could be affected by another CCL instance B. CCL B can provide a conflict monitoring context/ scope to CCL A informing CCL A about CCL B’s latest actions on the managed entity and its tolerance w.r.t to its parameters and metrics in this managed entity. CCL A (the context recipient CCL) should work within these bounds, i.e., its actions shall not violate the said tolerances to avoid counter-productiveness. CCL A observes the conflict monitoring context, so that if it observes the violations of the said tolerances, it reports the conflict to the CCL B.

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

* Introduce an attribute on the CCL representing the scope that is being monitored for counter productiveness by the CCL, say called CCLMonitorScope. The CCLMonitorScope is notifiable, when an action is executed, the responsible CCL (acting as CCL B) updates the CCLMonitorScope and notifies the other CCLs (e.g. via the coordination CCL) about the current CCLMonitorScope.
	+ Introduce an attribute in the CCLMonitorScope for the monitored parameters, say called monitoredScopeParameters with corresponding tolerance for each monitored parameter, say called monitoredScopeParameterTolerance. The monitoredScopeParameters and their monitoredScopeParameterTolerance are notifiable, notified together with the CCLMonitorScope notification.

5.6.4 Evaluation of solutions

The potential solution described in clause 5.8.3 is a fully NRM-based approach that extends the existing NRM to realise detection of CCL actions conflicts and avoidance of counter productiveness even where there are no concrete conflicts. The solutions make small additions with a few attributes that enable to exchange information when conditions of counter-productiveness are observed, or conflicts detected. Therefore, the solution described in clause 5.6.3 is a feasible solution for Detection of action plan conflicts and avoidance of counter-productiveness.

|  |
| --- |
| **Next Change** |

# 6. Conclusions and Recommendations

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

* detection of CCL actions conflicts, the normative specification development should follow the solution outlined in clause 5.8.3.A
* avoidance of counter-productive CCL actions, the normative specification development should follow the solution outlined in clause 5.8.3.B