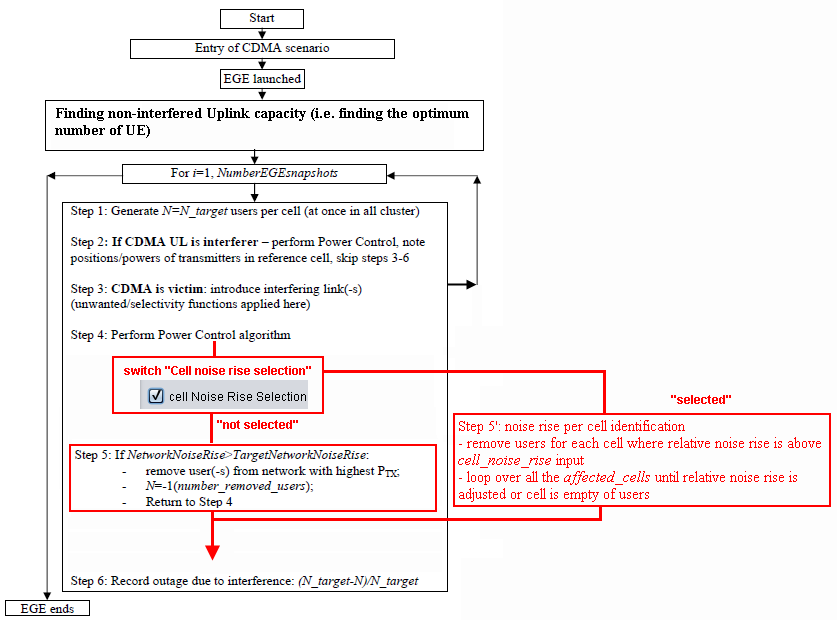
**STG(12)26**

**CDMA UL – enhancement to allow cell selection**

# 1- Overview



**Figure 1: Overview of the “new” implementation (in red) with respect to the “old” CDMA UL (in black)**

Note: The “old” algorithm (i.e. removal of UE with highest PTx in the network) is untouched (i.e. same results as for the official version).

# 2- Inputs

Two new input have been created

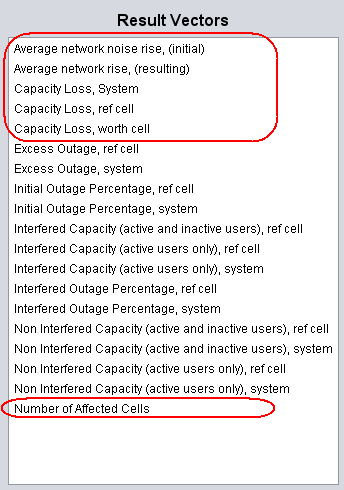
* *Cell noise rise selection*: It selects the new algorithm (type: Switch). By default it is “false” (i.e. not selected)
* *Target cell noise rise* (type: double). It is only available when *Cell noise rise selection* is selected

|  |  |
| --- | --- |
|  |  |
| **Figure 2: Example when “old” algorithm is activated** | **Figure 3: Example when “new” algorithm is activated** |

# 3- Outputs

The following new output vectors are now available:

* Average network noise rise (initial): value at step 5 (i.e. before the algorithm)
* Average network noise rise (resulting): value at step 10 (i.e. after the algorithm)
* The CDF of the capacity loss in the whole network (for each event, calculate the capacity loss in %).
* The CDF of the capacity loss in the reference cell per event
* The CDF of the capacity loss in the worst cell per event (the first strongest cell: selectedCell[1]). The cell-ID can be different from event to event but the capacity loss is to be extracted).
* The number of cells affected per event.



# 4- Algorithm implemented 29 June 2012

Algorithm agreed at the STG web meeting 25 June 2012

Convention: *i* is the index of the cells

1. Balance the power of the UE (i.e. set their Tx power) (*internalPowerBalance()*), the reference network capacity is calculated with the average noise rise reaching the defined threshold without external interference (i.e. *cellNoiseRiseInitial[i]* )
2. Calculate the average noise rise (with external interference) in the whole network and the noise rise in each cell (i.e. *cellNoiseRiseInterferer[i])*
3. Arrange the active users in a list so that they are sorted based on their Tx power (i.e. *activeUsers*)
4. Calculate the relative cell noise rise which is the difference between noise rise with and without external interference for that cell.

(i.e. *relativeCellNoiseRise[i] = - cellNoiseRiseInitial[i] + cellNoiseRiseInterferer[i])*

1. If the network average noise rise is above a defined threshold (i.e. *targetNetworkNoiseRise* input to SEAMCAT, usually between 5.5 dB or 6dB) then start the process of dropping users.
2. Loop to identify the affected cells based on *relativeCellNoiseRise[i]*

If - *cellNoiseRiseInitial[i] + cellNoiseRiseInterferer[i] > TargetCellNoiseRise* (in dB) then Cell[i] is considered affected

A list of affected cells is created (i.e. *selectedCell*)

1. Selection of the algorithm: switch made by the SEAMCAT user
2. Sort the list selectedCell so that the cell with the highest relative noise rise is the first element
3. Start loop over the sorted list selectedCells (starting from the first element)
   1. Extract the active users of *selectedCell[i]* from *activeUsers* into a list *selectedCellActiveUsers (one list per cell)*
   2. Calculate the current new noise rise for *selectedCell[i]* (i.e. *currentCellNoiseRise[i]*).
   3. Start removing (i.e. dropping) the active users from *selectedCellActiveUsers (by default the ActiveUsers list is reduced)*. The UE with strongest Tx power is removed first.
   4. Proceed with *internalPowerBalance()* over the remaining users for the whole
   5. Calculate the new noise rise for *selectedCell[i]* (i.e. *currentCellNoiseRise[i]*).

If *currentCellNoiseRise[i] >cellNoiseRiseInitial[i]*, continue until there is no more users or the cell noise rise is below or equal *cellNoiseRiseInitial[i]*.

Calculate the average network noise rise for output processing.

If average noise rise below the *targetNetworkNoiseRise* exit the loop.

1. Outside the loop
2. Produce:

a)The CDF of the capacity loss in the whole network (for each event, calculate the capacity loss in %).

b) The CDF of the capacity loss in the worst cellper event (the first strongest cell: *selectedCell[1]*). The cell-ID can be different from event to event but the capacity loss is to be extracted).

c) The number of cells affected per event.

d) Get vector of average network noise rise

# 4- Testing GUI

Tester: Jean-Philippe Kermoal (ECO)

|  |  |  |
| --- | --- | --- |
| **GUI** | **procedure** | **Status** |
| All inputs | load | pass |
|  | save | pass |
|  | Used in the model (i.e. calculation) | pass |
| All output vectors | Load | pass |
|  | Save | pass |
|  | Used in the model | pass |

# 5- Source code

**Algorithm:** CDMAUplinkSystem.java,

* main implementation done in balanceInterferedSystem()
* see also createAffectedCellList() and extractActiveUsersToSelectedCellList()

**GUI:** CdmaUplinkSettingsPanel.java