**Doc xx**

**Minutes of STG web meeting 25 June 2012**

Participants:

Jean-Philippe Kermoal (ECO – chair), Karl Koch (BnetzA), Jörgen Karlsson (Ericsson), Stephen Harding (Ofcom-UK), Yannick LI (France Telecom Orange), Stefan Hiensch (BNetzA), Martin Deutsch (BNetzA), Paul Deedman (Inmarsat), Mario Neri (Inmarsat) and Luzango Pangani Mfupe (Tshwane University of Technology)

Output document:

**Draft algorithm CDMA UL algorithm**

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] > 1 dB* then Cell[i] is considered affected

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

***[Editor’s note: STG decided that the 1 dB to be an input parameters]***

1. Selection of the algorithm

*If step 5 = true*

*AND step 6 = true -> use new algo (as described here)*

*AND step 6 = false OR optimisation -> use old algo*

*If step 5 = false*

 *Leave the loop*

***[editor’s note: input on the optimisation implementation needed by the 27 June so far optimization = “OR more than 7-1(omni) or 21-3 (tri-sector) affected”]***

1. Sort the list *selectedCell* so that the cell with the highest relative noise rise is the first element
2. 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 5.5/6dB exit the loop.

1. 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

Action Point:

* ECO to start the implementation of the above algorithm. Completion deadline estimated 29 June 2012. To be released as Beta 1 for testing.
* input of implementation needed by the 27 June on the optimisation of step 7

Next web meeting:

* The next webmeeting will be set up as soon as the Beta version is ready for testing.