3GPP TSG RAN WG1 Meeting #9
Dresden, Germany, 30 Nov - 3 Dec 1999

Document R1-99h80
e.g. for 3GPP use the format TP-99xxx
or for SMG, use the format P-99-xxx

CHANGE REQUEST

Please see embedded help file at the bottom of this page for instructions on how to fill in this form correctly.

25.213	**CR**	**005**	Current Version: v3.0.0

GSM (AA.BB) or 3G (AA.BBB) specification number ↑ *↑ CR number as allocated by MCC support team*

For submission to:	RAN #6	for approval	X	strategic		*(for SMG*
list expected approval meeting # here ↑		for information		non-strategic		*use only)*

Form: CR cover sheet, version 2 for 3GPP and SMG *The latest version of this form is available from: ftp://ftp.3gpp.org/Information/CR-Form-v2.doc*

Proposed change affects: *(at least one should be marked with an X)*	(U)SIM		ME	X	UTRAN / Radio	X	Core Network

Source: Nokia **Date:** 1 Nov 1999

Subject: Harmonization of notations scrambling codes

Work item:

Category:				**Release:**		
	F	Correction			Phase 2	
	A	Corresponds to a correction in an earlier release			Release 96	
(only one category	B	Addition of feature			Release 97	
shall be marked	C	Functional modification of feature			Release 98	
with an X)	D	Editorial modification	X		Release 99	X
					Release 00	

Reason for change: The current text in 4.3.2.1 is misleading, notation is clarified.
To improve the quality of the definition of uplink long scrambling codes in 4.3.2.2.
5.2.2 is updated in line with changes in 4.3.2.1 and 4.3.2.2

Clauses affected: 4.3.2.1, 4.3.2.2 and 5.2.2 of TS25.213

Other specs affected:			
	Other 3G core specifications		→ List of CRs:
	Other GSM core specifications		→ List of CRs:
	MS test specifications		→ List of CRs:
	BSS test specifications		→ List of CRs:
	O&M specifications		→ List of CRs:

Other comments:

<--------- double-click here for help and instructions on how to create a CR.

4.3.2 Scrambling codes

4.3.2.1 General

~~There are 2^{24} uplink scrambling codes.~~ All uplink channels shall use either short or long complex valued scrambling codes, indicated by higher layers. ~~except for the~~ However, PRACH~~, for which~~ uses only the long scrambling code.~~ is used. Both short and long scrambling codes are represented with complex-value.~~

The uplink scrambling generator ~~(either short or long)~~ shall be initialised by a 24-~~-~~bit value, resulting in 2^{24} uplink scrambling codes of each type.

The n:th uplink scrambling code, denoted by $C_{scramb,\,n}$, is defined as either short

$$C_{scramb,n}(i) = \begin{cases} c_{1,n}(i \bmod 256)(1 + jc_{2,n}(i \bmod 256)) & if\ i\ is\ even \\ c_{1,n}(i \bmod 256)(1 - jc_{2,n}((i-1) \bmod 256)) & if\ i\ is\ odd \end{cases}$$

or long

$$C_{scramb,n}(i) = \begin{cases} c_{1,n}(i)(1 + jc_{2,n}(i)) & if\ i\ is\ even \\ c_{1,n}(i)(1 - jc_{2,n}(i-1)) & if\ i\ is\ odd \end{cases}$$

where i = 0, 1, 2, …, and the lowest index corresponds to the chip transmitted first in time.

The constituent codes $c_{1,n}$ and $c_{2,n}$ are formed differently for the short and long scrambling codes as described in Sections 4.3.2.2 and 4.3.2.3.

Both short and long uplink scrambling codes for dedicated channels are formed as follows:

$S_{ul,n}(i) = C_{scramb,n}(i)$

Where i = 0, 1, 2, …, 38399, and i =0 corresponds to the chip transmitted first in time.

~~$C_{scramb,n} = c_1(w_0 + jc_2'w_1)$~~

~~where w_0 and w_1 are chip rate sequences defined as repetitions of:~~

~~$w_0 = \{1 \quad 1\}$~~

~~$w_1 = \{1 \quad -1\}$~~

~~Also, c_1 is a real chip rate code, and c_2' is a decimated version of the real chip rate code c_2.~~

~~With a decimation factor 2, c_2' is given as:~~

~~$c_2'(2k) = c_2'(2k+1) = c_2(2k), \quad k=0,1,2\ldots$~~

~~The constituent codes c_1 and c_2 are formed differently for the short and long scrambling codes as described in Sections 4.3.2.2 and 4.3.2.3.~~

4.3.2.2 Long scrambling code

The long scrambling codes are formed as described in Section 4.3.2, where $c_{1,n}$ and $c_{2,n}$ are constructed as the position wise modulo 2 sum of 38400 chip segments of two binary *m*-sequences generated by means of two generator polynomials of degree 25. Let *x*, and *y* be the two *m*-sequences respectively. The *x* sequence is constructed using the primitive (over GF(2)) polynomial $X^{25}+X^3+1$. The y sequence is constructed using the polynomial $X^{25}+X^3+X^2+X+1$. The resulting sequences thus constitute segments of a set of Gold sequences.

The code~~,~~ $c_{2,n}$~~, used in generating the quadrature component of the complex spreading code~~ is a 16,777,232 chip shifted version of the code~~,~~ $c_{1,n}$~~, used in generating the in phase component~~.

The uplink scrambling code word has a period of one radio frame.

Let $n_{23} \dots n_0$ be the 24 bit binary representation of the scrambling code number n (decimal) with n_0 being the least significant bit. The x sequence depends on the chosen scrambling code number n and is denoted x_n, in the sequel. Furthermore, let $x_n(i)$ and $y(i)$ denote the i:th symbol of the sequence x_n and y, respectively

The m-sequences x_n and y are constructed as:

Initial conditions:

$x_n(0)=n_0$, $x_n(1)= n_1$, … $=x_n(22)= n_{22}$,$x_n(23)= n_{23}$, $x_n(24)=1$

$y(0)=y(1)= \dots =y(23)= y(24)=1$

Recursive definition of subsequent symbols:

$x_n(i+25) =x_n(i+3) + x_n(i)$ *modulo 2*, $i=0,\dots, 2^{25}-27$,

$y(i+25) = y(i+3)+y(i+2) +y(i+1) +y(i)$ *modulo 2*, $i=0,\dots, 2^{25}-27$.

~~The definition of the n:th scrambling code word for the in phase and quadrature components follows as (the left most index correspond to the chip scrambled first in each radio frame):~~

Define

$z_n(i) = x_n(i)+y(i)$, $i = 0, 1, 2, \dots, 2^{25}-2$,

~~$c_{1,n} = < x_n(0)+y(0), x_n(1)+y(1), \dots, x_n(N-1)+y(N-1) >$,~~

~~$c_{2,n} = < x_n(M)+y(M), x_n(M+1)+y(M+1), \dots, x_n(M+N-1) + y(M+N-1) >$,~~

again all sums of symbols being modulo 2 ~~additions~~

~~Where N is the period in chips and M = 16,777,232.~~

The real valued Gold sequence Z_n is defined by

$$Z_n(i) = \begin{cases} +1 & if \;\; z_n(i) = 0 \\ -1 & if \;\; z_n(i) = 1 \end{cases} \quad for \;\; i = 0,1,\dots,2^{25}-2.$$

Now, the real valued codes $c_{1,n}$ and $c_{2,n}$ for the long scrambling are defined as follows:

$c_{1,n} = Z_n(i)$ for $i = 0, 1, 2, \dots, 2^{25} - 2$ and

$c_{2,n} = Z_n((i+16,777,232)$ *modulo* $(2^{25} - 1))$ for $i = 0, 1, 2, \dots, 2^{25} - 2$;

the lowest index corresponds to the chip applied first in time.

~~These binary code words are converted to real valued sequences by the transformation ‘0’ -> ‘+1’, ‘1’ -> ‘-1’.~~

In case the OVSF code on the PDSCH varies from frame to frame, the OVSF codes shall be allocated such a way that the OVSF code(s) below the smallest spreading factor will be from the branch of the code tree pointed by the smallest spreading factor used for the connection. This means that all the codes for UE for the PDSCH connection can be generated according to the OVSF code generation principle from smallest spreading factor code used by the UE on PDSCH.

In case of mapping the DSCH to multiple parallel PDSCHs, the same rule applies, but all of the branches identified by the multiple codes, corresponding to the smallest spreading factor, may be used for higher spreading factor allocation.

5.2.2 Scrambling code

A total of 2^{18}-1 = 262,143 scrambling codes, numbered 0…262,142 can be generated. However not all the scrambling codes are used. The scrambling codes are divided into 512 sets each of a primary scrambling code and 15 secondary scrambling codes.

The primary scrambling codes consist of scrambling codes n=16*i where i=0…511. The i:th set of secondary scrambling codes consists of scrambling codes 16*i+k, where k=1…15.

There is a one-to-one mapping between each primary scrambling code and 15 secondary scrambling codes in a set such that i:th primary scrambling code corresponds to i:th set of scrambling codes.

Hence, according to the above, scrambling codes k = 0, 1, …, 8191 are used. Each of these codes are associated with an even alternative scrambling code and an odd alternative scrambling code, that may be used for compressed frames. The even alternative scrambling code corresponding to scrambling code k is scrambling code number k + 8192, while the odd alternative scrambling code corresponding to scrambling code k is scrambling code number k + 16384.

The set of primary scrambling codes is further divided into 64 scrambling code groups, each consisting of 8 primary scrambling codes. The j:th scrambling code group consists of primary scrambling codes 16*8*j+16*k, where j=0..63 and k=0..7.

Each cell is allocated one and only one primary scrambling code. The primary CCPCH and primary CPICH are~~is~~ always transmitted using the primary scrambling code. The other downlink physical channels can be transmitted with either the primary scrambling code or a secondary scrambling code from the set associated with the primary scrambling code of the cell.

The mixture of primary scrambling code and secondary scrambling code for one CCTrCH is allowable.

The scrambling code sequences are constructed by combining two real sequences into a complex sequence. Each of the two real sequences are constructed as the position wise modulo 2 sum of 38400 chip segments of two binary *m*-sequences generated by means of two generator polynomials of degree 18. The resulting sequences thus constitute segments of a set of Gold sequences. The scrambling codes are repeated for every 10 ms radio frame. Let x and y be the two sequences respectively. The x sequence is constructed using the primitive (over GF(2)) polynomial $1+X^7+X^{18}$. The y sequence is constructed using the polynomial $1+X^5+X^7+ X^{10}+X^{18}$.

The sequence depending on the chosen scrambling code number n is denoted z_n, in the sequel. Furthermore, let $x(i)$, $y(i)$ and z_n~~(i)~~(i) denote the i:th symbol of the sequence x, y, and z_n, respectively

The *m*-sequences *x*and *y* are constructed as:

Initial conditions:

x *is constructed with* $x(0)=1, x(1)= x(2)=...= x(16)= x(17)=0$

$y(0)=y(1)= \ldots =y(16)= y(17)=1$

Recursive definition of subsequent symbols:

$x(i+18) =x(i+7) + x(i) \text{ modulo } 2, i=0,\ldots,2^{18}-20,$

$y(i+18) = y(i+10)+y(i+7)+y(i+5)+y(i) \text{ modulo } 2, i=0,\ldots, 2^{18}-20.$

The n:th Gold code sequence z_n, $n=0,1,2,\ldots,2^{18}-2$, is then defined as

$z_n(i) = x((i+n) \text{ modulo } (2^{18} - 1\cancel{2}) + y(i) \text{ modulo } 2,\ i=0,\ldots, 2^{18}-2.$

These binary sequences ~~code words~~ are converted to real valued sequences Z_n by the following transformation: ~~'0' -> '+1', '1' -> '-1'.~~

$$Z_n(i) = \begin{cases} +1 & if\ \ z_n(i) = 0 \\ -1 & if\ \ z_n(i) = 1 \end{cases} \quad for \quad i = 0,1,\ldots,2^{18}-2.$$

Finally, the n:th complex scrambling code sequence $S_{dl,n}$ is defined as ~~(the lowest index corresponding to the chip scrambled first in each radio frame)(where N is the period in chips and M is 131,072)~~:

$S_{dl,n}(i) = Z\cancel{z}_n(i) + j\, Z\cancel{z}_n((i+131072\cancel{M}) \text{ modulo } (2^{18}-1)),\ i=0,1,\ldots,38399\cancel{N-1}.$

Note that the pattern from phase 0 up to the phase of 38399 is repeated.

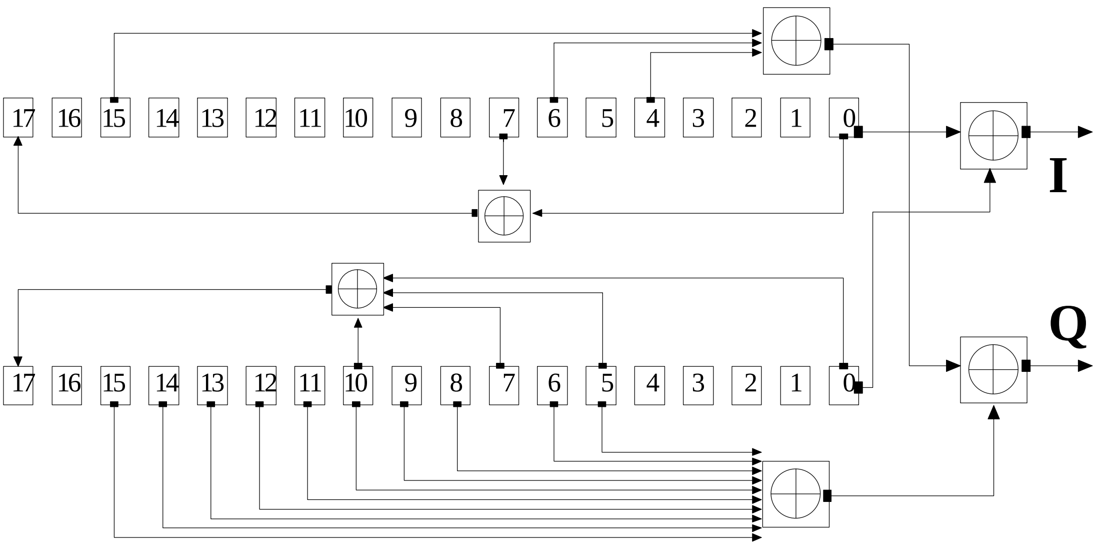

Figure 11: Configuration of downlink scrambling code generator