TSG RAN WG1#7bis **TSGR1#7bis(99)g04**

Kyongju, Korea
October 3 – 4, 1999

Agenda Item:	**Plenary**
Source:	**Golden Bridge Technology**
Title:	**Proposed editorial modifications to 25.213 (CPCH Sections)**
Document for:	**Discussion and Approval**

~~4.2.3 PCPCH~~

~~The spreading and modulation of the message part of the CPCH message part is basically the same as for the uplink dedicated physical channels, see section 4.2.1, where the uplink DPDCH and uplink DPCCH are replaced by the data part and the control part respectively.~~

4.3.4.1 Access Preamble

4.3.4.1.1 Preamble code construction

Similar to RACH access preamble codes, the CPCH access preamble codes $C_{c\text{-}acc,n,s}$, are complex valued sequences. The CPCH access preamble codes are built from the preamble scrambling codes $S_{c\text{-}acc,n}$ and a preamble signature $C_{sig,s}$ as follows:

$$C_{c\text{-}acc,n,s}(k) = S_{c\text{-}acc,n}(k) \times C_{sig,s}(k) \times e^{j(\frac{\pi}{4}+\frac{\pi}{2}k)}, \quad k = 0, 1, 2, 3, \ldots, 4095,$$

where $S_{c\text{-}acc,n}$ is defined in Section 4.3.4.1.2, below.

4.3.4.1.2 Access preamble scrambling code

The access preamble scrambling code generation is done in a way similar to that of PRACH with a difference of the initialisation of the x m-sequence in section 4.3.2.2. ~~–~~The long code $C_{scramb,n}$ ~~c257~~ (as described in Sections ~~4.3.3.1~~4.3.2.1 and 4.3.2.2) for the in-phase component is used directly on both in phase and quadrature branches without offset between branches. Only the first 4096 chips of the code are used for preamble scrambling. In the case when the ~~RACH~~ access resources ~~preambles~~ are shared between the RACH and CPCH, the ~~same~~ scrambling codes used in the RACH preamble will be used for the CPCH preamble as well.

The definition of the CPCH access preamble scrambling code sequence follows (the left most index correspond to the chip transmitted first in each slot):

$S_{c\ acc,n} = \mathrm{Re}\{C_{scramb,n}\}$,for chip indexes 0…4095 of $C_{scramb,n}$

4.3.4.2 CD Preamble

4.3.4.2.1 CD Preamble code construction

Similar to RACH access preamble codes, the CPCH CD preamble codes $C_{c\text{-}cd,n,s}$ are complex valued sequences. The CPCH CD preamble codes are built from the preamble scrambling codes $S_{c\text{-}cd,n}$ and a preamble signature $C_{sig,s}$ as follows:

$$C_{c\text{-}cd,n,s}(k) = S_{c\text{-}cd,n}(k) \times C_{sig,s}(k) \times e^{j(\frac{\pi}{4}+\frac{\pi}{2}k)}, \; k = 0, 1, 2, 3, \ldots, 4095,$$

where $S_{c\text{-}cd,n}$ is defined in Section 4.3.4.2.2 below.

4.3.4.2.2 CD preamble scrambling code

The CPCH CD preamble scrambling code is derived from the same scrambling code used in the CPCH access preamble. The long code $C_{scramb,n}$ (as described in Sections 4.3.2.1 and 4.3.2.2) for the in-phase component is used directly on both in phase and quadrature branches without offset between branches. The 4096 chips of the code from 4096 to 8191 are used for CPCH CD preamble scrambling. In the case when the access resources are shared between the RACH and CPCH, the scrambling codes used in the RACH preamble will be used for the CPCH CD preamble as well.
The definition of the CPCH CD access preamble scrambling code sequence follows (the left most index correspond to the chip transmitted first in each slot):
$S_{c\text{-}cd,n} = \mathrm{Re}\{C_{scramb,n}\}$,for chip indexes 4096…8191 of $C_{scramb,n}$

~~The 4096 chips from 4096 to 8191 of the code are used for the CD preamble scrambling with the chip rate of 3.84 Mchip/s. The long code c257 for the in-phase component is used directly on both in phase and quadrature branches without offset between branches. In the case when the RACH preambles are shared between the RACH and CPCH, the 4096 chips from 4096 to 8191 of the long code c1 used to scramble the RACH preamble will be used for the CPCH CD preamble.~~

4.3.4.5 Scrambling code for the CPCH message part

In addition to spreading, the message part is also subject to scrambling with a 10 ms complex code. The scrambling code is cell-specific and has a one-to-one correspondence to the scrambling code used for the preamble part.
$S_{c\text{-}msg,n} = C_{scramb,n}$,for chip indexes 8191…46591 of $C_{scramb,n}$.
In the case when the access resources are shared between the RACH and CPCH,
$S_{c\text{-}msg,n} = C_{scramb,n}$,for chip indexes 4095…42495 of $C_{scramb,n}$.

The generation of these codes is explained in 4.3.2.2. The mapping of these codes to provide a complex scrambling code is also the same as for the dedicated uplink channels and is described in 4.3.2.1.

~~In addition to spreading, the message part is also subject to scrambling. The scrambling code is cell specific and has a one to one correspondence to the spreading code used for the preamble part.~~

~~The scrambling codes used are formed from the continuation of the sequence xn and y used for the CD scrambling code and described in 4.3.4.2. Specifically, the values x_n(8192), x_n (8193),…, x_n (46591), and y(8192), y(8193),…, y(46591) are generated according to the recursive relations~~

~~in 4.3.3.1 and used to form the nth constituent codes, c_{1n} and c_{2n} (the left most index corresponding to the first chip scrambled in the message) as described in section 4.3.3.5.~~

~~In the case when the RACH preambles are shared between the RACH and CPCH, the same code set used to scramble the message part for the RACH will be used to scramble the message part for CPCH. The difference in this case is that the starting point of the codes will be chip 8192.~~

~~Scrambling code for the message part~~

~~In addition to spreading, the message part is also subject to scrambling with a 10 ms complex code. The scrambling code is cell specific and has a one to one correspondence to the scrambling code used for the preamble part.~~

~~$S_{r\text{-}msg,n} = C_{scramb,n}$,for chip indexes 4095…42495 of $C_{scramb,n}$~~

~~The generation of these codes is explained in 4.3.2.2. The mapping of these codes to provide a complex scrambling code is also the same as for the dedicated uplink channels and is described in 4.3.2.1.~~