ECS455: Chapter 6
Applications

6.1 3G (UMTS and WCDMA)

Dr. Prapun Suksompong

prapun.com/ecs455

Office Hours:

BKD 3601-7

Tuesday	**9:30–10:30**
Tuesday	**13:30–14:30**
Thursday	**13:30–14:30**

UMTS

- **Universal Mobile Telecommunications System** (**UMTS**)
- The research activity on UMTS started in **Europe** at the beginning of the 1990s.
  - Even before the earliest 2G systems arrived on the market
- Designed to support wideband services with data rates up to **2Mbit/s**.
- Developed **from GSM**
  - Keep the core network more-or-less intact
  - Change the air interface to use **CDMA**
- Compatibility between UMTS and GSM:
  - Most UMTS mobiles also implement GSM, and the network can *hand* them *over* from a UMTS base station to a GSM one if they reach the edge of the UMTS coverage area.
  - However, network operators can*not* implement the two systems in the same frequency band, so they are not fully compatible with each other.

Bandwidth Comparison

| Generation	Transmission Bandwidth	Standard
1G	25 and 30 kHz	
2G	200 kHz	GSM
	1.25 MHz	IS-95 (CDMA)
2.5G	1.25 MHz	CDMA2000 1X-RTT
3G	5 MHz	WCDMA and CDMA2000 3X-RTT
"4G"	Up to 20 MHz	LTE and WiMAX (802.16) **Wi-Fi (802.11a/g)**

Wider and wider radio frequency bands!

[Myung and Goodman, 2008]

UMTS: FDD

- The **chip rate** for spectrum spreading is 3.84 Mc/s.
- The maximum transmitter power of the user equipment is in the range of 21 to 33 dBm (that is, 125 mW to 2 W)

$$\left(10^{21/10}\,\text{mW}\right) \quad \left(10^{33/10}\,\text{mW}\right)$$

(0 dBm or dBmW = 1 mW)
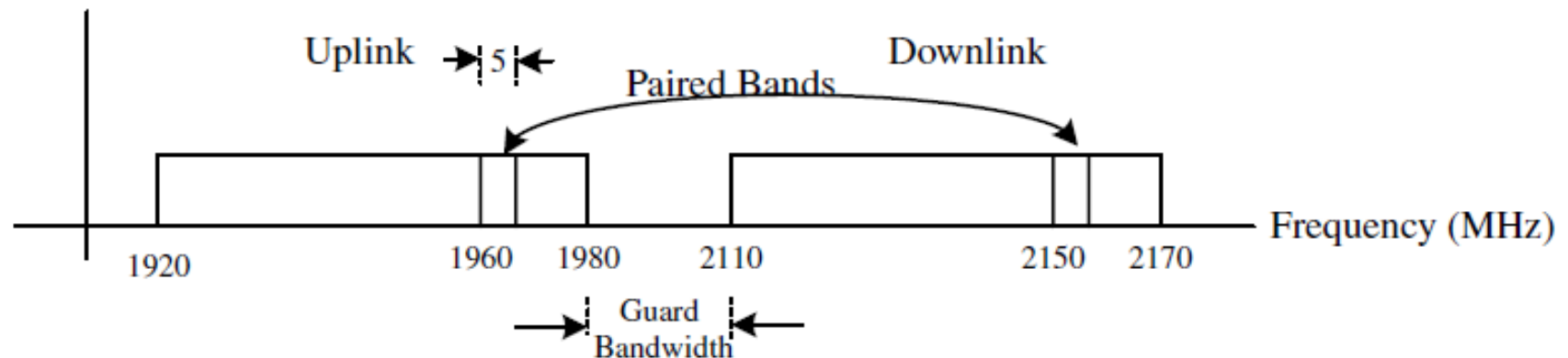
[Karim and Sarraf, 2002, Fig 6-1]

Review: CDMA

- Two Users. Suppose the code length $= N$. Taken from $\mathbf{H}_N$.

- User 1 uses code $\underline{c}_1$. Want to send messages $a_1,\, a_2,\, a_3,\, a_4,\, \ldots$

  - Send $\underline{x}_1 = \left[\, \boxed{a_1\underline{c}_1} \;\; \boxed{a_2\underline{c}_1} \;\; \boxed{a_3\underline{c}_1} \;\; \boxed{a_4\underline{c}_1} \cdots \right]$

- User 2 uses code $\underline{c}_2$. Want to send messages $b_1,\, b_2,\, b_3,\, b_4,\, \ldots$

  - Send $\underline{x}_2 = \left[\, \boxed{b_1\underline{c}_2} \;\; \boxed{b_2\underline{c}_2} \;\; \boxed{b_3\underline{c}_2} \;\; \boxed{b_4\underline{c}_2} \cdots \right]$

- Receiver gets $\underline{r} = \underline{x}_1 + \underline{x}_2$

  - To recover a_1, calculate $\dfrac{1}{N}\langle \underline{r}(1:N), \underline{c}_1 \rangle$

  - To recover b_1, calculate $\dfrac{1}{N}\langle \underline{r}(1:N), \underline{c}_2 \rangle$

  - To recover a_2, calculate $\dfrac{1}{N}\langle \underline{r}((N+1):(2N)), \underline{c}_1 \rangle$

Observe that, for successful transmission, we need

$$\underline{c}_1 \perp \underline{c}_2$$

or, equivalently,

$$\langle \underline{c}_1, \underline{c}_2 \rangle = 0.$$

Ex: CDMA

$$\mathbf{H}_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

- Two Users. Suppose the code length $= 4$. Taken from $\mathbf{H}_4$.

- User 1 uses code $\underline{c}_1$. Want to send messages $a_1, a_2, a_3, a_4, \ldots$

  - Send $\underline{x}_1 = [\; a_1 \quad a_1 \quad a_1 \quad a_1 \mid a_2 \quad a_2 \quad a_2 \quad a_2 \mid a_3 \quad a_3 \quad a_3 \quad a_3 \mid a_4 \quad a_4 \quad a_4 \quad a_4 \;]$

- User 2 uses code $\underline{c}_2$. Want to send messages $b_1, b_2, b_3, b_4, \ldots$

  - Send $\underline{x}_2 = [\; b_1 \quad -b_1 \quad b_1 \quad -b_1 \mid b_2 \quad -b_2 \quad b_2 \quad -b_2 \mid b_3 \quad -b_3 \quad b_3 \quad -b_3 \mid b_4 \quad -b_4 \quad b_4 \quad -b_4 \;]$

- Receiver gets $\underline{r} = \underline{x}_1 + \underline{x}_2$

  - To recover a_1, calculate $\dfrac{1}{4}\langle \underline{r}(1:4), \underline{c}_1 \rangle$

  - To recover b_1, calculate $\dfrac{1}{4}\langle \underline{r}(1:4), \underline{c}_2 \rangle$

  - To recover a_2, calculate $\dfrac{1}{4}\langle \underline{r}(5:8), \underline{c}_1 \rangle$

OVSF (1)

- Channelization codes used in UMTS W-CDMA and cdma2000 are **variable-length Walsh codes**, also known **as orthogonal variable spreading factor (OVSF)** codes.

- The spreading factors in UMTS may vary from 4 to 256 chips on uplink channels and from 4 to 512 chips on downlink channels.

  - In cdma2000, OVSF codes used on traffic channels may vary from 4 to 128 chips.

- **Comparison**: IS-95 uses a set of 64 fixed-length Walsh codes to spread forward physical channels. In the reverse direction, they are used for orthogonal modulation where every six symbols from the block interleaver output are modulated as one of 64 Walsh codes.

Ex: Multiple Code Lengths

- Two Users. Suppose the code length $= 4$.

- User 1 uses code $\underline{c}_{4,2}$. Want to send messages $a_1, a_2, a_3, a_4, \ldots$
  - Send $\underline{x}_1 = \left[\; \boxed{a_1 \underline{c}_{4,2}} \;\; \boxed{a_2 \underline{c}_{4,2}} \;\; \boxed{a_3 \underline{c}_{4,2}} \;\; \boxed{a_4 \underline{c}_{4,2}} \cdots \right]$

- User 2 uses code $\underline{c}_{8,7}$. Want to send messages $b_1, b_2, b_3, b_4, \ldots$
  - Send $\underline{x}_2 = \left[\; \boxed{b_1 \underline{c}_{8,7}} \;\;\; \boxed{b_2 \underline{c}_{8,7}} \cdots \right]$

- Receiver gets $\underline{r} = \underline{x}_1 + \underline{x}_2$

  $= \frac{1}{4}\left\langle \underline{x}_1(1:4) + \underline{x}_2(1:4), \underline{c}_{4,2} \right\rangle = \frac{1}{4}\left\langle a_1 \underline{c}_{4,2}, \underline{c}_{4,2} \right\rangle + \frac{1}{4}\left\langle b_1 \underline{c}_{8,7}(1:4), \underline{c}_{4,2} \right\rangle$

  - To recover a_1, calculate $\frac{1}{4}\left\langle \underline{r}(1:4), \underline{c}_{4,2} \right\rangle = a_1 + \frac{b_1}{4}\left\langle \underline{c}_{8,7}(1:4), \underline{c}_{4,2} \right\rangle$

  - To recover a_2, calculate $\frac{1}{4}\left\langle \underline{r}(5:8), \underline{c}_{4,2} \right\rangle = a_2 + \frac{b_1}{4}\left\langle \underline{c}_{8,7}(5:8), \underline{c}_{4,2} \right\rangle$

  - To recover b_1, calculate $\frac{1}{8}\left\langle \underline{r}(1:8), \underline{c}_{8,7} \right\rangle = \frac{a_1}{8}\left\langle \underline{c}_{8,7}(1:4), \underline{c}_{4,2} \right\rangle + \frac{a_2}{8}\left\langle \underline{c}_{8,7}(5:8), \underline{c}_{4,2} \right\rangle + b_1$

  - To recover b_2, calculate $\frac{1}{8}\left\langle \underline{r}(9:16), \underline{c}_{8,7} \right\rangle$

Observe that, for successful transmission, we need $\left\langle \underline{c}_{8,7}(1:4), \underline{c}_{4,2} \right\rangle = \left\langle \underline{c}_{8,7}(5:8), \underline{c}_{4,2} \right\rangle = 0$

So, we have some idea of how to define orthogonality for codes with different lengths.

Ex: Multiple Code Lengths

- Two Users. Code lengths are not the same

- User 1 uses code $\underline{c}_{4,2} = [1 \ -1 \ \ 1 \ -1]$.
  - Send $\underline{x}_1 = [a_1 \ -a_1 \ a_1 \ -a_1 \ a_2 \ -a_2 \ a_2 \ -a_2 \ a_3 \ -a_3 \ a_3 \ -a_3 \ a_4 \ -a_4 \ a_4 \ -a_4]$

- User 2 uses code $\underline{c}_{8,7} = [1 \ -1 \ -1 \ \ 1 \ -1 \ \ 1 \ \ 1 \ -1]$.
  - Send $\underline{x}_2 = [b_1 \ -b_1 \ -b_1 \ b_1 \ -b_1 \ b_1 \ b_1 \ -b_1 \ b_1 \ -b_1 \ -b_1 \ b_1 \ -b_1 \ b_1 \ b_1 \ -b_1]$

- Receiver gets $\underline{r} = \underline{x}_1 + \underline{x}_2$
  - To recover a_1, calculate $\frac{1}{4}\langle \underline{r}(1:4), \underline{c}_{4,2} \rangle$
  - To recover a_2, calculate $\frac{1}{4}\langle \underline{r}(5:8), \underline{c}_{4,2} \rangle$
  - To recover b_1, calculate $\frac{1}{8}\langle \underline{r}(1:8), \underline{c}_{8,7} \rangle$
  - To recover b_2, calculate $\frac{1}{8}\langle \underline{r}(9:16), \underline{c}_{8,7} \rangle$
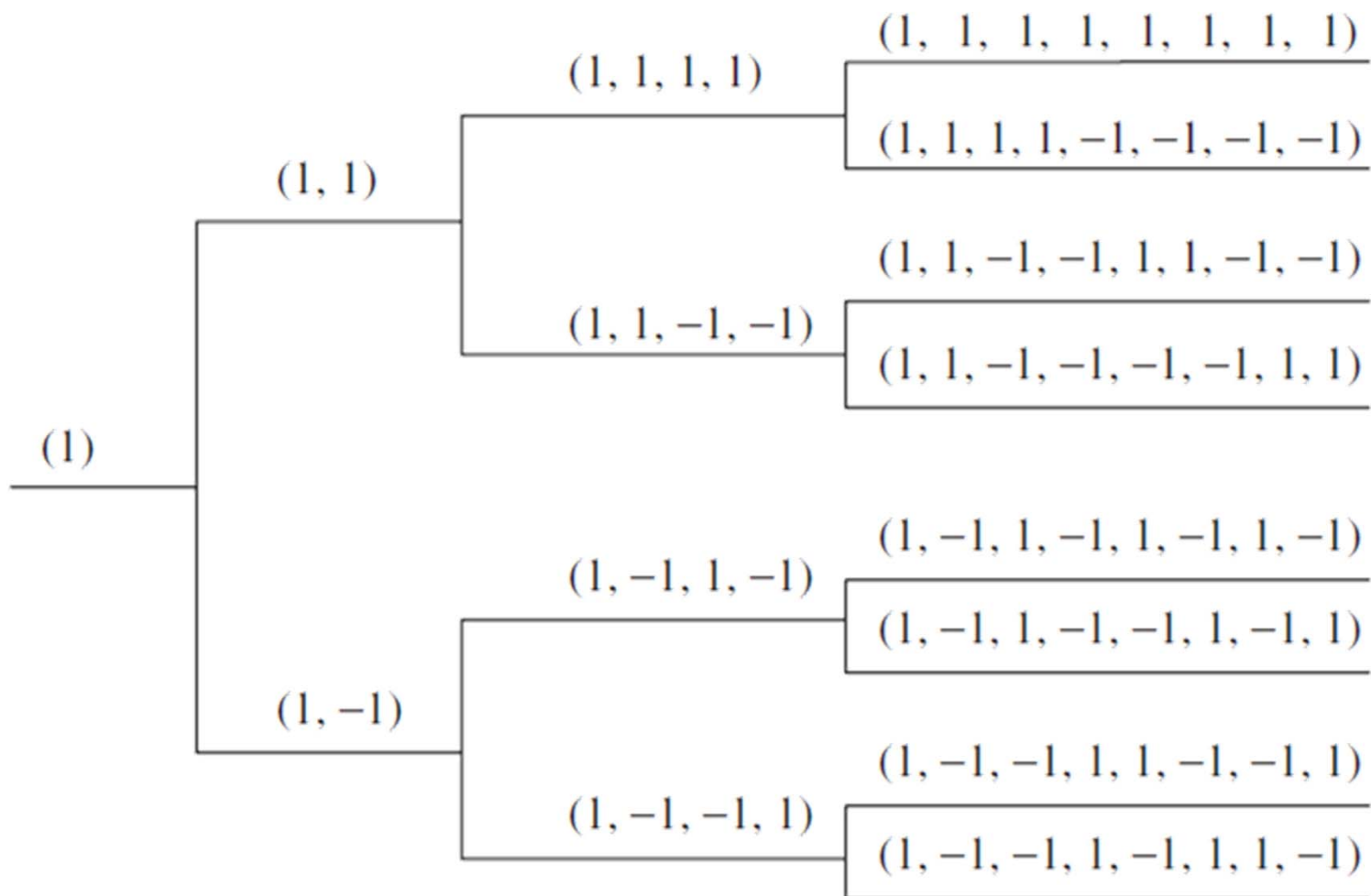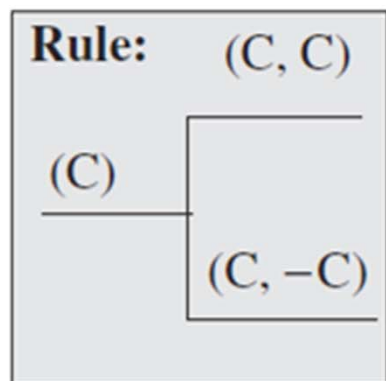
OVSF (2)

- Similar to Walsh sequences
  - Arranged and numbered in a different way
- Use a tree structure
- For each **spreading factor SF** = 1, 2, 4, . . . , which is a power of 2, there are $N = $ SF orthogonal codes obtained by the recursion relations:

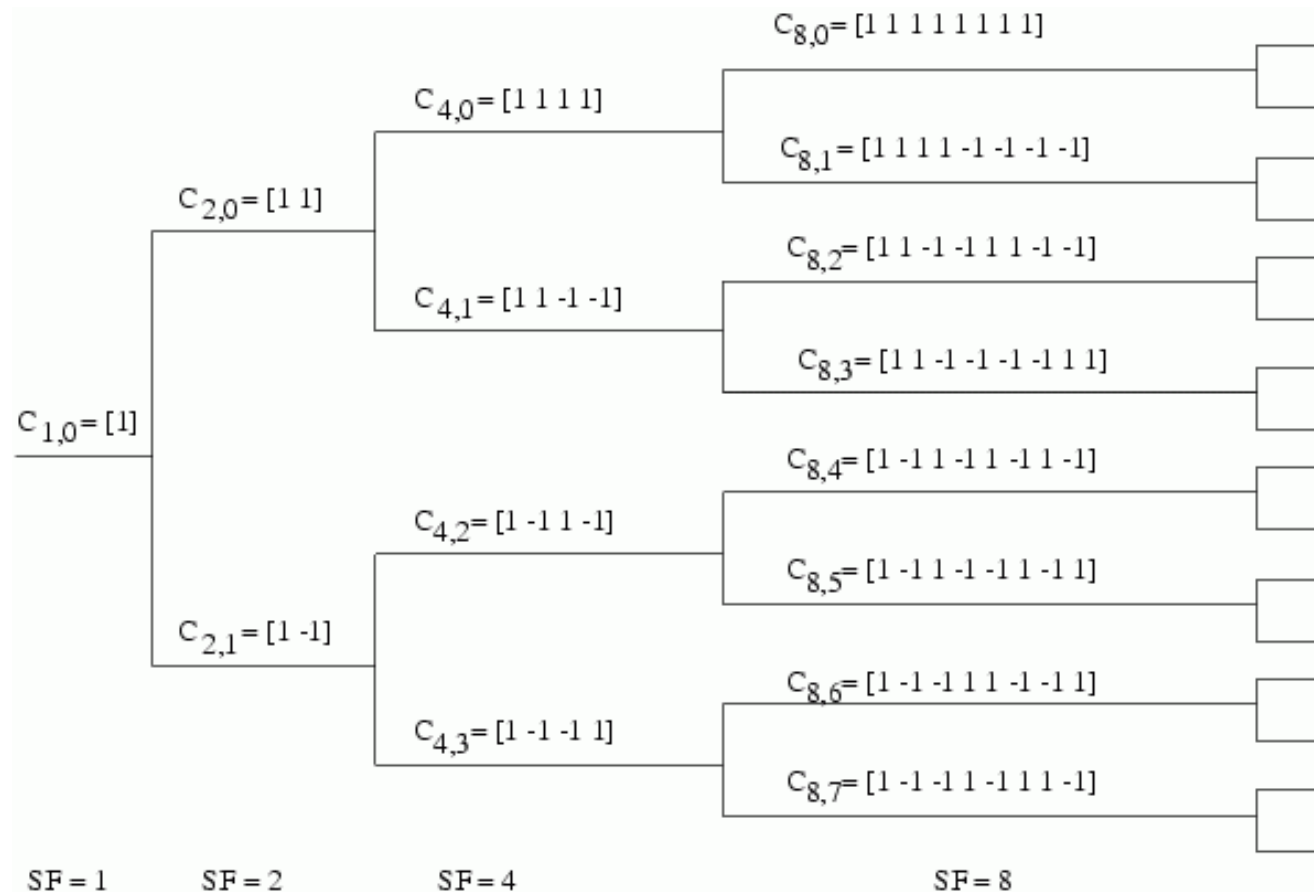$$c_{2SF,2m} = [c_{SF,m}, c_{SF,m}], \quad m = 0, 1, \ldots, SF - 1.$$

$$c_{2SF,2m+1} = [c_{SF,m}, -c_{SF,m}], \quad m = 0, 1, \ldots, SF - 1.$$

- Different data rates are supported on a physical channel by simply changing the spreading factor of the associated code.

Tree structure for OVSF codes (1)

Tree structure for OVSF codes (2)

$C_{1,0} = [1]$

$C_{2,0} = [1\ 1]$

$C_{2,1} = [1\ -1]$

$C_{4,0} = [1\ 1\ 1\ 1]$

$C_{4,1} = [1\ 1\ -1\ -1]$

$C_{4,2} = [1\ -1\ 1\ -1]$

$C_{4,3} = [1\ -1\ -1\ 1]$

$C_{8,0} = [1\ 1\ 1\ 1\ 1\ 1\ 1\ 1]$

$C_{8,1} = [1\ 1\ 1\ 1\ -1\ -1\ -1\ -1]$

$C_{8,2} = [1\ 1\ -1\ -1\ 1\ 1\ -1\ -1]$

$C_{8,3} = [1\ 1\ -1\ -1\ -1\ -1\ 1\ 1]$

$C_{8,4} = [1\ -1\ 1\ -1\ 1\ -1\ 1\ -1]$

$C_{8,5} = [1\ -1\ 1\ -1\ -1\ 1\ -1\ 1]$

$C_{8,6} = [1\ -1\ -1\ 1\ 1\ -1\ -1\ 1]$

$C_{8,7} = [1\ -1\ -1\ 1\ -1\ 1\ 1\ -1]$

SF = 1   SF = 2   SF = 4   SF = 8

12

Code allocation rules (1)

- OVSF codes can be applied to realize connections with **different data rates** by varying the spreading factor.

- Smaller SF = Faster data rate

- To have connections with different data rates, need some rules (for selecting the codes) to **maintain orthogonality**

- **Code blocking property**: If a certain code is already used for one connection, neither this code nor a code that is a descendant or an ancestor of this code (on the tree) is allowed to be used for another connection

  - These codes are not orthogonal to the already allocated one.

Code allocation rules (2)

Codes not allowed to be assigned to other connections

[Schulze Luders, 2005, Fig 5.12]

14
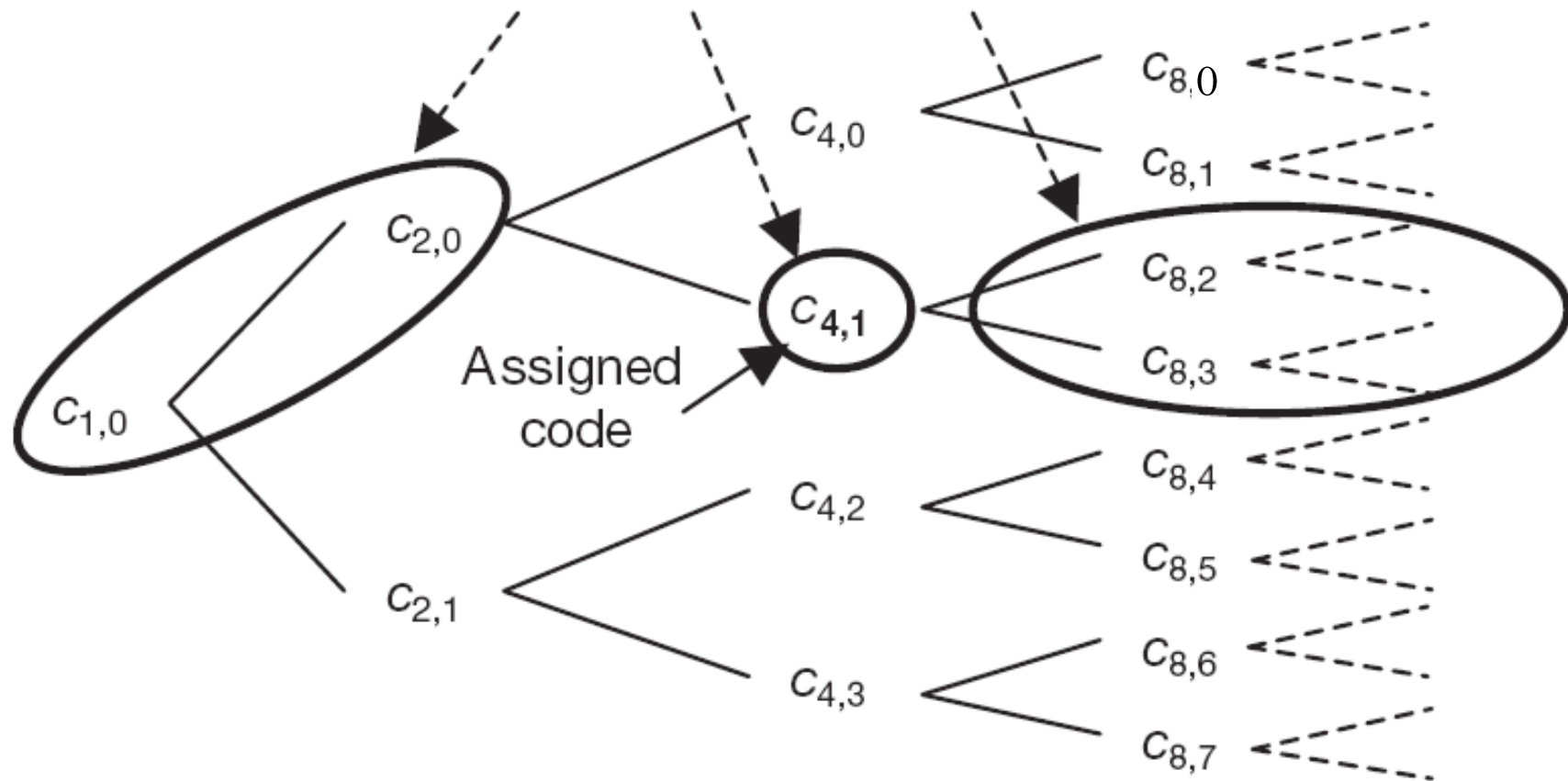
Code allocation rules (3)

- Two OVSF codes are orthogonal if and only if neither code lies on the path from the other code to the root.

- If, for example, code $c_{4,1}$ is in use, another connection with a different data rate is not allowed to use the encircled codes.

  - Other cades can still be used.

- If, for example, the second connection has twice the data rate of the first one, it has to select the code $c_{2,1}$.

  - Within the period of one data bit of connection 1, connection 2 transmits two data bits.