

Számítógépes Hálózatok

5. gyakorlat

Elérhetőségek

- honlap: <http://szalaigj.web.elte.hu/>
- email: szalaigindl@inf.elte.hu
- szoba: 2.507 (déli tömb)

Óra eleji kisZH

- Elérés:
 - <https://oktnb16.inf.elte.hu>



Connect to the TAO platform

Login

Password

[Guest access](#)

[Log in](#)

Gyakorlat tematika

- Szinkron CDMA
- Órai / házi feladat

Szinkron Code Division Multiple Access

- Induljunk ki kölcsönösen ortogonális vektorok halmazából.
- $v_i \perp v_j$ (az alábbi belső szorzat definíciót vesszük):

$$\mathbf{v}_i \cdot \mathbf{v}_j = \mathbf{v}_i^T \mathbf{v}_j = \sum_{k=0}^{D-1} \mathbf{v}_i[k] \mathbf{v}_j[k] = 0$$

- Pl. Walsh-mátrix sorai vagy oszlopai:

$$H(2^1) = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad H(2^2) = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

$$\forall k \in \mathbb{N} \wedge k \geq 2: H(2^k) = \begin{bmatrix} H(2^{k-1}) & H(2^{k-1}) \\ H(2^{k-1}) & -H(2^{k-1}) \end{bmatrix}$$

- Az i . állomáshoz rendeljünk hozzá egy \mathbf{v}_i vektort (chip szekvencia) az előző halmazból úgy, hogy mindegyiknek egyedi legyen.
- Kódolás: 0-ás bit $-\mathbf{v}_i$, 1-es bit \mathbf{v}_i

Szinkron Code Division Multiple Access

- Pl. ha az i . állomás átvitelre szánt adata: az 1011 bitsorozat, akkor az i . átviteli vektor: $(\mathbf{v}_i, -\mathbf{v}_i, \mathbf{v}_i, \mathbf{v}_i)$
- A csatornára kerülő szignál az átviteli vektorok összege
- A fogadott szignál dekódolása koordinátánként történik:
 - Tegyük fel, hogy a k . koordinátájában szereplő vektor \mathbf{u} . Ezt összeszorozzuk a küldő állomás megfelelő chip szekvenciájával, pl. \mathbf{v}_i -vel.
 - Ha $\mathbf{v}_i \cdot \mathbf{u} < 0$, akkor 0
 - Ha $\mathbf{v}_i \cdot \mathbf{u} > 0$, akkor 1
 - Ha $\mathbf{v}_i \cdot \mathbf{u} == 0$, akkor nincs üzenet az adott feladótól

Szinkron Code Division Multiple Access

- Miért működik? Két részvevőre $\{\mathbf{v}_1, \mathbf{v}_2\}$ fogjuk megmutatni egy speciális esetben, onnan már könnyen általánosítható
- Jelölés: $\|\mathbf{x}\| = \sqrt{\mathbf{x} \cdot \mathbf{x}}$ a vektor hossza
- Tegyük fel, hogy a szignál k . koordinátájában $\mathbf{u} = \mathbf{v}_1 - \mathbf{v}_2$ van
 - tehát az első állomásnak 1, a másodiknak 0 szerepel az átvitelre szánt bitsorozatának k . bitjénél
- Ekkor:
$$\mathbf{v}_1 \cdot \mathbf{u} = \mathbf{v}_1 \cdot (\mathbf{v}_1 - \mathbf{v}_2) = \mathbf{v}_1 \cdot \mathbf{v}_1 - \mathbf{v}_1 \cdot \mathbf{v}_2 = \|\mathbf{v}_1\|^2 - 0 > 0$$
$$\mathbf{v}_2 \cdot \mathbf{u} = \mathbf{v}_2 \cdot (\mathbf{v}_1 - \mathbf{v}_2) = \mathbf{v}_2 \cdot \mathbf{v}_1 - \mathbf{v}_2 \cdot \mathbf{v}_2 = 0 - \|\mathbf{v}_2\|^2 < 0$$
- Tehát a dekódolási módszer mindkettő esetben jól fog következtetni.

Feladat 1

- Adott két résztvevő, akik egy időben (azonos fázis) küldenek: az egyik $(1,-1)$ chip szekvenciával küldi a 1010 bitsorozatot, a másik az $(1,1)$ -gyel a 0110 sorozatot.
 - Adjuk meg külön-külön az átviteli vektort!
 - Adjuk meg az együttes átviteli vektort (a csatornára kerülő szignált)!
 - Írjuk fel az átvitt vektor dekódolásának lépéseit mindkét fél vevőjénél!

Feladat 1 megoldása

- Átviteli vektorok:

$$1010: (\mathbf{v}_1, -\mathbf{v}_1, \mathbf{v}_1, -\mathbf{v}_1) = ((1, -1), (-1, 1), (1, -1), (-1, 1))$$

$$0110: (-\mathbf{v}_2, \mathbf{v}_2, \mathbf{v}_2, -\mathbf{v}_2) = ((-1, -1), (1, 1), (1, 1), (-1, -1))$$

- Együttes átviteli vektor:

$$(\mathbf{v}_1 - \mathbf{v}_2, -\mathbf{v}_1 + \mathbf{v}_2, \mathbf{v}_1 + \mathbf{v}_2, -\mathbf{v}_1 - \mathbf{v}_2) = ((0, -2), (0, 2), (2, 0), (-2, 0))$$

- Dekódolás

- Az elsőnél:

$$\begin{bmatrix} 0 & -2 \\ 0 & 2 \\ 2 & 0 \\ -2 & 0 \end{bmatrix} \begin{bmatrix} 0 \cdot 1 + (-2) \cdot (-1) = 2 \\ 0 \cdot 1 + 2 \cdot (-1) = -2 \\ 2 \cdot 1 + 0 \cdot (-1) = 2 \\ -2 \cdot 1 + 0 \cdot (-1) = -2 \end{bmatrix} \rightarrow \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

- A másodiknál:

$$\begin{bmatrix} 0 & -2 \\ 0 & 2 \\ 2 & 0 \\ -2 & 0 \end{bmatrix} \begin{bmatrix} 0 \cdot 1 + (-2) \cdot 1 = -2 \\ 0 \cdot 1 + 2 \cdot 1 = 2 \\ 2 \cdot 1 + 0 \cdot 1 = 2 \\ -2 \cdot 1 + 0 \cdot 1 = -2 \end{bmatrix} \rightarrow \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

- Ha az érték pozitív $\rightarrow 1$, negatív $\rightarrow 0$, ha 0 \rightarrow nem küldött

Feladat 2

- Adott három állomás (A, B és C), amelyek CDMA-t használnak. Az A állomás az 1110, míg a B az 1010 bitsorozatot küldi azonos időben. C nem küld semmit. A chip szekvenciák a következők: $(6, -4, -12)$, $(-4, 12, -6)$, $(-12, -6, -4)$
 - Adjuk meg a jelsorozatokat (átviteli vektorokat), amit A és B elküld!
 - Tegyük fel, hogy interferencia történik az átvitel során és a két jel összeadódik. Mutassuk meg, hogyan dekódolható az interferált jelből az egyes állomások üzenete!

Feladat 2 megoldása I.

- Átviteli vektorok:

1110: $((6, -4, -12), (6, -4, -12), (6, -4, -12), (-6, 4, 12))$

1010: $((-4, 12, -6), (4, -12, 6), (-4, 12, -6), (4, -12, 6))$

- Együttes átviteli vektor:

$((2, 8, -18), (10, -16, -6), (2, 8, -18), (-2, -8, 18))$

- Dekódolás

– A:

$$\begin{bmatrix} 6 \\ -4 \\ -12 \end{bmatrix}$$

$$\begin{bmatrix} 2 & 8 & -18 \\ 10 & -16 & -6 \\ 2 & 8 & -18 \\ -2 & -8 & 18 \end{bmatrix} \begin{bmatrix} 12 - 32 + 216 = 196 \\ 60 + 64 + 72 = 196 \\ 12 - 32 + 216 = 196 \\ -12 + 32 - 216 = -196 \end{bmatrix} \rightarrow \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

Feladat 2 megoldása II.

- Dekódolás folytatása:

– B:

$$\begin{bmatrix} -4 \\ 12 \\ -6 \end{bmatrix}$$

$$\begin{bmatrix} 2 & 8 & -18 \\ 10 & -16 & -6 \\ 2 & 8 & -18 \\ -2 & -8 & 18 \end{bmatrix}$$

$$\begin{bmatrix} 196 \\ -196 \\ 196 \\ -196 \end{bmatrix}$$



$$\begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

– C:

$$\begin{bmatrix} -12 \\ -6 \\ -4 \end{bmatrix}$$

$$\begin{bmatrix} 2 & 8 & -18 \\ 10 & -16 & -6 \\ 2 & 8 & -18 \\ -2 & -8 & 18 \end{bmatrix}$$

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$



nem küldött

Órai / házi feladat

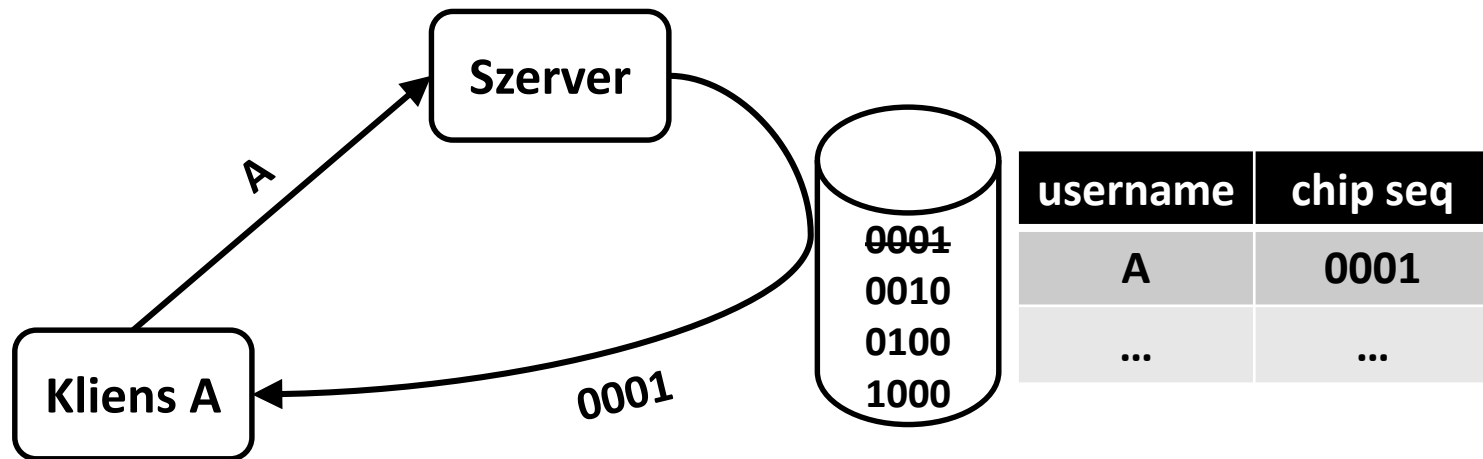
CDMA feladat

- Legyen 4 kliens (A,B,C,D), akik üzeneteket akarnak váltani egymással.
- A szerver tárolja a kiosztandó chip szekvenciákat: (0,0,0,1), (0,0,1,0), (0,1,0,0), (1,0,0,0)
- A kliens csatlakozik a szerverhez és az azonosítóját, majd kap egy üzenetet a szervertől, amely a chip szekvenciája lesz.
- A chip szekvenciát a szerver egy Queue példány tetejéről veszi le
 - Ha egy ötödik kliens is be akarna csatlakozni, annak "NNNN" üzenetet fog küldeni, ekkor az lépjen ki

Órai / házi feladat

CDMA feladat ábra – csatlakozás:

0



Órai / házi feladat

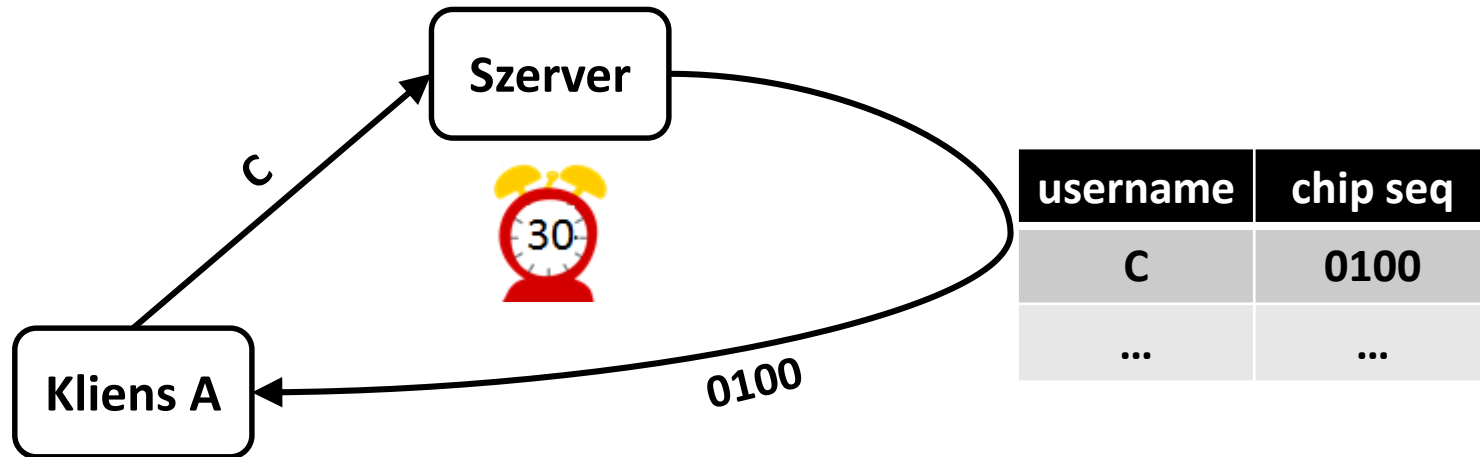
CDMA feladat folytatás

- A kliens oldalon megadhatunk üzenetet az alábbi formátumban:
- <cél azonosító> <0-1 bitekből álló üzenet>
 - Pl. "C 10010011"
 - Alapbeállításként az üzenet max. 8 darab bitből állhat, ha annál nagyobb, akkor a többit vágja le
- Mielőtt az üzenetet elküldi a szervernek, lekérdezi tőle a cél azonosítóhoz tartozó chip szekvenciát, és kódolja az üzenetet

Órai / házi feladat

CDMA feladat ábra – "C 10010011" üzenetküldése:

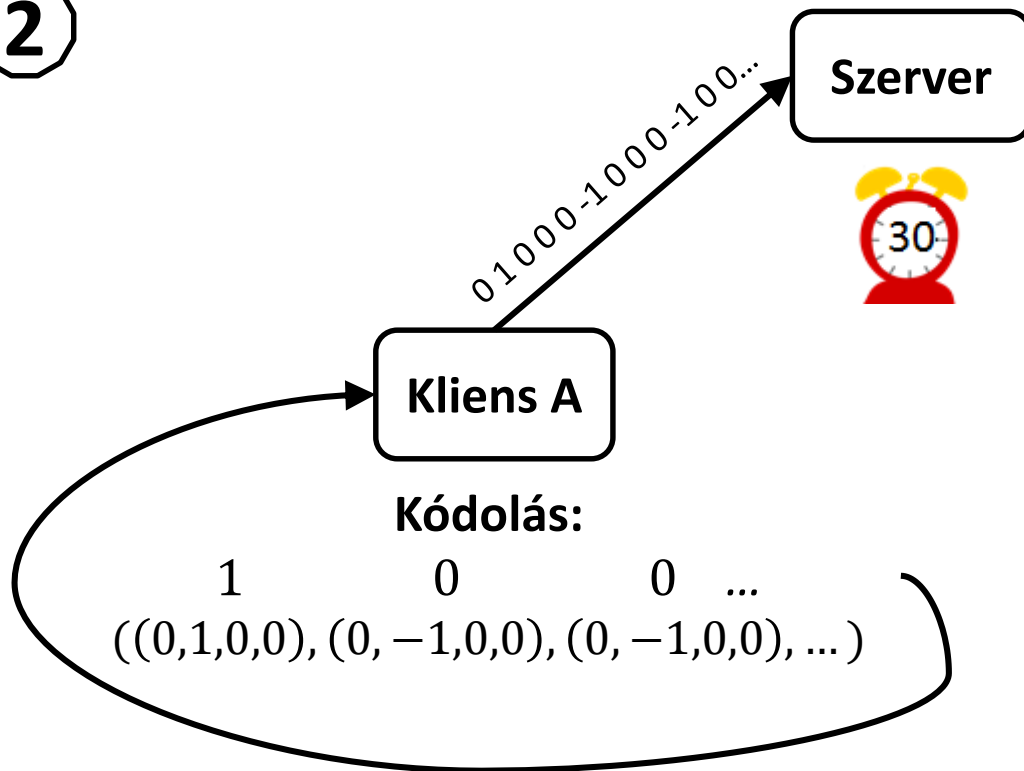
1



Órai / házi feladat

CDMA feladat ábra – "C 10010011" üzenetküldése:

2



Órai / házi feladat

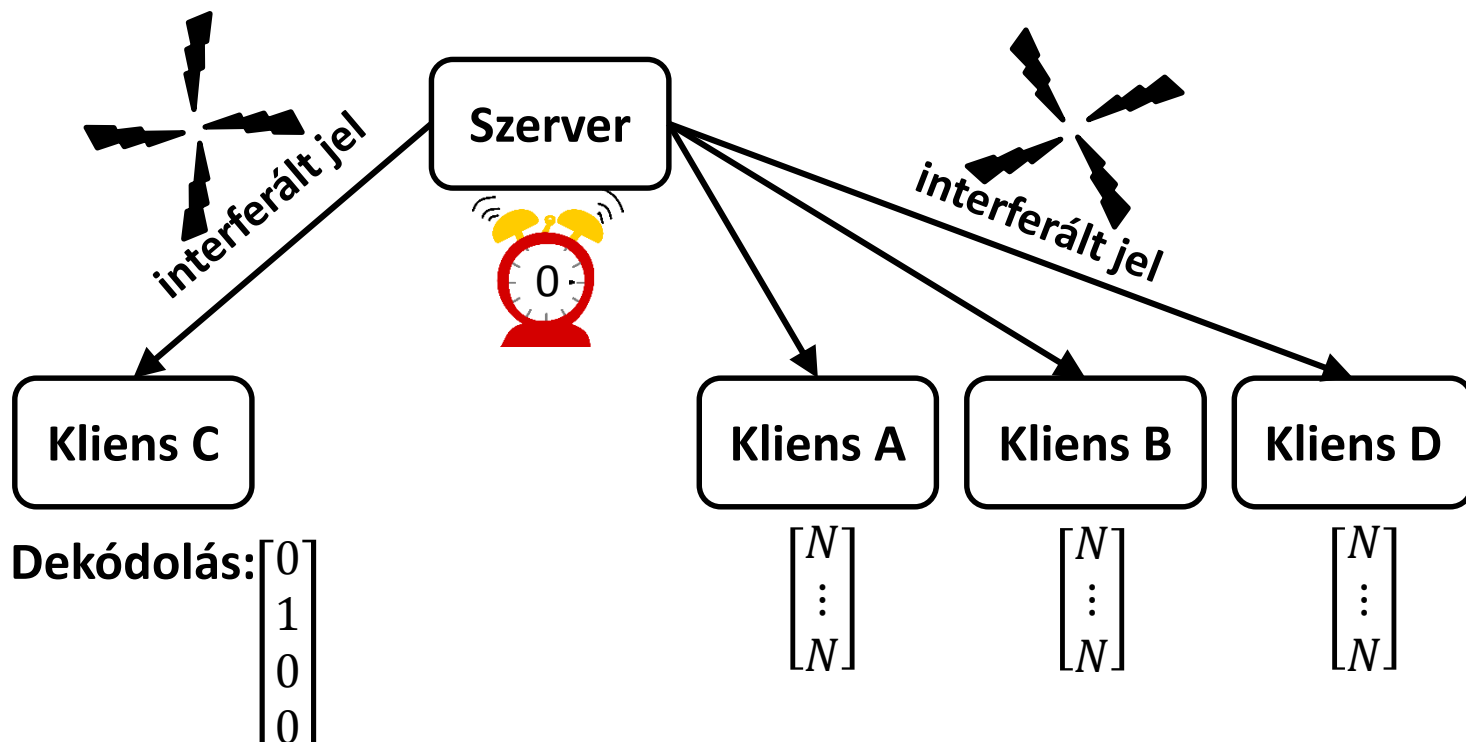
CDMA feladat folytatás

- A szerver 30 másodperces periódusonként folyamatosan gyűjti be az üzeneteket, és összeadja ezeket (interferencia), aztán kiküldi az összes kliensnek az interferált jelet
 - Ez egy 'b...b' formátumú struct-ba van csomagolva
 - Megjegyzés: azzal az esettel nem foglalkozunk, ha egy perióduson belül akár két kliens is ugyanoda küldeni
- A kliensek megkapják ezt, és a saját chip szekvenciájukkal dekódolják
 - Ha nem kaptak üzenetet, vagy rövidebb üzenetet kaptak, mint a maximális üzenet méret, akkor azokon a biteknél a kimeneten "N" szerepeljen

Órai / házi feladat

CDMA feladat ábra – "C 10010011" üzenetfogadása:

3



$$\begin{bmatrix} x_{1,1} & \cdots & x_{1,4} \\ \vdots & \ddots & \vdots \\ x_{8,1} & \cdots & x_{8,4} \end{bmatrix} \begin{bmatrix} x_{1,2} > 0 \\ \vdots \\ x_{8,2} > 0 \end{bmatrix} \rightarrow \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$$

Órai / házi feladat

CDMA feladat folytatás

- A szerver kódja letölthető a honlapomról:

<https://szalaigj.web.elte.hu/data/oktatas/szamitogepes-halozatok-gyakorlat/codes/haziCDMAServer.py>

- A feladat: a kliens implementálása

Órai / házi feladat

CDMA feladat példa futtatás

```
\Gyak5>python haziCDMAServer.py
new connection from ('127.0.0.1', 12772) with username B
new connection from ('127.0.0.1', 12773) with username A
new connection from ('127.0.0.1', 12774) with username D
new connection from ('127.0.0.1', 12775) with username C
closing ('127.0.0.1', 12773) after reading no data
closing ('127.0.0.1', 12772) after reading no data
closing ('127.0.0.1', 12775) after reading no data
closing ('127.0.0.1', 12774) after reading no data
Close the system
```

```
\Gyak5>python haziCDMAClient.py A
<A> Incoming message: 01110011
<A> Incoming message: NNNNNNNN
<A> Incoming message: 00110NNN
<A>
Close the client
```

```
\Gyak5>python haziCDMAClient.py C
<C> A 01110011
<C> Incoming message: 00001111
<C> Incoming message: NNNNNNNN
<C> D 0011
<C> Incoming message: NNNNNNNN
<C>
Close the client
```

```
\Gyak5>python haziCDMAClient.py B
<B> C 00001111
<B> Incoming message: 00111000
<B> Incoming message: NNNNNNNN
<B> A 00110
<B> Incoming message: NNNNNNNN
<B>
Close the client
```

```
\Gyak5>python haziCDMAClient.py D
<D> B 001110000000
<D> Incoming message: NNNNNNNN
<D> Incoming message: NNNNNNNN
<D> Incoming message: 0011NNNN
<D>
Close the client
```

VÉGE
KÖSZÖNÖM A FIGYELMET!