

Lab 3 - Detectia si corectarea erorilor

Detectia erorilor

- checksum → se calculeaza o suma finita pe un numar arbitrar de octeti
- Internet checksum
 - descompune un sir de octeti intr-un sir de cuvinte de cate 2 octeti
 - calculeaza negarea sumei lor pe biti
 - trimite rezultatul impreuna cu datele
 - la receptie, se recalculeaza checksum-ul si se compara
 - in caz de eroare, se cere retransmiterea datelor

Corectarea erorilor

- coduri Hamming → codeaza secventele de 4 biti in secvente de 7 biti (se intercaleaza 3 biti de paritate)

```
# SEND
# 1) se porneste de la un sir de 4 biti: d1, d2, d3, d4
# 2) se calculeaza bitii de paritate
p1 = d1 ^ d2 ^ d4
p2 = d1 ^ d3 ^ d4
p3 = d2 ^ d3 ^ d4
# 3) se trimite sirul de biti p1, p2, d1, p3, d2, d3, d4

# RECV
# 1) se primeste sirul r1, r2, r3, r4, r5, r6, r7
# 2) se calculeaza sindromul z
z1 = r1 ^ r3 ^ r5 ^ r7
z2 = r2 ^ r3 ^ r6 ^ r7
z3 = r4 ^ r5 ^ r6 ^ r7
z = (z3 << 2) | (z2 << 1) | (z1 << 0)
# 3) sindromul indica ce bit din cei 7 transmisi a avut valoarea schimbata
```

Implementare

- exercitiul 1 → afisati checksum-ul in *send.c* (t.hdr.sum) dupa ce-l calculati (23259)
- exercitiile 2-4

```
# am un octet c, cum iau al X-lea bit al lui intr-un bool b?
bool b = (c >> X) & 1;

# am un bool b care trebuie sa fie pe pozitia X dintr-un octet c, cum il setez?
uint8_t c = (b << X) | ...;
```

Link-uri

<https://ocw.cs.pub.ro/courses/pc/laboratoare/03>

https://inst.eecs.berkeley.edu/~cs61c/sp10/lec/C/C_Crash_Course.pdf

<https://beej.us/guide/bgnet/html/>