Telekommunikációs Hálózatok

6. gyakorlat

Feladat 1 (Szamológép)

- Készítsünk egy proxy szervert
  - TCP-s számológép klienstől kapja az üzenetet. (korábbi gyakorlat)
  - UDP-s számológép szervernek küldi tovább
  - A szerver visszaküldi a proxynak, aki visszaküldi a kliensnek

Feladat 2 (Számológép 2)

- Egy számológép kliens az UDP szervertől kérje el a TCP-s szerver elérhetőségét!
  - Küldjön egy ‚GET' üzenetet
- A kliens küldjön egy ‚Hello Server' üzenetet a UDP szervernek, aki visszaküldi a TCP szerver elérését, ahova a számokat és az operátort fogja elküldeni.
- A TCP szerver legyen a korábbi számológép szerver

Feladat 3 (Web proxy)

- Készítsünk proxyt, ahol a kliens egy webböngésző, a szerver pedig egy webszerver.
- A proxy tovvábítsa a böngésző kérését a szervernek.

- Pl: python netProxy.py ggombos.web.elte.hu 9000

- Böngészőben: localhost 9000

Feladat 4

- Tekintsük a következő paritás-technikát. Tekintsük az *n* küldendő adatbitet, mint egy *k* × *l* bit mátrixot. Minden oszlophoz számoljunk ki egy paritás-bitet (odd parity) és egészítsük ki a mátrixot egy új sorral, mely ezeket a paritás-biteket tartalmazza. Küldjük el az adatokat soronként.
- Példa *k* = 2, *l* = 3 esetén:

1 0 1

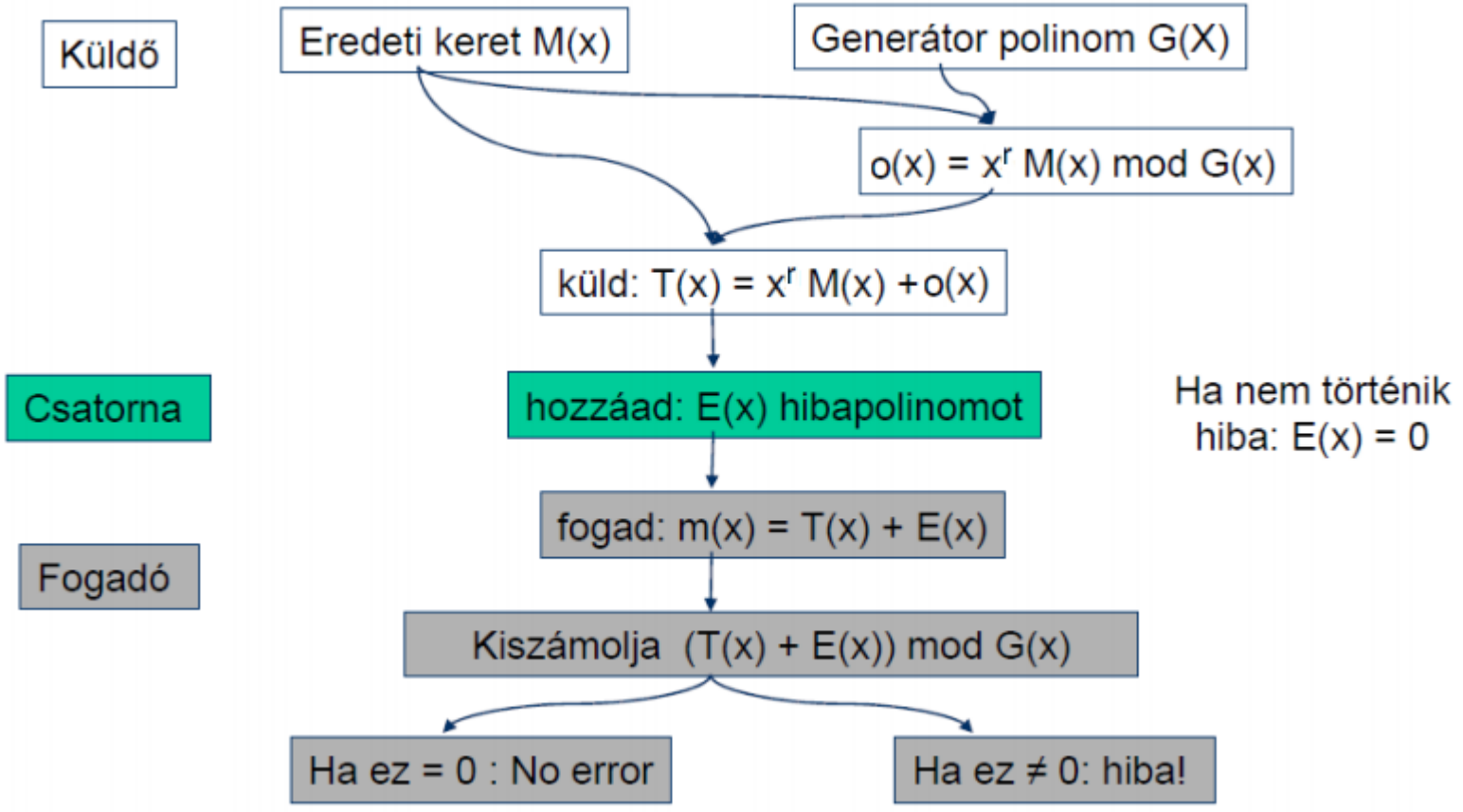
0 1 1

0 0 1

- Hogy viselkedik ez a módszer egyszerű bit-hibák és löketszerű (burst) bit-hibák esetén, ha *k* = 3, *l* = 4? Milyen hosszú lehet egy bitsorozat, melynek minden bitje hibás, hogy a hibázást meg tudjuk állapítani? (Löketszerű: egymás utáni bitek hibásan jönnek át)
- Egészítsük ki a mátrixot egy új oszloppal is, amely minden sorhoz paritás-bitet tartalmaz (két dimenziós paritás technika). Hogyan használható ez a módszer 1-bithiba javítására? Mi a helyzet több bithibával és löketszerű-hibákkal?

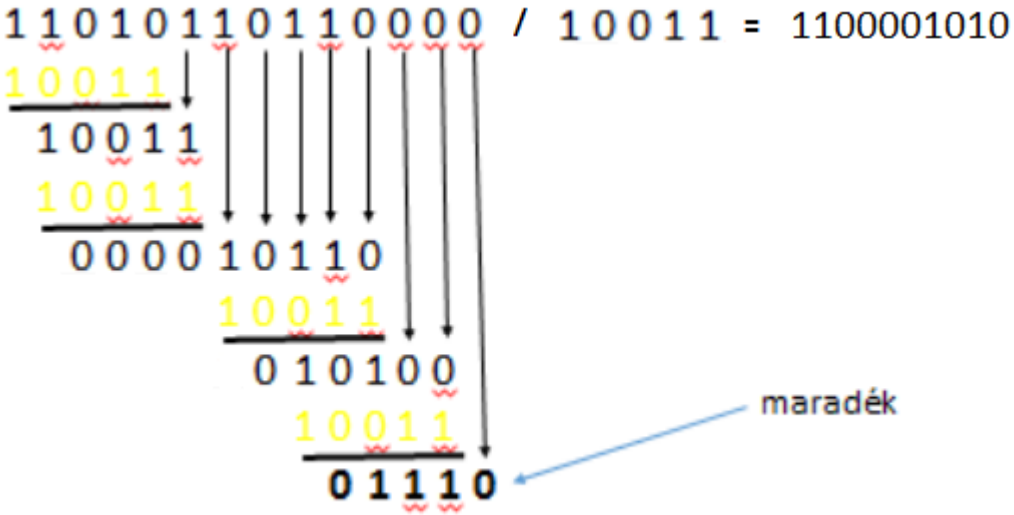
CRC hibajelző kód – emlékeztető

- Forrás: Dr. Lukovszki Tamás fóliái alapján



Példa CRC számításra – emlékeztető

- Keret ($M(x)$): 1101011011
- Generátor ($G(x)$): 10011
- Végezzük el a következő maradékos osztást: $\frac{11010110110000}{10011}$
- (A maradék lesz a CRC ellenőrzőösszeg)



Feladat 5

- Adva a $G(x) = x^4 + x^3 + x + 1$ generátor polinom.
- Számoljuk ki a $1100\ 1010\ 1110\ 1100$ bemenethez a 4-bit CRC ellenőrzőösszeget!
- A fenti üzenet az átvitel során sérül, a vevő adatkapcsolati rétege az $1100\ 1010\ 1101\ 1010\ 0100$ bitsorozatot kapja. Történt-e olyan hiba az átvitel során, amit a generátor polinommal fel lehet ismerni? Ha nem, akkor ennek mi lehet az oka?

CRC, MD5, SHA1 pythonban

- CRC

```
import binascii, zlib
test_string= "Fekete retek rettenetes".encode('utf-8')
print(hex(binascii.crc32(bytearray(test_string))))
print(hex(zlib.crc32(test_string)))
```

- MD5

```
import hashlib
test_string= "Fekete retek rettenetes".encode('utf-8')
m = hashlib.md5()
m.update(test_string)
print(m.hexdigest())
```

- SHA1

```
import hashlib
test_string= "Fekete retek rettenetes".encode('utf-8')
m = hashlib.sha1()
m.update(test_string)
print(m.hexdigest())
```

Házi feladat
netcopy alkalmazás

https://ggombos.web.elte.hu/halobeadando/5-netcopy/

VÉGE
KÖSZÖNÖM A FIGYELMET!