

## Laboratorul 03 - Implementarea unui protocol cu fereastra glisanta. Suma de control

Responsabili:

### Fereastra glisanta:

Legatura de date poate fi asemanata cu un cilindru in care datele sunt introduse de catre transmitator si primite de catre receptor. Aria sectiunii cilindrului reprezinta viteza de transmisie, iar inaltimea este timpul de propagare.

- Bandwidth
  - reprezinta cantitatea de informatie care poate fi transmisa intr-o unitate de timp pe legatura de date
  - termeni sinonimi: rata de transmisie, capacitatea legaturii, latimea de banda
  - unitate de masura: Mb/s - megabiti/secunda
- Delay
  - reprezinta timpul in care un bit se propaga de la sursa la destinatie de-a lungul legaturii de date
  - unitate de masura: s - secunda

Volumul cilindrului determina cantitatea de informatie aflata pe legatura de date, la un anumit moment de timp. Deci, cantitatea de informatie aflata pe fir la un anumit moment de timp este: Bandwidth × Delay.

Pentru a descrie timpul petrecut de un cadru in retea, se foloseste si notiunea de RTT (Round Trip Time), ce reprezinta timpul scurs din momentul in care un cadru este trimis pana in momentul in care este primita confirmarea.



Observatie: simulatorul din scheletul de cod considera timpul de propagare pentru confirmare egal cu 0, deci RTT = DELAY.

Fereastra W (window) = numarul maxim de cadre neconfirmate la orice moment de timp. Pentru protocolul STOP AND WAIT, fereastra este de 1 cadru. Asadar, o utilizare din plin a legaturii presupune o fereastra egala cu BANDWITH \* DELAY / sizeof(FRAME).



Atentie la unitatile de masura! Pentru laboratorul curent ne vom exprima in milisecunde si biti. Asadar vom avea WINDOW\_SIZE = (bandwidth \* delay \* 1000) / (sizeof(FRAME) \* 8).

### Suma de control:

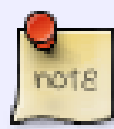
Ne intereseaza verificarea corectitudinii cadrelor primite deoarece in timpul transmiterii datelor, integritatea cadrului poate sa fie compromisa. Pentru a se depista acest lucru folosim o suma de control. Pentru detectie in cadrul laboratorului vom folosi Longitudinal Redundancy Check (Parity Word).

Metoda pe care o vom folosi in cadrul laboratorului se bazeaza pe spargerea mesajului trimis in blocuri de dimensiuna fixa N si calcularea sumei XOR al tuturor blocurilor. Rezultatul este adaugat ca un bloc de lungime N in fiecare cadru trimis.


Pentru verificarea corectitudinii mesajului primit, destinatarul o sa aplice aceeasi pasi pe mesajul primit si o sa compare cele doua sume de control.

### Cerinte laborator:

In laboratorul curent se cere sa se implementeze un protocol prin care legatura de date sa fie utilizata in mod eficient.



In cadrul laboratorului curent, legatura de date nu pierde si nu corupe informatia transmisa.

1. Descarcati arhiva de laborator  schelet\_lab03.zip
2. Modificati, din scriptul run experiment, parametrii SPEED si/sau DELAY. Observati schimbarile in timpul de rulare. Folositi utilitarul "time".
3. Implementati un protocol care sa utilizeze eficient legatura de date.
4. Adaugati in structura de date ce reprezinta frame-ul un nou camp pentru calculul sumei de control. Calculati si verificati aceasta suma pentru fiecare cadru.
5. BONUS: Ce se intampla atunci cand se trimite o cantitate de date care depaseste valoarea WINDOW\_SIZE? Modicati programul pentru a evidentia acest caz. Discutati cu asistentul problemele care pot sa aparea.
6. BONUS: Implementati verificarea integritatii cadrului utilizand Codul Hamming.

### Software disponibil:

1. Simulator legatura de date - executabilul link - generat in urma comenzii make in directorul /link\_emulator.
2. Schelet de cod pentru transmitator si receptor
3. API simulator:
  - a. int send message(msg\* m)
    - parametru: mesajul care va trimis
    - rezultat: numarul de octeti transferati(in caz de succes) sau -1 in caz de eroare
  - b. int recv message(msg\* m)
    - parametru: adresa la care se memoreaza datele primite
    - rezultat: numarul de octeti receptionati(in caz de succes) sau -1 in caz de eroare
4. API lucru cu fisiere:
  - a. open(char\* file\_name, int permissions)
    - file\_name = numele fisierului
    - permissions = O\_RDONLY, O\_WRONLY, O\_CREAT. Pentru a deschide pentru scriere si pentru creare, in caz ca nu exista, dati ca parametru "O\_WRONLY | O\_CREAT".
    - rezultat: O valoare int care reprezinta FILE DESCRIPTOR-ul fisierului sau -1 in caz de eroare.
  - b. read(int fd, void\* buf, size\_t n\_bytes)
    - fd = FILE DESCRIPTOR-ul fisierului din care se va face citirea
    - buf = zona de memorie in care se vor retine datele citite
    - n\_bytes = numarul de octeti care vor fi cititi
    - rezultat: numarul de octeti care au fost cititi sau -1 in caz de eroare
  - c. write(int fd, void\* data, size\_t n\_bytes)
    - fd = FILE DESCRIPTOR-ul fisierului in care se va scrie.
    - data = datele ce vor fi scrise in fisier.
    - n\_bytes = numarul de octeti al datelor care vor fi scrise in fisier.
    - rezultat: numarul de octeti care au fost scrisi sau -1 in caz de eroarea
5. Compilare schelet de cod: make
6. Rulare cod: ./run\_experiment.sh
7. Generare fisiere: utilitarul dd.
  - Exemplu: dd if=/dev/urandom of=<output\_file> bs=<file\_size\_in\_bytes> count=<no\_of\_blocks\_of\_bs\_size>

Search

#### Cursuri

- [Cursul 01.](#)
- [Cursul 02.](#)
- [Cursul 03.](#)
- [Cursul 04.](#)
- [Cursul 05.](#)
- [Cursul 06.](#)
- [Cursul 07.](#)
- [Cursul 08.](#)
- [Cursul 09.](#)
- [Cursul 10.](#)
- [Cursul 11.](#)
- [Cursul 12.](#)

#### Laboratoare

- [Laboratorul 01 - Notiuni pregatitoare pentru laboratorul de PC](#)
- [Laboratorul 02 - Folosirea unei legaturi de date pentru transmiterea unui fisier](#)
- [Laboratorul 03 - Implementarea unui protocol cu fereastra glisanta. Suma de control](#)
- [Laboratorul 04 - Forwarding](#)
- [Laboratorul 05 - ICMP](#)
- [Laboratorul 06 - Socketi UDP](#)
- [Laboratorul 07 - Protocolul de transport TCP](#)
- [Laboratorul 08 - TCP și multiplexare I/O](#)
- [Laboratorul 09 - Protocolul DNS](#)
- [Laboratorul 10 - Protocolul HTTP](#)
- [Laboratorul 11 - E-mail](#)
- [Laboratorul 12 - Protocoale de securitate. OpenSSL CLI tools](#)
- [Laboratorul 13 - Protocoale de securitate. utilizarea programatica](#)

#### Resurse

- [Mașina virtuală](#)

#### Table of Contents

- [Laboratorul 03 - Implementarea unui protocol cu fereastra glisanta. Suma de control](#)
  - [Fereastra glisanta:](#)
  - [Suma de control:](#)
  - [Cerinte laborator:](#)
  - [Software disponibil:](#)