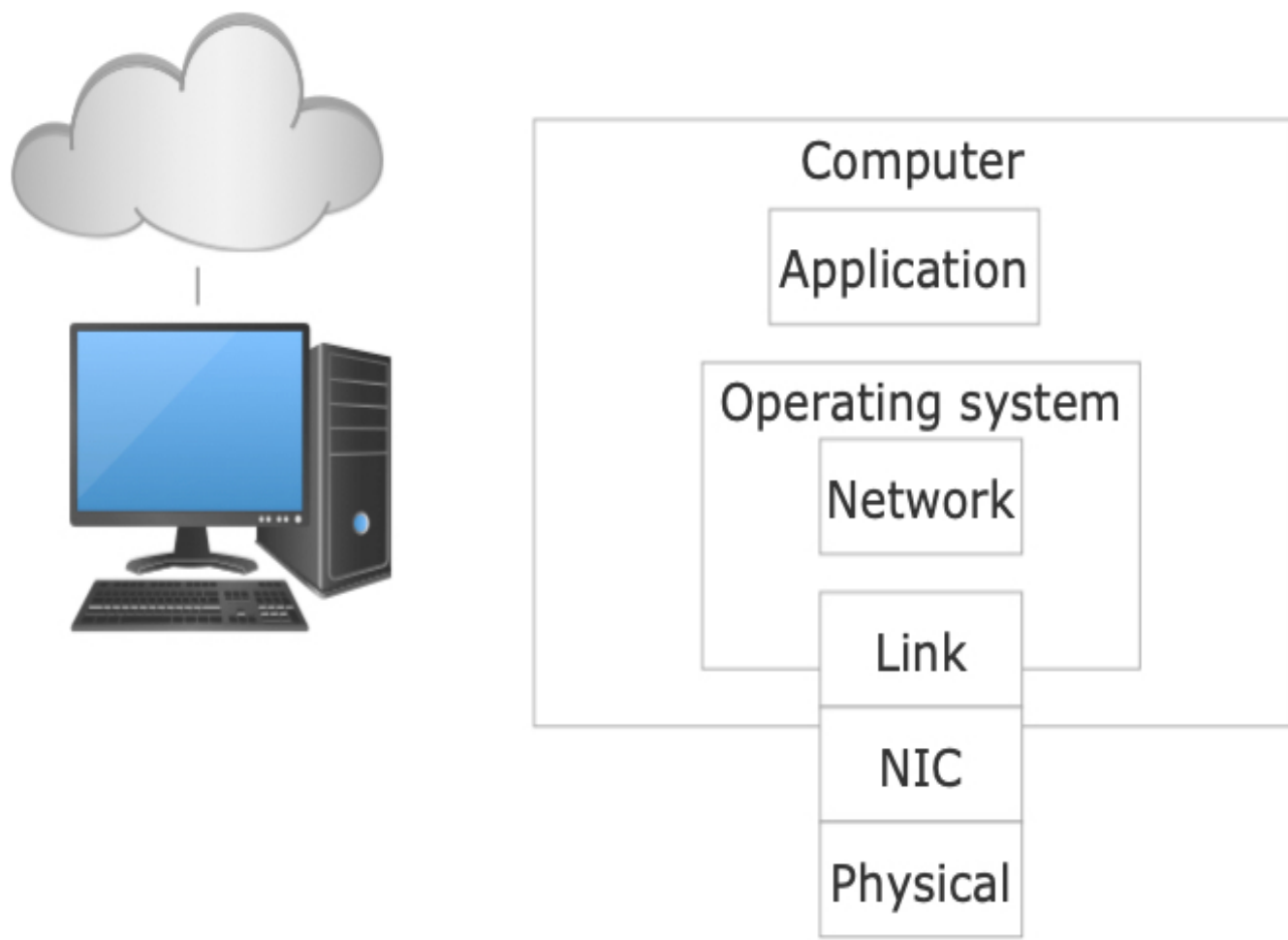


Laboratorul 02 - Folosirea unei legaturi de date pentru transmiterea unui fisier

Responsabili: Petruța Balint, Antoniu Neacsu, Andrei Stanca, Cătălin Leordeanu

Inainte de laborator

Pentru inceput, vom analiza modul in care se realizeaza comunicarea intre nivelurile stivei OSI: nivelul Fizic, Legatura de Date si Retea sunt procese independente ce comunica prin schimburi de mesaje. Procesele ce tin de nivelul Fizic si de sub-layerul Media Access Control din Data Link ruleaza pe hardware dedicat (de exemplu, Network Interface Card). Restul proceselor executate de Data Link si de Network se executa pe CPU, prin sistemul de operare. In figura, putem observa o implementare:



In acest laborator ne dorim sa realizam un schimb de date intre doua statii diferite (ce vor fi simulate prin instante diferite a programului), unul fiind un transmitator, celalalt un receptor, pentru a simula comportamentul Data Link-ului.

In realitate, transmitatorul preia un pachet de la nivelul Retea, construiește un cadru (adauga antetele specifice nivelului Data Link) si il trimite mai departe. Pentru simplitate si pentru a vedea mai usor comportamentul protocolului, vom ignora nivelul Retea si antetele, scopul celor din urma fiind corectarea erorilor (presupunem faptul ca Legaturile de Date nu corup si nu pierd date).

Receptorul asteapta sa se petreaca un eveniment, sigura posibilitate fiind cea de a primi un mesaj de la transmitator. Pentru a oferi receptorului timp de a procesa datele primite, de a garanta faptul ca puterea sa de procesare mai lenta (fata de un transmitator rapid) nu impacteaza integritatea datelor, se va introduce ideea de feedback, prin mesaje de tip **ACKNOWLEDGED** (prescurtat ACK). Receptorul trimite un astfel de mesaj, semnaland transmitatorului ca a primit corect mesajul disponibilitatea sa de a primi cadre din nou.

Protocoloalele in care transmitatorul trimite un cadru, asteapata un mesaj de tip ACK si continua trimiterea datelor de unde a ramas doar daca primeste acest mesaj, se numesc protocoale **STOP AND WAIT**.

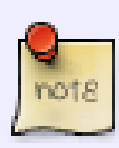
Desi cadrele circula in ambele directii (transmitator → sender, sender → transmitator), datorita secventialitatii impuse de protocol, un canal de comunicatie half-duplex este suficient.

Cerinte laborator:

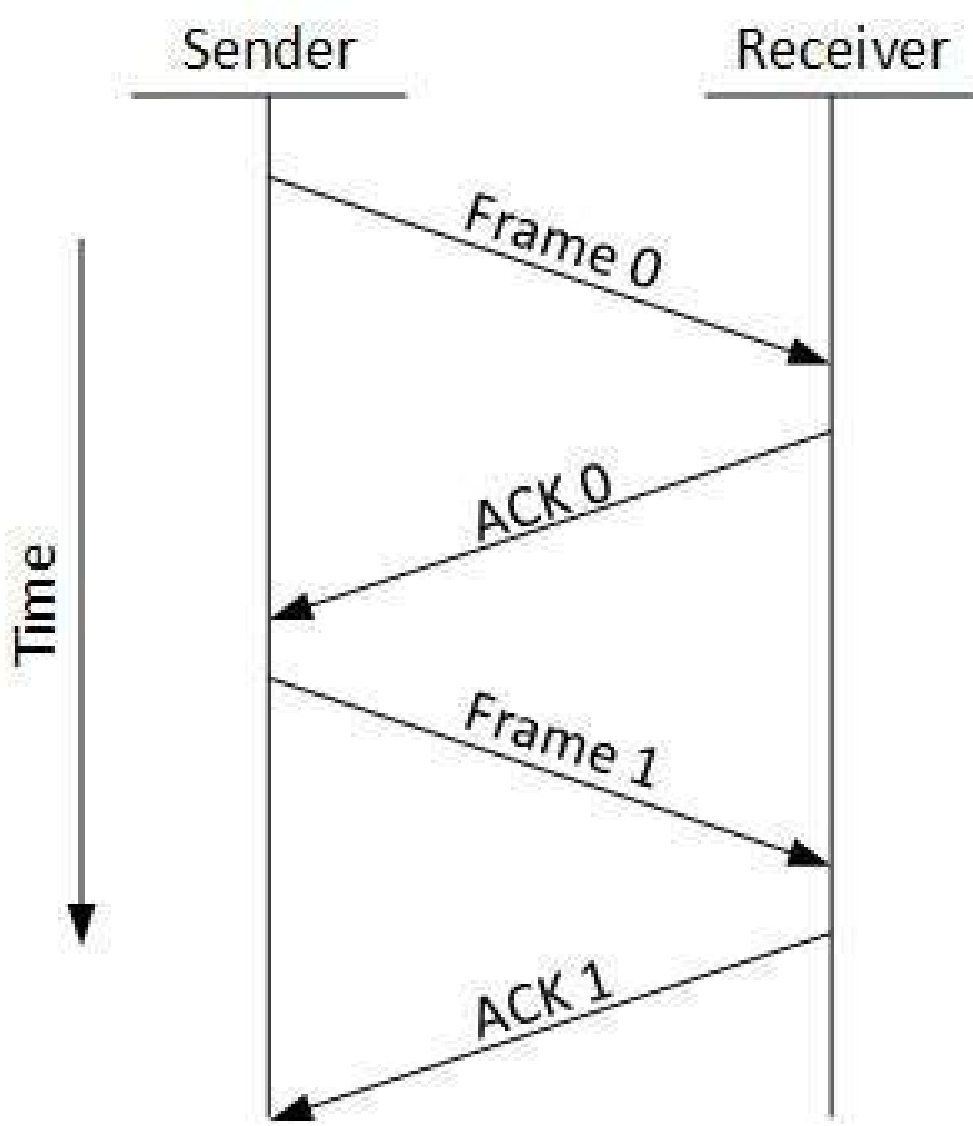
Se cere sa se implementeze un protocol pentru transmisia de fisiere pornind de la scheletul de cod pus la dispozitie. Legatura de date dintre transmitator si receptor este simulata prin intermediul executabilului **link**. Mai precis, se simuleaza doua legaturi:

- legatura de date pentru transmiterea datelor (transmitator → receptor)
- legatura de date pentru confirmari (receptor → transmitator)

Protocolul implementat trebuie sa fie de tipul STOP AND WAIT. Transmitatorul trimite un mesaj cu date, asteapta confirmarea primirii mesajului de la receptor si trimite urmatorul mesaj, ordinea pachetelor fiind astfel asigurata. Receptorul asteapta mesaje cu date de la transmitator si trimite mesaje de confirmare.



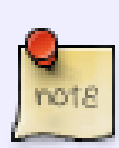
Legaturile de date nu pierd si nu corup mesaje. Confirmarile se transmit instantaneu.



Detalii de implementare:

- transmitatorul primeste ca argument numele fisierului care va fi transmis
- receptorul nu cunoaste numele fisierului care va fi transmis si nici dimensiunea - aceste informatii trebuie primite de la transmitator
- transmitatorul trimite continutul fisierului catre receptor
- receptorul primeste datele de la transmitator si le scrie intr-un fisier

Pasi de rezolvare:

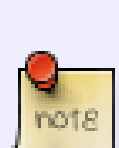


Nu trebuie sa fie modificata structura definita in lib.h - se vor utiliza doar campurile len si payload. Va trebui sa fie definita o noua structura, utila pentru a gestiona tipurile de mesaje stabilite, care sa fie scrisa in payload.

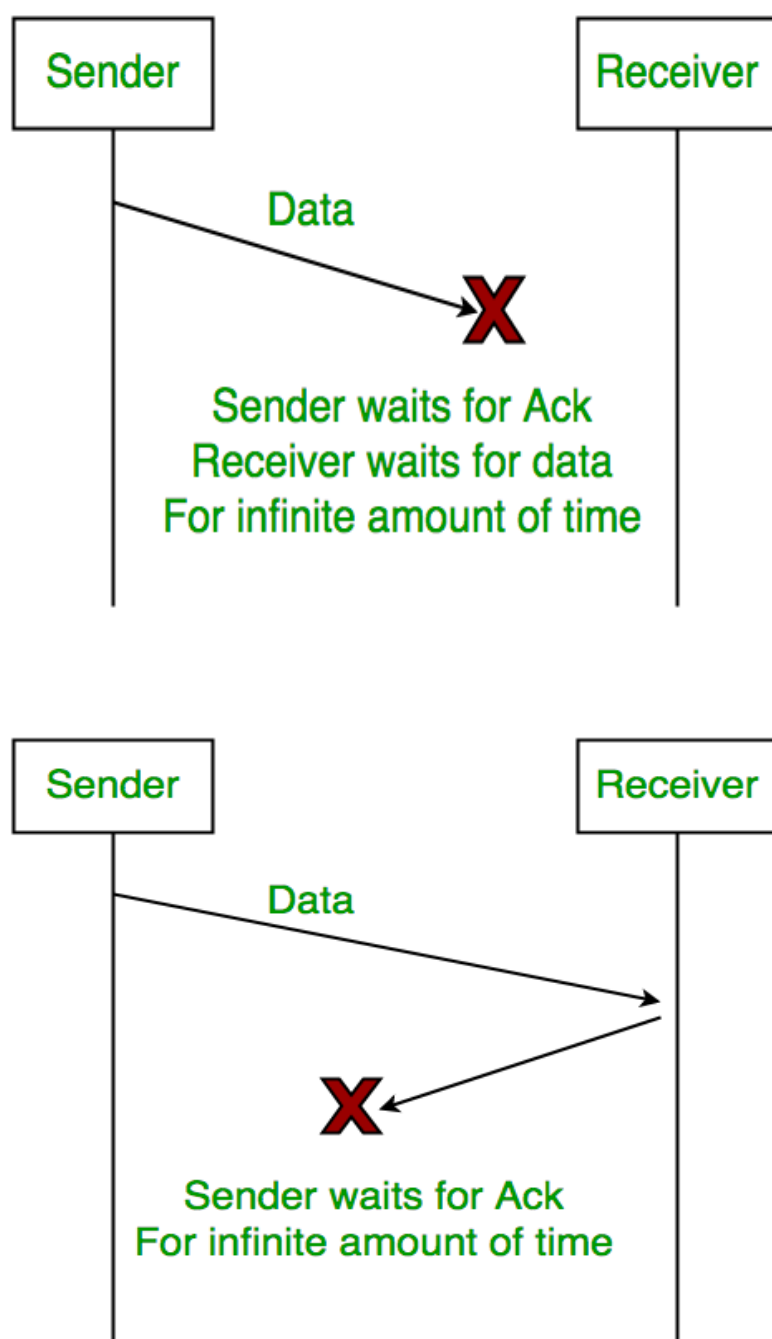


Executati scriptul **./run_experiment.sh** pentru a observa un demo al functionalitatilor disponibile. Il puteti folosi in **testarea codului**, acesta pornind simulatorul de data link, ruland **./recv** si **./send**.

- Descarcati arhiva de laborator [schelet_lab2.zip](#)
- Stabiliti tipurile de mesaje necesare
- Implementarea transmisiei unui singur fisier
- BONUS: Implementare transmisie simultana pentru mai multe fisiere



Ce s-ar intampla intr-un caz real? Ce dezavantaje puteti identifica pentru aceasta implementare de STOP AND WAIT? Discutati cu asistentul ce se poate intampla in cazul in care mesajele de confirmare sunt alterate/pierdute, sau legatura de date este incepta?



Software disponibil:

- Simulator legatura de date - executabilul link - generat in urma comenzii make
- Schelet de cod pentru transmitator si receptor
- API simulator:
 - int send message(msg* m)
 - parametru: mesajul care va fi trimis
 - rezultat: numarul de octeti transferati (in caz de succes) sau -1 in caz de eroare
 - int rcv message(msg* m)
 - parametru: adresa la care se memoreaza datele primite
 - numarul de octeti receptionati(in caz de succes) sau -1 in caz de eroare
- Compilare schelet de cod: make

Utilitare necesare

- Generare fisiere de o anumita dimensiune: dd
- Verificare corectitudine transmisie: diff

Alte resurse

- [Stiva OSI](#)
- Stop-and-wait ARQ https://en.wikipedia.org/wiki/Stop-and-wait_ARQ
- <https://www.geeksforgeeks.org/stop-and-wait-arq/>

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 fereastră 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 02 - Folosirea unei legaturi de date pentru transmiterea unui fisier
 - Inainte de laborator
 - Cerinte laborator:
 - Detalii de implementare:
 - Pasi de rezolvare:
 - Software disponibil:
 - Utilitare necesare
 - Alte resurse