

I. Noțiuni teoretice

Socket

- un tip de comunicare între un server și un client, în care datele sunt transmise în timp real (*live*).
- datele sunt transmise la milisecundă de la server la client, sau invers (comunicație duplex).
- există două tipuri de socket-uri:
 1. **datagram** - conectează două noduri pentru transmiterea și primirea pachetelor (asemănător unei căsuțe poștale);
 2. **stream socket** - o comunicare între procese care are o conexiune orientată și un singur flux al datelor; se definește clar modalitatea prin care se creează conexiunea, se distruge și se tratează erorile.

Protocolul de Control al Transmisiei(TCP)

- un protocol folosit de obicei de aplicații care au nevoie de confirmare de primire a datelor (asigură livrarea în ordine a unui flux de octeți de la un program de pe un calculator către un alt program de pe un alt calculator).

Congestia rețelei

- apare când numărul de pachete transmise prin rețea este aproape de capacitatea de manipulare a rețelei.
- se manifestă prin întârzieri, pierderea datelor sau blocarea unor noi conexiuni.

Controlul congestiei

- presupune păstrarea numărului de pachete sub nivelul la care performanța scade dramatic.

Fereastra de congestie

- pentru fiecare conexiune realizată, TCP-ul reține câte o fereastră de congestionare, limitând astfel numărul de pachete care pot fi transmise de la un capăt la altul.
- este menținută de expeditor și este un mijloc de a opri o legătură între expeditor și receptor de la supraîncărcarea traficului.

Algoritmul Slow Start

- la început, fereastra de congestie are o dimensiune mică, dar, pe măsură ce sunt primite confirmări pentru primirea pachetelor (ACK), dimensiunea acesteia se dublează, crescând astfel exponențial.
- atunci când se atinge o anumită valoare de prag a ferestrei de congestie, dimensiunea acesteia va crește liniar. (factorul de limitare, sau **ssthresh**)

Retransmisia rapidă

- reduce timpul de așteptare al expeditorului înainte de retransmiterea unui pachet pierdut.
- dacă nu se primește o confirmare pentru un anumit segment într-un timp specificat (o funcție a timpului estimat de întârziere dus-întors), expeditorul va presupune că segmentul s-a pierdut în rețea și îl va retransmite.
- confirmarea duplicat este baza mecanismului de retransmitere rapidă: după primirea

unui pachet, se trimite o confirmare pentru ultimul octet, în ordinea datelor primite. Atunci când un expeditor primește trei confirmări duplicate, își dă seama că segmentele care au fost trimise după aceasta s-au pierdut.

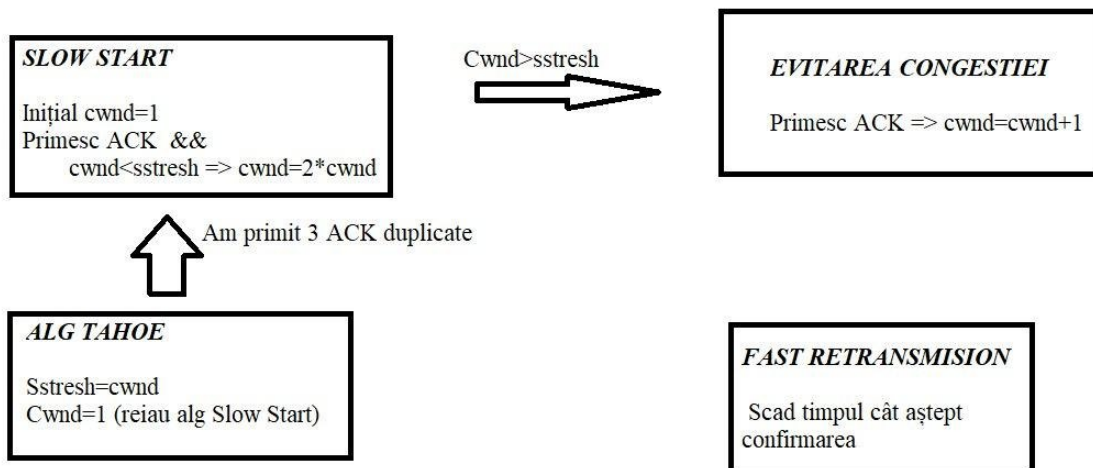
Algoritmul Tahoe

- când apare o pierdere, se folosește retransmisia rapidă, **sstresh** devine dimensiunea curentă a ferestrei, se reia algoritmul **Slow Start**, cu dimensiunea inițială a ferestrei.
- când se ajunge la pragul **sstresh**, TCP schimbă algoritmul de evitare a congestiei astfel încât dimensiunea **cwnd** crește cu $SMS/CWMD$ (creștere liniară).
- dacă se primesc trei ACK-uri duplicate (adică patru ACK-uri care recunosc același pachet, care nu sunt salvate pe date și nu modifică fereastra publicitară a receptorului), Tahoe efectuează o retransmisie rapidă, setează pragul de pornire lentă la jumătate din congestia curentă fereastră, reduce fereastra de congestie la 1 MSS și se resetează la starea de **Slow Start**.

User Datagram Protocol

- face posibilă livrarea mesajelor într-o rețea utilizând o comunicație fără conexiune; pachetele sunt transmise fără confirmare de primire.
- nu dispune nici de mecanisme de verificare de sosire a datagramelor sau a duplicatelor.

II. Proiectarea aplicației



Figură 1-Legătura dintre algoritmii folosiți

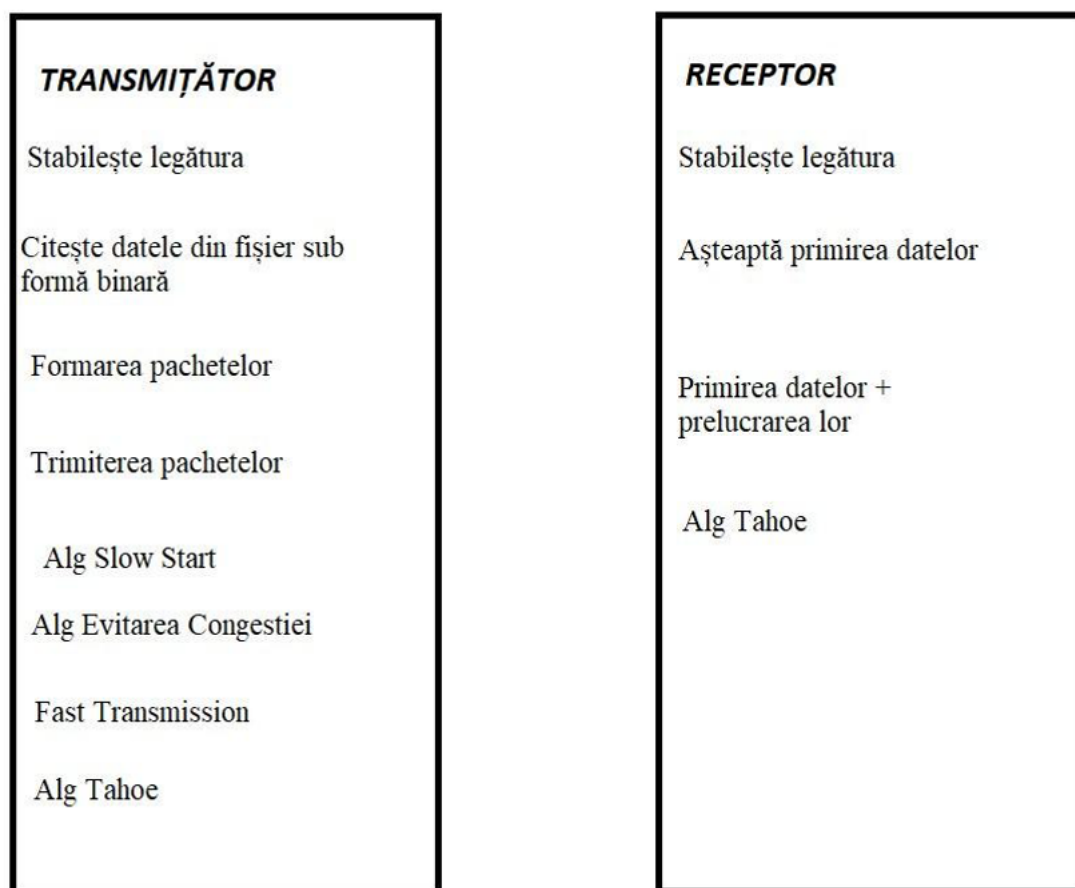


Figura 2 – Schema aplicației

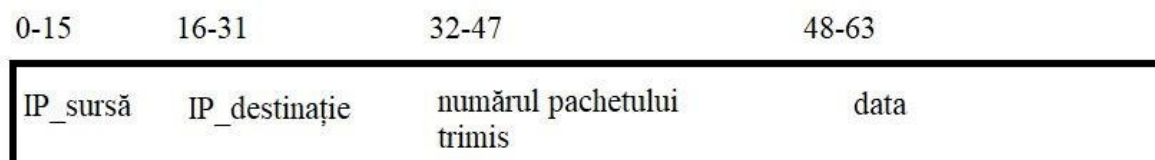


Figura 1 – Formatul pachetelor

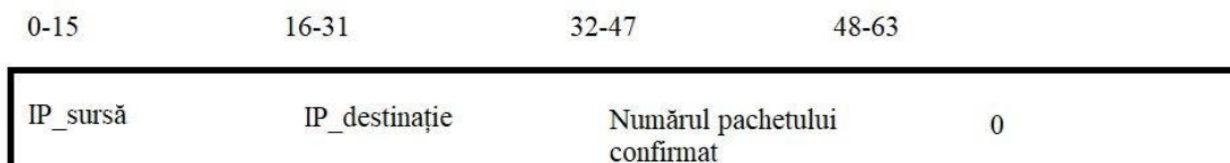


Figura 2 – Formatul confirmărilor

Pentru realizarea aplicației, vom folosi:

- limbajul de programare Python
- biblioteca **socket** pentru comunicarea între cele două entități
- biblioteca **tkinter** pentru realizarea interfețelor grafice

III. Referințe bibliografice

https://en.wikipedia.org/wiki/TCP_congestion_control
https://en.wikipedia.org/wiki/User_Datagram_Protocol