

Universitatea Tehnică „Gheorghe Asachi” Iași

Facultatea de Automatică și Calculatoare

TRANSFER FIȘIERE – FEREASTRĂ GLISANTĂ

DOCUMENTAȚIE

Disciplină:

Rețele de calculatoare – proiect

Student:

Chihalău Adrian-Ioan

1. Protocolul de transfer fișiere fereastră glisantă(Sliding window protocol)

1.1. Considerații generale

Acest protocol constă în ideea de a transmite de la un sender (emițător) un pachet (un șir de informații), împreună cu alte date care asigură receiver-ului (receptorului) primirea corectă al pachetului (de exemplu, un checksum, un bloc de date pentru detecția erorilor în timpul transmiterii datelor). Când receiver-ul verifică datele, trimite un semnal de confirmare(ACK) înapoi la sender pentru a putea transmite următorul pachet. Folosind protocolul ARQ(automatic repeat request), sender-ul se oprește pentru fiecare pachet și așteaptă de la receiver semnalul ACK. Acest lucru asigură trimiterea pachetelor în ordinea corectă, deoarece doar unul poate fi trimis odată.

Totuși timpul în care este primit semnalul ACK poate fi mai mare decât timpul necesar trimiterii unui pachet, iar în consecință, cantitatea de pachete procesate este mult mai mică decât ar trebui posibilă. Astfel protocolul sliding window permite selecția a unui număr de pachete, o „fereastră”, pentru a fi trimisă fără a mai aștepta un semnal ACK. Fiecare pachet primește un număr de secvență, iar semnalele ACK vor trimite înapoi numărul. Protocolul ține evidența pachetelor care au fost confirmate, iar când ele sunt primite, sunt trimise alte pachete. În acest mod, fereastră „glisează” printr-un șir de pachete care realizează transferul, și asigură ca evitarea supraaglomerării rețelei.

1.2. Metoda selective repeat fără Nack

Este cel mai general caz de protocol fereastră glisantă, care necesită un receiver mult mai capabil. Atât sender-ul, cât și receiver-ul vor avea aceeași dimensiune a ferestrei de pachete (în mod obișnuit, este aleasă dimensiunea 2, însă în aplicația proiectului va putea fi mărită dimensiunea ferestrei). Sender-ul va trimite câte un pachet spre receiver, pornindu-se în același timp un timer de expirare (timeout). Dacă nu este primit un semnal de ACK de la receiver înaintea expirării timpului, se trimite din nou pachetul și se resetează timpul timeout.

Când un pachet ajunge în receiver, se va trimite un semnal ACK către sender, iar fereastră receiver-ului se va muta cu o poziție la dreapta; astfel se va asigura primirea pachetelor de receiver. Când semnalul de ACK ajunge în sender înainte de expirarea timpului timeout, sender-ul va considera că pachetul a fost preluat cu succes de receiver și va muta fereastră lui cu o poziție la dreapta.

2. Descriere proiect

Proiectul constă în realizarea unei aplicații care va simula transferul de fișiere între 2 calculatoare conectate la aceeași rețea, prin intermediul protocolului sliding window, folosind metoda selective repeat fără Nack. Va avea 2 view-uri: una pentru transmisie, și una pentru recepție, iar comunicația va fi implementată prin diagrame UDP. View-ul pentru recepție va permite „pierderea” voită a pachetelor pentru a putea demonstra funcționalitatea mecanismului de retransmitere a pachetelor. De asemenea, se vor putea configura parametrii de funcționare (dimensiune fereastră. temporizări etc.).

Aplicația va fi scrisă în limbajul de programare Python, care are la dispoziție biblioteca sockets pentru comunicarea între calculatoare.

2.1. Dimensiune pachet

Un pachet va conține un cuvânt de maxim 30 litere, fără caractere albe, având dimensiunea de 30 octeti.

2.2. Comunicația între mașini

Conexiunea se va realiza prin datagrame UDP. Acestea reprezintă un protocol de comunicație pentru calculatoare, făcând posibilă livrarea mesajelor într-o rețea. UDP nu dispune de mecanisme de verificare a siguranței datelor, nici pentru ordinea de sosire corectă a mesajelor sau de detectare a duplicatelor, dar dispune de sume de control pentru verificarea integrității datelor sau informații privind numărul portului pentru adresarea diferitelor funcții la sursa/destinație.

Caracteristici UDP:

- **orientat către tranzacții** (util în aplicații simple de tip întrebare-răspuns);
- **foarte util în aplicații de configurări** (ex. DHCP);
- **lipsa întârzierilor de retransmisie**, pentru aplicații în timp real (VoIP, jocuri online);
- **lucrează excelent în medii de comunicații unidirecționale** precum furnizarea de informații broadcast, sau în partajarea de informații către alte noduri.

Antetul UDP este alcătuit din 4 câmpuri fiecare având lungimea de 2 octeți:

- **Portul sursa** - în adresarea bazată pe IPv4 acest câmp este opțional. Dacă nu este utilizat acest câmp, are valoarea zero; când reprezintă informație semnificativă, el va indica portul inițiator al procesului de transmisie a datagramelor.

Biti	0 - 15	16 - 32
0	Portul sursa	Portul destinație
32	Lungime	Suma de control
64	Date	

- **Portul destinație** - spre deosebire de portul sursa, câmpul este obligatoriu și indica portul de recepție
- **Lungime** - acest câmp indica lungimea în octeți a datagramelor: antet plus secțiune de date (valoarea minimă a lungimii este 8).
- **Suma de control** - asigură imunitatea la erori; se calculează ca fiind complementul față de 1 (pe 16 biți) a pseudo-antetului cu informații extrase din antetul IP, antetului UDP și a câmpului de date, eventual se completează cu zerouri pentru a atinge lungimea stabilită.

//TO DO: stabilire IPv4 sau IPv6

Aplicația aferentă va avea acces la stiva de comunicație a rețelei folosind biblioteca sockets din Python.

2.3. Vizualizare transmisie/recepție

După cerința proiectului, vizualizarea transmisiei și a recepției se va realiza prin 2 view-uri separate. Acestea vor prelua de asemenea toate informațiile despre transmisia/recepția dintre calculatoare (portul sursa, portul destinație, număr pachet, informația transmisă, data, ora) și vor fi scrise în 2 fișiere log (transmit.log, respectiv recept.log).

În view-ul de recepție, se vor putea **modifica** următorii **parametrii de funcționare**:

- **end to end delay** (durata în care un pachet ajunge de la o stație la alta – unitate de măsură: ms – milisecunde, implicit 4000 ms);
- **timeout** (timpul de expirare pentru așteptarea unui răspuns de ACK de la receiver către sender – ms, implicit 5000 ms);
- **window size** (dimensiunea ferestrelor de transmisie și recepție, minim 2, maxim 10, implicit 2);
- **number of packets/min** (număr de pachete transmise de stații pe minut, minim 20, maxim 100).
- **packet failure chance** (șansa de a „pierde” un pachet, un număr cuprins în intervalul [0,1], implicit 0,1).

Ultimul parametru de funcționare va permite „pierderea” voită a unor pachete pentru demonstrarea funcționării mecanismului.

2.4. Interfața grafică

Interfața grafică pentru care va fi permisă demonstrarea tuturor modurilor de funcționare va fi realizată folosind interfața PyQt.

//TO DO: detalii despre interfață

3. Implementare

//TO DO: implementare proiect