

**Ministerul Educației, Culturii și Cercetării Universitatea Tehnică a Moldovei
Facultatea Calculatoare, Informatică și Microelectronică Departamentul Ingineria
Software și Automatică**

RAPORT

la Lucrarea de Laborator Nr. 5
Disciplina: Programarea în Rețea
TEMA: Aplicație Client-Server UDP

A elaborat:

st. gr. TI-172, Movileanu Vladislav

A verificat:

asistent universitar, Buldumac Oleg

Chișinău 2020

Sarcina lucrării:

Să se creeze o aplicație Client-Server UDP utilizând Sockets API

Pentru nota 8, 9 si 10:

- Să se creeze **GUI(Graphical User Interface)** pentru aplicație

Atenție:

- Nu se admite aplicații simple de genul Echo Client-Server

Raspunsuri la întrebări de control:

- Ce este un protocol orientat către tranzații, fără conexiune?
Un protocol orientat către tranzații este un protocol orientat pe tranzații bazat pe un transfer minim de segmente, deci nu are probleme de viteză asociate cu TCP.
- Ce tipuri de aplicații beneficiază în general de utilizarea protocolului UDP?
UDP este utilizat în mod obișnuit pentru aplicațiile cu pierderi (pot face față unor pierderi de pachete), precum streamingul audio și video. Este de asemenea utilizat pentru aplicații de răspuns la interogare, cum ar fi interogări DNS.
- De ce protocolul UDP nu garantează că datele vor fi transmise cu succes?
UDP nu garantează că datele vor fi transmise cu succes, fiindcă acesta nu efectuează controlul fluxului, controlul erorilor sau retransmiterea unui segment rău.
- Diferența dintre blocking si non-blocking sockets?
În modul blocking, apelurile API socket, recv, send, connect (doar TCP) și accept (doar TCP) vor bloca la nesfârșit până când acțiunea solicitată a fost efectuată. În modul non-blocking, aceste funcții revin imediat. select se va bloca până când socket-ul este gata.
- Diferența dintre blocking multithreaded și non-blocking single thread socket.
Un non-blocking single thread socket permite operarea I/O pe un canal fără a bloca procesele care îl utilizează. Aceasta înseamnă că putem folosi un singur fir pentru a gestiona mai multe conexiuni concurente și pentru a obține o operație de I/O „asincronă de înaltă performanță”.
- În protocolul TCP există Three Way Handshake, de ce în UDP nu există?
UDP este fără conexiune, rezultă că UDP nu stabilește conexiuni așa cum face TCP, deaceia UDP nu realizează Three Way Handshake și din acest motiv este considerat un protocol nesigur.
- Numiți cele 2 apeluri de sistem necesare pentru a crea un server UDP?
 - 1) Folosind create(), Creăm un UDP socket;
 - 2) Folosind bind(), legăm socket-ul la adresa serverului;

- Care este rolul metodei `bind()`?
Metoda `bind()` în programarea socket-ului este utilizată pentru a asocia socketul la adresa locală, adică adresa IP, portul și familia de adrese.
- Care este rolul metodelor `sendto()` și `recvfrom()`?
1) Metoda `sendto()`, trimite datagrame pe socket cu ajutorul descriptorului;
2) Metoda `recvfrom`, primește date pe un socket și le stochează într-un buffer;
- Care este dimensiunea antetului unui pachet UDP în octeți?
Mărimea unui antet a unui pachet UDP este de 8 octeți.
- Într-o conexiune UDP, clientul sau serverul trimite mai întâi datele?
În dependență de arhitectura serverului, dar cel mai des clientul trimite mai întâi datele.
- Care este adresa de loopback IPv6 și care este rolul ei?
O adresă de loopback IPv6(127.0.0.1) este folosită ca mijloc de validare a faptului că placa de rețea fizică conectată local funcționează corect și stiva TCP / IP este instalată.
- De unde știe un sistem de operare ce aplicație este responsabilă pentru un pachet primit din rețea?
Port source și Port destination sunt utilizate pentru a afla ce aplicație este responsabilă pentru un pachet primit din rețea.
- Datele primite prin `recv()` au întotdeauna aceeași dimensiune cu datele trimise cu `send()`?
Datele primite prin `recv()` nu au întotdeauna aceeași dimensiune cu datele trimise cu `send()`.
- Este acceptabil să închei execuția programului dacă este detectată o eroare de rețea?
Da, este acceptabil să închei execuția programului dacă este detectată o eroare de rețea.
- De ce nu este folosit algoritmul Nagle în protocolul UDP?
Algoritmul Nagle funcționează practic prin întârzierea intenționată a pachetelor. Făcând acest lucru, crește eficiența lății de bandă și latența, rezultă că se creează o pauză, care în UDP nu este nevoie, fiindcă acesta nu efectuează controlul fluxului, controlul erorilor sau retransmiterea unui segment rău.
- Ce instrumente listează socket-urile UDP deschise în sistemele de operare Windows și Linux?
Windows - <https://www.howtogeek.com/howto/28609/how-can-i-tell-what-is-listening-on-a-tcpip-port-in-windows/>
Linux/MacOS - netsat (<https://www.tecmint.com/find-open-ports-in-linux/>)

- Același program poate folosi UDP și TCP?
Da, același program poate utiliza același număr de port atât pentru TCP, cât și pentru UDP.
- Diferența dintre aplicații UDP Unicast, Broadcast, și Multicast.
Broadcast-urile sunt trimise la toate posturile din rețea. Unicast este trimis numai unei stații din rețea. Multicast-urile sunt trimise unui grup de stații, de exemplu, mai multe camere de tip video.
- Ce face mai ușor multiplexarea cu UDP decât cu TCP?
Viteza de transmitere și mărimea pachetelor, face mai ușor multiplexarea cu UDP decât cu TCP.
- În protocolul UDP este un antet „Total length”, cum se calculează și care este rolul lui?
Antetul “Total length” este lungimea antetului UDP plus datele UDP. Este redundant, deoarece această lungime poate fi calculată din câmpul lungimii totale a antetului IP în care lungimea datagramelor UDP este lungimea totală IP minus lungimea antetului IP. Deoarece UDP poate fi transmis prin alt protocol decât IP.

Descrierea Aplicației:

Am utilizat limbajul **GO(golang)**, pentru a crea un live-streaming—server.

Utilizând limbajul **Swift**, am realizat o aplicație de tip **Twitch**.

Am utilizat **fire de execuție, socket și data chunking**.

Aplicația permite utilizatorului să transmită flux de intrare a camerei dispozitivului, către un server utilizând o conexiune de tipul UDP.

Este de menționat faptul că chiar dacă cineva străin va obține acest flux, acesta nu va putea primi imaginile datorită encodingului utilizat de **Swift**, pentru crea imaginea de tipul quartz.