

## Tema 2

### Implementarea protocolului Reliable User Datagram Protocol (RUDP)

Peste headerul de UDP care are 4 campuri: port sursa si destinatie, checksum si header length, in payload vom utiliza primii 5 bytes ca sa reprezentam headerul de RUDP.

Acesta va contine:

- Sequence number pe 2 bytes
- Acknowledgement number pe 2 bytes
- Flags ( SYN, SEQ, ACK, PSH, FIN ) primii 5 biti din al 5-lea byte al headerului, restul de 3 raman nefolositi

Tema trebuie sa contina un client si un server de RUDP care sa poata face urmatoarele:

- 3 way handshake ca la TCP la inceputul conexiunii
- trimiterea pachetelor intr-un mod reliable (simulati cu comanda de pierderea pachetelor ca functioneaza corect, i.e. trimiteti 10 pachete cu packet loss 40%, pachetele pierdute se retrimite si serverul primeste 10 pachete in total, asa voi verifica si eu ca merge) folosind SEQ si ACK la fel ca TCP
- seteaza corect flag-urile ( SYN cand incepe conexiunea, PUSH cand se trimite date, FIN cand se doreste incheierea conexiunii)
- incheie conexiunea cu FIN la fel ca la TCP

Observatii:

- datele trimise vor avea dimensiuni mai mici de 100 bytes ca sa nu avem fragmentarea datelor
- sequence number trebuie sa fie unic pentru fiecare pachet (cum vreti voi, in ordine crescatoare, uint16(sha256(unixtimestamp + last sequence no)) etc .. )
- clientul are sequence number diferit fata de server

Explicatie: Folosim UDP, toate aceste procesari se fac in payload-ul de UDP la application layer. Implementati in ce limbaj doriti, atata timp cat respectati cele 4 cerinte (sugerez python).

Tema se poate prezenta doar fizic cu explicarea codului pana pe data de 13.04.2022 inclusiv. Daca din anumite motive cineva chiar nu poate veni in aceste 2 saptamani, accept sa primesc pe mail [andrei.ciobanu@unibuc.ro](mailto:andrei.ciobanu@unibuc.ro) rezolvarea pana la deadline si sa prezinte ulterior.

Header RUDP! Header UDP

Data

Seq no	Ack no	Port Src	Port Dst
Flags		Checksum	Length

Tot data pt  
UDP