



POUZDAN PRENOS PODATAKA PREKO UDP PROTOKOLA

OSNOVI RAČUNARSKIH MREŽA 1

DANILO MALAVRAZIĆ RA138/2017

KRSTO ZARIĆ RA229/2020

ZADATAK

- ▶ Realizovati sistem za pouzdan prenos podataka posredstvom UDP transportnog protokola.
- ▶ Potrebno je implementirati mehanizme potvrde i retransmisije.
- ▶ Posle svakog poslatog paketa podataka, aplikacija pokrece vremensku kontrolu i od druge strane čega potvrdu da je dobila paket.
- ▶ Primjer se sastoji iz klijentske strane (sender.c) i serverske strane (reciever.c).

UDP protokol

- ▶ UDP (engl. User Datagram Protocol) protokol definiše uslugu nepouzdanu isporuku korisničkih podataka (datagrama) preko IP-a. Konceptualno, jedina bitna razlika između UDP datagrama i IP datagrama je u tome što UDP sadrži brojeve portova, što omogućava predajnoj aplikaciji da se obrati tačno određenoj aplikaciji na odredišnoj mašini.



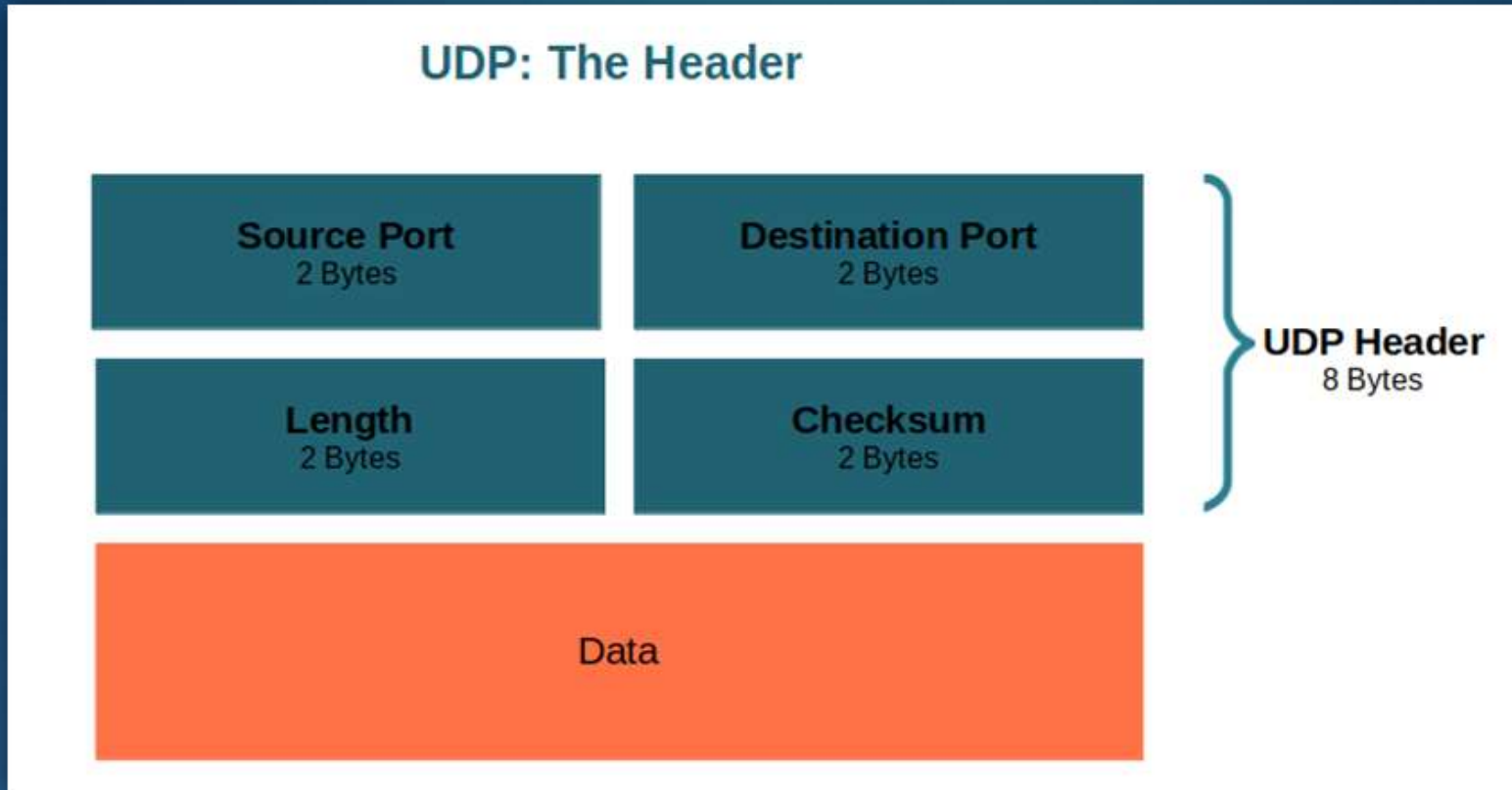
Osobine UDP-a:

- ▶ Nepouzdana isporuka poruka
- ▶ Nije obavezna uspostava konekcije pre slanja podataka.
- ▶ Ne koristi sequence number, acknowledgment number, niti obavlja kontrolu toka podataka. Ne postoji zaštita od zagušenja prijemnika velikim brojem poruka.
- ▶ Kada se pošalju podaci, nema načina da se sazna da li su stigli ili da li su stigli redosledom kojim su poslani. Međutim, podaci koji stižu, stižu brzo što je bitno za neke vrste servisa tj. aplikacija kojima je brzina važnija od pouzdanosti isporuke podataka (npr. real-time audio i video).

Oblik UDP poruke

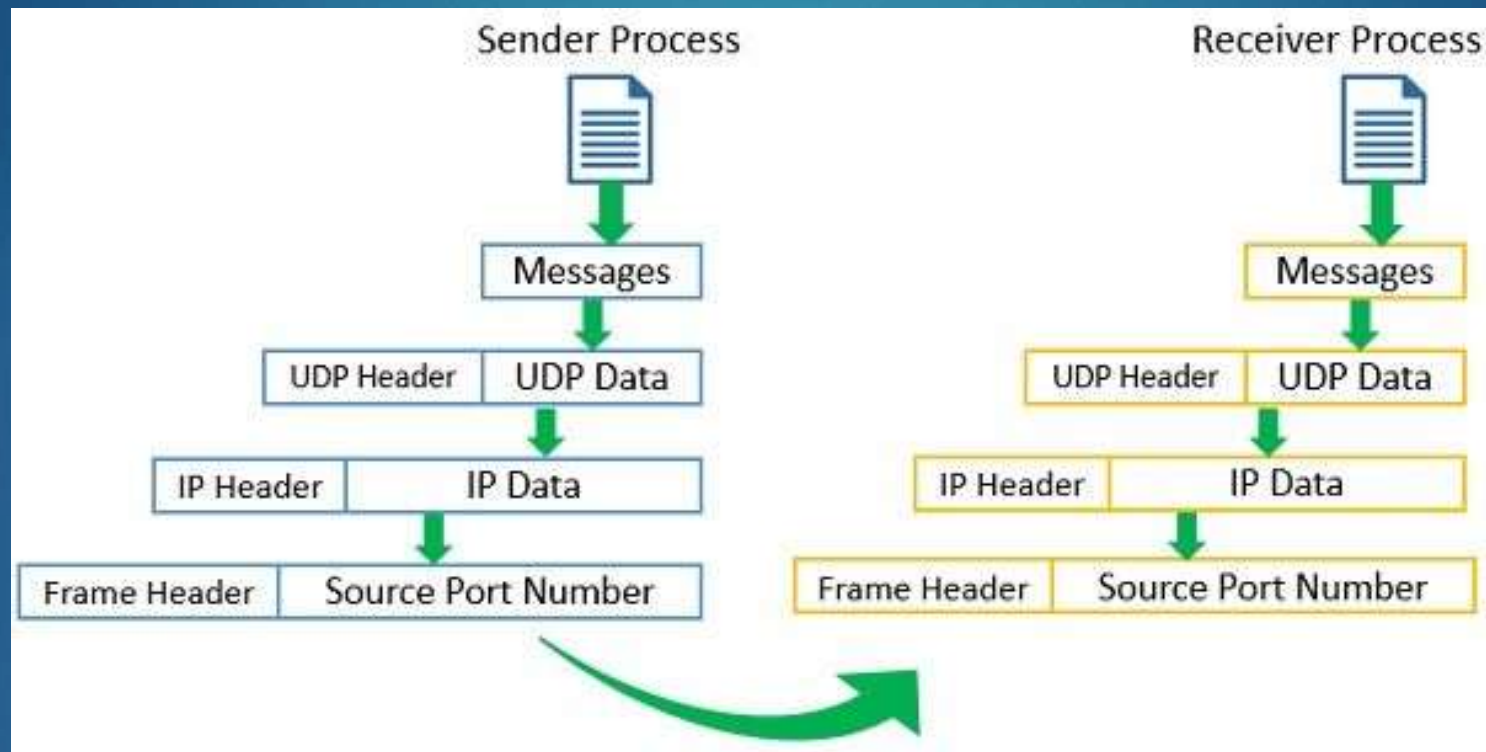
- ▶ Polje “UDP SOURCE PORT” je neobavezno. Ako se koristi, specificira prolaz na koji treba slati odgovore. Ako se ne koristi, treba da bude 0. Pošto se portovi zadaju kao 2- bajtni neoznačeni brojevi, 65536 različitih mogućih UDP portova je dostupno po hostu.
- ▶ Polje “UDP MESSAGE LENGTH” sadrži broj okteta u UDP datagramu, uključujući zaglavlje i korisničke podatke. Dužina je takođe 2-bajtni neoznačeni cijeli broj, pa je broj bajtova u datagram-u ograničen na 65536-8 za zaglavlje. U praksi je veličina znatno manja.
- ▶ Polje “UDP CHECKSUM” sadrži kontrolnu sumu koja se izračunava nad pseudozaglavljem UDP datagrama. Ako se kontrolna suma ne poklopi, native mrežni software tiho odbaci datagram, ne obavještavajući o tome ni pošiljaoca ni primaoca.
- ▶ Količina podataka u UDP datagramu je na mnogim platformama ograničena na 8192 bajta (8K). Implementacije ne moraju prihvatiti datagrame sa više od 576 bajtova ukupno, uključujući podatke i zaglavlja.

Oblik UDP poruke



Enkapsulacija UDP poruke

- ▶ UDP poruka se smešta u IP datagram, a on u fizički okvir



Receiver.c

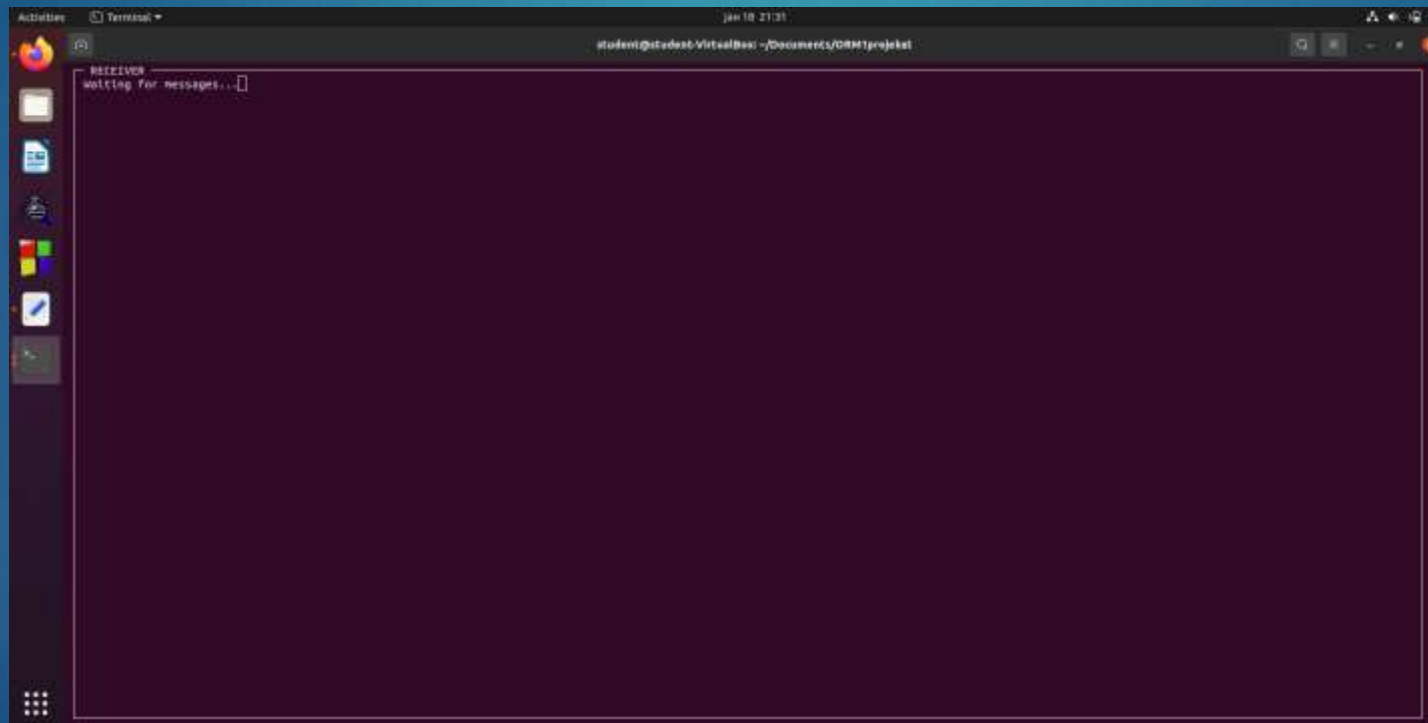
- ▶ Ovaj kod predstavlja primjer UDP servera koji sluša na određenom portu (8080) i prihvata poruke od klijenta koji šalje poruke na taj port.
- ▶ Definišemo funkcije za čišćenje, dobijanje trenutnog vremena, provjere veličine terminala, inicijalizaciju datoteke za ispis, ispis poruke u datoteku i provjeru paketa.
- ▶ U glavnom dijelu koda, se inicijalizuje socket, podešavaju parametri za servaddr i cliaddr strukture, te se koristi funkcija `bind()` da se socket veže na određeni IP i port.
- ▶ Koristi se `ncurses` biblioteka za inicijalizaciju prozora i prikaz poruka.
- ▶ Koristi se funkcija `recvfrom()` za primanje poruka i ispis primljenih poruka na terminalu.
- ▶ Kod se završava sa funkcijom `cleanUp()` koja zatvara socket i ispisuje poruku o uspješnom završetku programa.

Sender.c

- ▶ Ovaj kod predstavlja implementaciju klijenta UDP protokola.
- ▶ Koristi se struktura podataka "Header" koja sadrži informacije o tipu poruke, ID-u i veličini podataka.
- ▶ Takođe se koristi struktura podataka "Package" koja sadrži "Header" i podatke.
- ▶ Kod se sastoji od više funkcija koje se koriste za različite aktivnosti kao što su: zatvaranje socket-a, oslobađanje memorije, provera veličine terminala, provera ACK-a, itd.
- ▶ Klijent šalje poruke serveru iz datoteke "message.txt" i čeka potvrdu (ACK) od servera. Ukoliko se ne dobije potvrda unutar određenog vremena ili ako potvrda nije ispravna, klijent će ponoviti slanje poruke.

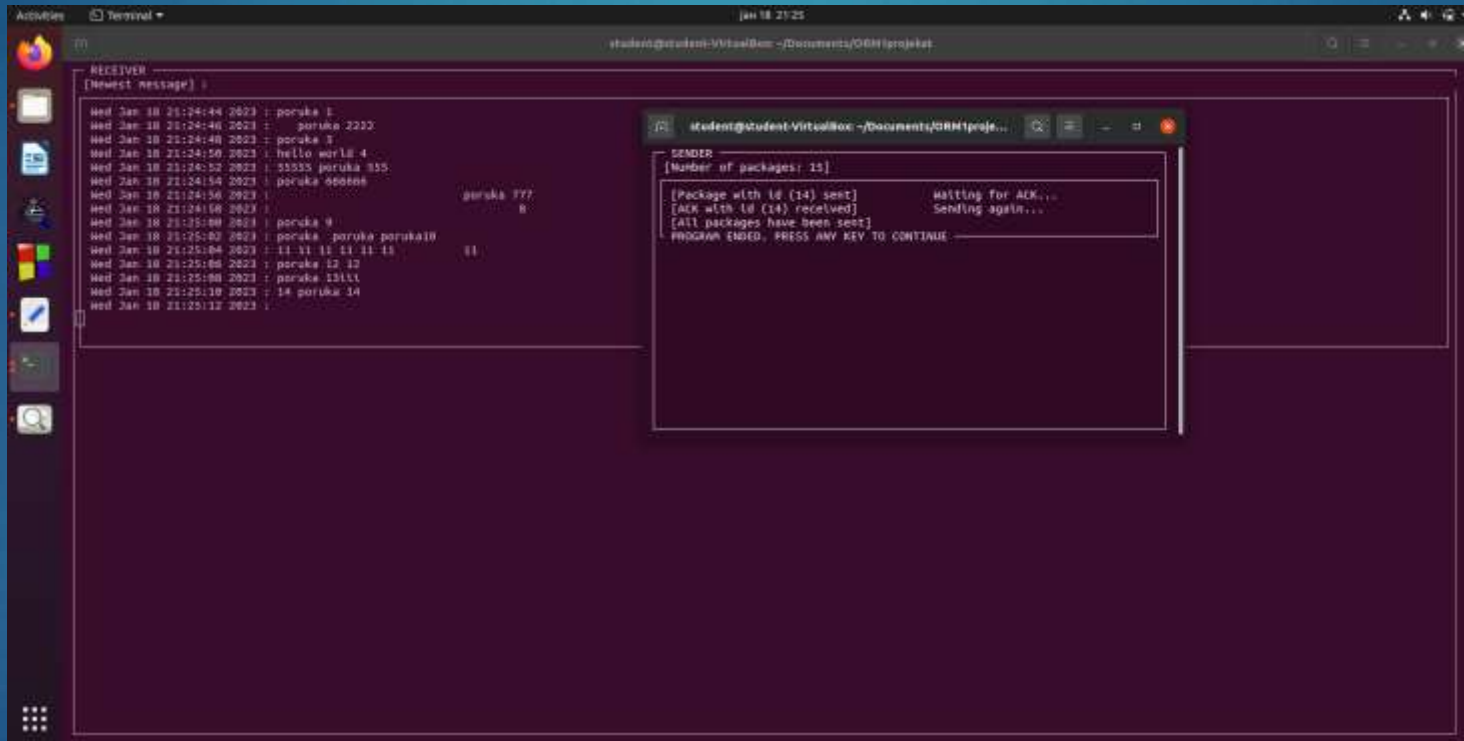
Aplikacija

- ▶ Aplikacija se sastoji od datoteka **receiver.c** i **receivedMessage.txt** sa serverske strane, a **sender.c** i **message.txt** sa klijentske strane.
- ▶ Komandom `./receiver` prvo pokrećemo server i dobijamo poruku `Waiting for messages...`



Aplikacija

- ▶ Nakon pokretanja servera pokrećemo klijentsku stranu komandom `./sender` koja šalje poruke iz `message.txt` serveru.



```
RECEIVER
[Newest message]
Wed Jan 10 21:24:44 2023 : poruka 1
Wed Jan 10 21:24:46 2023 : poruka 2222
Wed Jan 10 21:24:48 2023 : poruka 3
Wed Jan 10 21:24:50 2023 : hello world 4
Wed Jan 10 21:24:52 2023 : 55555 poruka 555
Wed Jan 10 21:24:54 2023 : poruka 999999
Wed Jan 10 21:24:56 2023 : poruka 777
Wed Jan 10 21:24:58 2023 : 8
Wed Jan 10 21:25:00 2023 : poruka 9
Wed Jan 10 21:25:02 2023 : poruka poruka poruka10
Wed Jan 10 21:25:04 2023 : 11 11 11 11 11 11
Wed Jan 10 21:25:06 2023 : poruka 12 12
Wed Jan 10 21:25:08 2023 : poruka 13111
Wed Jan 10 21:25:10 2023 : 14 poruka 14
Wed Jan 10 21:25:12 2023 :

SENDER
[Number of packages: 15]
[Package with id (14) sent]      Waiting for ACK...
[ACK with id (14) received]     Sending again...
[All packages have been sent]
PROGRAM ENDED, PRESS ANY KEY TO CONTINUE
```

Aplikacija

- ▶ **message.txt** predstavlja paket podataka koji klijent šalje
- ▶ **receivedMessage.txt** predstavlja paket podataka koje je server primio

```
1 poruka 1
2      poruka 2222
3 poruka 3
4 hello world 4
5 55555 poruka 555
6 poruka 66666
7
8      poruka 777
9      8
10 poruka 9
11 poruka poruka poruka10
12 11 11 11 11 11 11
13 poruka 12 12
14 poruka 13111
15 14 14 poruka 14
16
```

Loading file "/home/student/Documents/ORM/projekat/message.txt"

message.txt

```
1 ===== NEW RECEIVED MESSAGES =====
2 Wed Jan 18 21:27:08 2023 : poruka 1
3 Wed Jan 18 21:27:05 2023 : poruka 1
4 Wed Jan 18 21:27:07 2023 :      poruka 2222
5 Wed Jan 18 21:27:09 2023 : poruka 3
6 Wed Jan 18 21:27:11 2023 : hello world 4
7 Wed Jan 18 21:27:13 2023 : 55555 poruka 555
8 Wed Jan 18 21:27:15 2023 : poruka 66666
9 Wed Jan 18 21:27:17 2023 :
10 Wed Jan 18 21:27:19 2023 :      poruka 777
11 Wed Jan 18 21:27:21 2023 :      8
12 Wed Jan 18 21:27:23 2023 : poruka 9
13 Wed Jan 18 21:27:25 2023 : poruka poruka poruka10
14 Wed Jan 18 21:27:27 2023 : 11 11 11 11 11 11
15 Wed Jan 18 21:27:29 2023 : poruka 12 12
16 Wed Jan 18 21:27:31 2023 : poruka 13111
17 Wed Jan 18 21:27:33 2023 : 14 14 poruka 14
```

Saving file "/home/student/Documents/ORM/projekat/receivedMessage.txt"

receivedMessage.txt

Hvala vam na pažnji!