

1. domaći zadatak - paketna obrada podataka

Deo 1:

- Posetiti link <http://mmklab.fon.bg.ac.rs/wp-content/rmt/domaci1.php>
- Pratiti uputstva na tom linku i analizirati saobraćaj koji je nastao.

Deo 2:

- Upotrebom alatke *nslookup* (ili odgovarajuće alternative - *dig* ili *host*) analizirati mrežni saobraćaj DNS protokola. Posebnu pažnju obratiti na uslugu nepouzdanog prenosa koju u ovom scenariju korišćenja pruža transportni sloj.

Uputstvo za odbranu:

- Pripremiti se za pitanja vezana za 4 sloja TCP/IP referentnog modela (aplikativni, transportni, mrežni, sloj veze podataka)
- Korišćenjem softverskog paketa Wireshark uraditi statističku analizu toka pouzdane konekcije i prikazati je grafički.
- Pre pristupanja izradi domaćeg zadatka preporučuje se proučavanje zadataka iz praktikuma
- Na odbranu poneti fajl koji sadrži saobraćaj snimljen korišćenjem Wireshark softverskog paketa (.pcap fajl)

2. domaći zadatak - mrežno programiranje

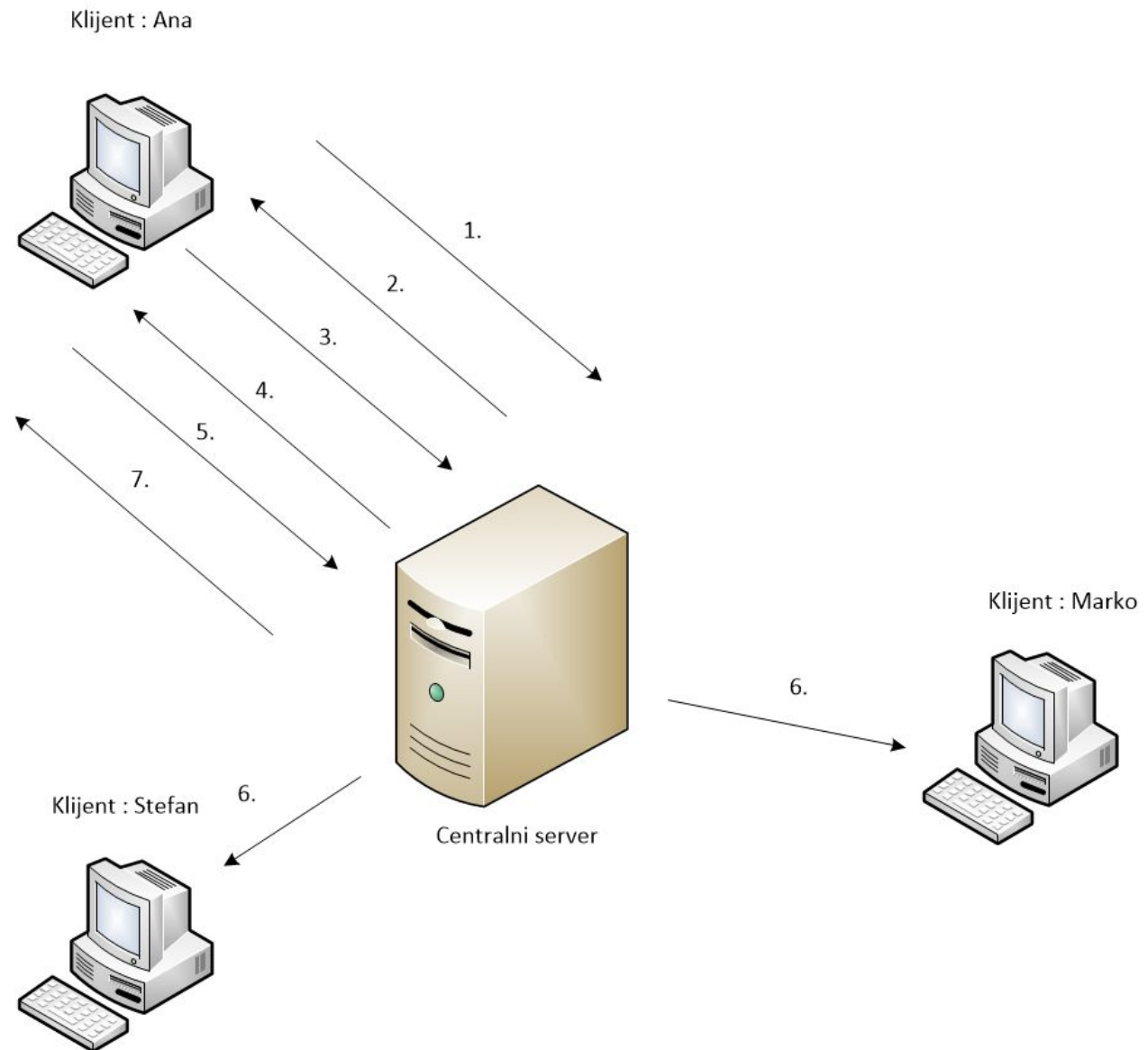
Osnovni zahtevi:

Potrebno je napraviti klijent server chat aplikaciju koja omogućava instant slanje poruka većem broju korisnika chat aplikacije. S obzirom da je u pitanju klijent server aplikacija, postoji centralni server i svi korisnici chat aplikacije znaju njegovu IP adresu. Na centralnom serveru se čuva lista korisnika koji su trenutno online. Prilikom prijavljivanja na aplikaciju, klijent je potrebno da unese svoje ime i pol (muški ili ženski). Nakon što se klijent poveže na server treba omogućiti klijentu da šalje poruku ljudima koje on izabere, ali pri tome postoji jedno ograničenje - muškarci mogu samo ženama slati poruke i obratno.

Za ostvarivanje osnovnih zahteva ovog domaćeg zadatka, potrebno je implementirati sledeći scenario komunikacije preko jedne TCP veze:

1. Klijent se povezuje na server.
2. Server šalje poruku o uspešnom povezivanju
3. Klijent šalje chat poruku.
4. Server šalje listu klijenta koji su trenutno povezani sa serverom i odgovarajućeg su pola u odnosu na pol klijenta.
5. Klijent bira osobe kojima želi da pošalje poruku, i šalje listu serveru
6. Server prosledjuje poruku odabranim klijentima
7. Server obaveštava klijenta po uspešno završenom slanju

Grafički prikaz osnovnog scenarija:



Dodatni zahtevi:

- Potrebno je da server proverava da li je moguće poslati poruku svim korisnicima koji se nalaze u listi(možda neko od klijenata nije više online, klijent nije potvrdio prijem) i onima kojima jeste šalje poruku. Ukoliko neko od slanja poruke ka drugim klijentima nije uspeo, potrebno je obavestiti klijenta čija poruka se šalje.
- Slanje liste korisnika sa servera ka klijentu se vrši pomoću transportnog protokola UDP. Ostatak komunikacije potrebno je realizovati pomoću transportnog protokola TCP.
- Serverska aplikacija treba da u tekstualnom fajlu pamti sve poruke zajedno sa vremenom pristizanja, primaozima, kao eventualnim problemima pri slanju.

NAPOMENE:

Potrebno je da klijentska aplikacija sadrži korisnički interfejs koji je u skladu sa zahtevima domaćeg zadatka. Nije neophodno da interfejs bude grafički (npr. Swing u slučaju JAVA implementacije), akcenat je na protokolu i mrežnoj komunikaciji. Serverska aplikacija i klijentska aplikacija su **odvojeni projekti**. Svi projekti moraju biti razvijeni na takav način da budu otporni na greške u radu i komunikaciji.

Uputstvo za odbranu:

- Aplikaciju je poželjno implementirati u JAVA ili Python programskom jeziku. U slučaju implementacije u nekom drugom programskom jeziku, na odbranu poneti i svoj laptop sa okruženjem u kome je aplikacija razvijana.
- Zadatak se radi samostalno. U slučaju plagijata, zadatak neće biti priznat ni plagijatoru ni originalnom autoru.
- Uslov za pristup odbrani zadatka je aplikacija koju je moguće kompajlirati bez prijave greške.
- Na odbranu obavezno poneti source kod aplikacije.