



УНИВЕРЗИТЕТ У НОВОМ САДУ ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА



УНИВЕРЗИТЕТ У НОВОМ САДУ
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА
НОВИ САД
Департман за рачунарство и аутоматику
Одсек за рачунарску технику и рачунарске комуникације

ИСПИТНИ РАД

Кандидат: Михаило Бабић
Број индекса: РА52/2020

Предмет: Међурачунарске комуникације и рачунарске мреже 1
Тема рада: Интерактивна комуникација на ЛАН
мрежи (децентрализована)

Ментор рада: Проф. Илија Башичевић

Нови Сад, Децембар 2022.

SADRŽAJ

1. Zadatak.....	1
2. Koncept rešenja.....	2
3. Opis rešenja.....	3
3.1 SDL Dijagrami	3
3.2 MSC Dijagram	4
4. Testiranje	5
5. Zaključak	7
6. Literatura.....	8

SPISAK SLIKA

<i>Slika 1</i>	3
<i>Slika 2</i>	4
<i>Slika 3</i>	4
<i>Slika 4</i>	5
<i>Slika 5</i>	6
<i>Slika 6</i>	6

1. Zadatak

Realizovati chat program tipa >>svaki sa svakim<<, bez postojanja centralne tacke razmene poruka.

Klijent po startovanju mora da otkrije sve ostale klijente u odgovarajucoj podmrezi. Otkrivanje aktivnih klijenata se može obaviti slanjem inicijalne poruke svakom računaru u podmreži, na zadatom portu.

Zatim, ostvaruje konekciju sa svakim otkrivenim klijentom. Svaki klijent mora da poseduje ime, koje se prosledjuje kao parametar u komandnoj liniji. Klijent odlazne poruke šalje svakom klijentu sa kojim je povezan. Poruke je potrebno ispisati na ekran, sa podatkom od koga je poruka dobijena. Odgovarajućim ispisom na ekranu potrebno je obavestiti korisnika o događajima kao što su pojava novog klijenta ili isključivanje klijenta.

2. Koncept rešenja

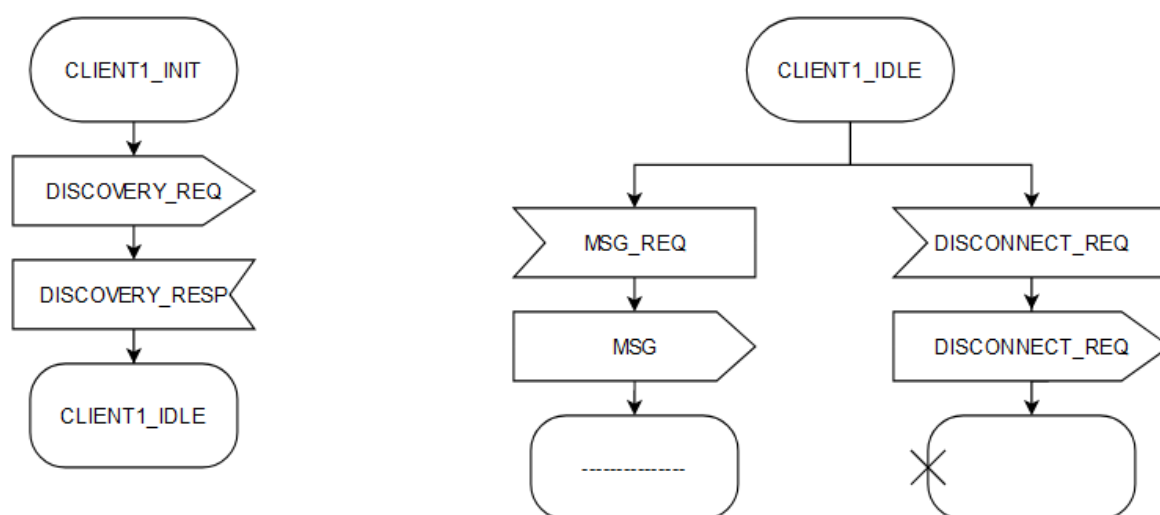
Rešenje je realizovano u QT razvojnom okruženju upotrebom signala i slotova za komunikaciju između objekata i UDP soketa. Svaki klijent otkriva ostale pomoću discovery paketa koji se šalje svima datoj grupi. Prilikom prijema discovery poruka svaki klijent šalje svoje korisnicko ime originalnom pošaljiocu poruke, na osnovu koje pošaljiac pravi uvid u to koji se svi korisnici nalaze u mreži.

Nakon što je pribavio listu korisnika, svaki klijent dalje vodi evidenciju o novim korisnicima, kao i o korisnicima koji su mrežu napustili

Svaki klijent šalje poruke svima u mreži. I ispisuje ime korisnika od koga je primio datu poruku, a napušta mrežu slanjem odgovarajuće poruke na ulaz u terminalu.

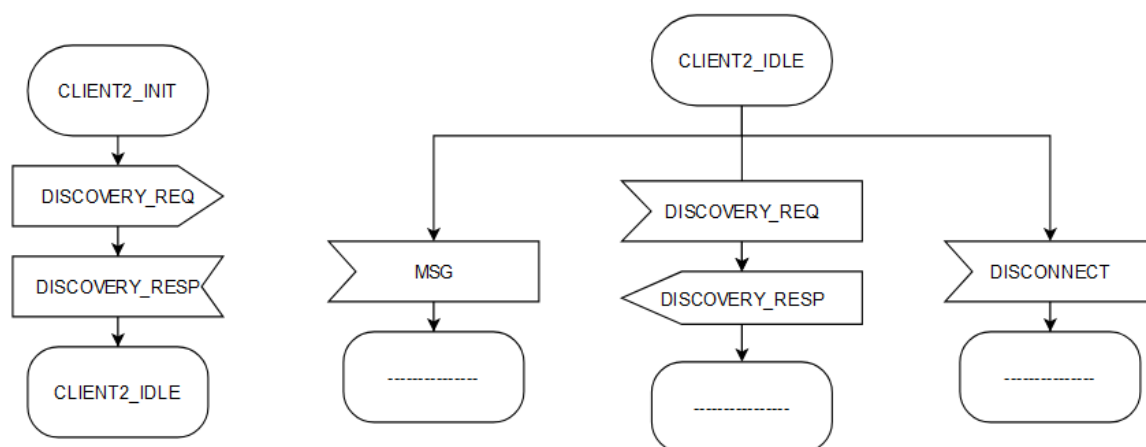
3. Opis rešenja

3.1 SDL Dijagrami



Slika 1

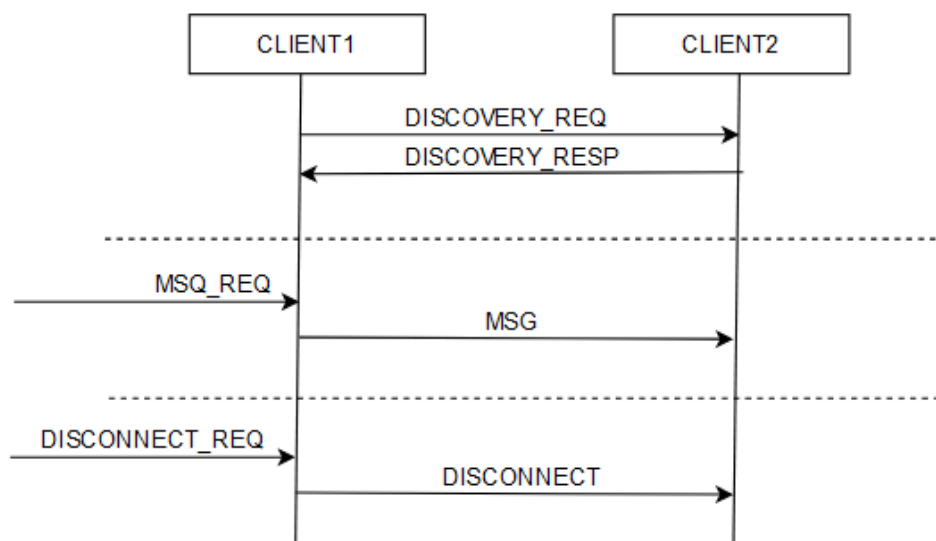
Na slici iznad su prikazana stanja novog korisnika koji se priključuje u mrežu.



Slika 2

Na slici iznad su prikazana stanja korisnika koji se već nalazi na mreži, i prima dolazne poruke postojećih korisnika i poruke otkrivanja novih korisnika.

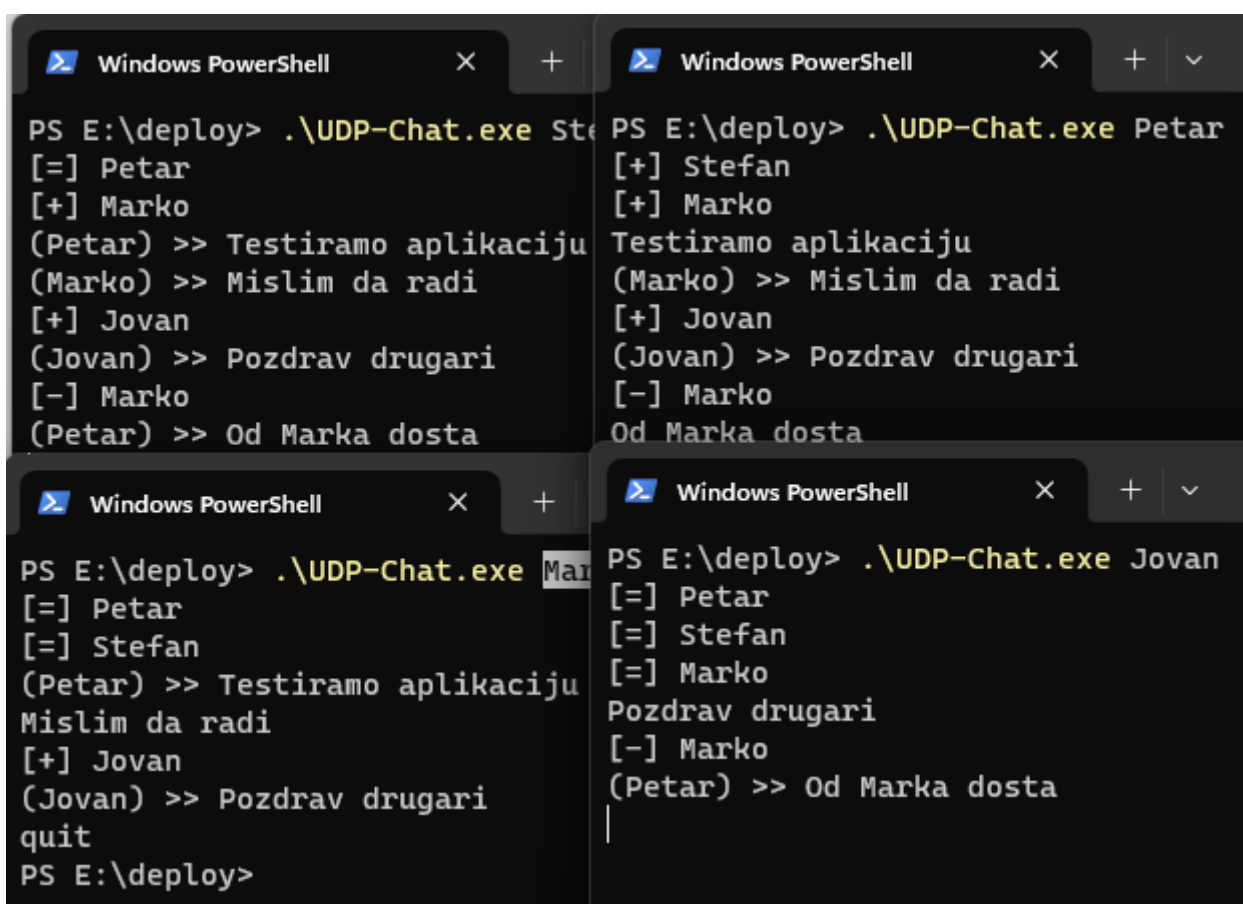
3.2 MSC Dijagram



Slika 3

Dijagram prikazuje sve moguće tipove poruka i komunikacija između klijenata u mreži

4. Testiranje



```
PS E:\deploy> .\UDP-Chat.exe Stefan
[=] Petar
[+] Marko
(Petar) >> Testiramo aplikaciju
(Marko) >> Mislim da radi
[+] Jovan
(Jovan) >> Pozdrav drugari
[-] Marko
(Petar) >> Od Marka dosta

PS E:\deploy> .\UDP-Chat.exe Petar
[+] Stefan
[+] Marko
Testiramo aplikaciju
(Marko) >> Mislim da radi
[+] Jovan
(Jovan) >> Pozdrav drugari
[-] Marko
Od Marka dosta

PS E:\deploy> .\UDP-Chat.exe Marko
[=] Petar
[=] Stefan
(Petar) >> Testiramo aplikaciju
Mislim da radi
[+] Jovan
(Jovan) >> Pozdrav drugari
quit
PS E:\deploy>

PS E:\deploy> .\UDP-Chat.exe Jovan
[=] Petar
[=] Stefan
[=] Marko
Pozdrav drugari
[-] Marko
(Petar) >> Od Marka dosta
|
```

Slika 4

Na slici se može videti stanje aplikacije kada je veći broj korisnika na mreži I uspešnu komunikaciju među njima, kao I obaveštenja o postojećim korisnicima pri priključivanju na mrežu, novim korisnicima kao I o korisnicima koji su mrežu napustili.


```

PS E:\deploy> .\UDP-Chat.exe
Usage: E:\deploy\UDP-Chat.exe <username>
PS E:\deploy> |

```

Slika 5

Na slici se vidi kada korisnik pokušava da pokrene aplikaciju bez adekvatnih parametara komandne linije

udp && ip.src==127.0.0.1						
No.	Time	Source	Destination	Protocol	Length	Info
63	45.162803	127.0.0.1	255.255.255.255	UDP	48	52622 → 12345 Len=16
88	70.109571	127.0.0.1	255.255.255.255	UDP	47	52623 → 12345 Len=15
89	70.112769	127.0.0.1	127.0.0.1	UDP	53	52622 → 52623 Len=21
96	80.067696	127.0.0.1	255.255.255.255	UDP	47	52624 → 12345 Len=15
97	80.068005	127.0.0.1	127.0.0.1	UDP	53	52622 → 52624 Len=21
98	80.068051	127.0.0.1	127.0.0.1	UDP	52	52623 → 52624 Len=20
113	85.240364	127.0.0.1	255.255.255.255	UDP	47	52625 → 12345 Len=15
114	85.240704	127.0.0.1	127.0.0.1	UDP	52	52623 → 52625 Len=20
115	85.240749	127.0.0.1	127.0.0.1	UDP	53	52622 → 52625 Len=21
116	85.240759	127.0.0.1	127.0.0.1	UDP	52	52624 → 52625 Len=20
117	87.726374	127.0.0.1	255.255.255.255	UDP	42	52625 → 12345 Len=10
491	298.078504	127.0.0.1	127.0.0.1	UDP	40	52623 → 52622 Len=8
492	298.078562	127.0.0.1	127.0.0.1	UDP	40	52623 → 52624 Len=8

Slika 6

Ovde pratimo tok podataka na mreži uz pomoć alata Wireshark, gde možemo da vidimo podatke klijenata koji se priključuju/napuštaju mrežu koji se šalju na broadcast adresu, kao i komunikaciju između samih klijenata između sebe.

5. Zaključak

QT Framework, njegova implementacija signala i soketa, kao i veliki broj urađenih tipova i funkcija dosta olakšava razvijanje programa. UDP komunikacija je vrlo pogodna za otkrivanje velikog broja klijenata na mreži kako zbog mogućnosti slanja jednog paketa na više klijenata tako i zbog mogućnosti slanja paketa bez uspostavljanja konekcije.

6. Literatura

- [1] *Qt-manual, Vežbe 6*
- [2] Dokumentacija QT Frameworka, <https://doc.qt.io/qt-6>