

VIDEO CHAT APLIKACIJA ZASNOVANA NA WEBRTC PROTOKOLU I STUN/TURN SERVERIMA**VIDEO CHAT APPLICATION BASED ON WEBRTC PROTOCOL AND STUN/TURN SERVERS**Nikola Marić, Milan Vidaković, *Fakultet tehničkih nauka, Novi Sad***Oblast – RAČUNARSTVO I AUTOMATIKA**

Kratak sadržaj – Predmet rada je omogućavanje audio i video poziva između dve osobe (dva korisnika aplikacije) u realnom vremenu, dakle da se podrži vid komunikacije između fizički odvojenih (udaljenih) mesta na planeti koja nisu iza istog rutera. Način da se to postigne jeste korišćenjem WebRTC protokola kao i STUN/TURN servera koji omogućavaju prenos podataka na daljinu.

Ključne reči: WebRTC, STUN, TURN, Video chat aplikacija

Abstract – Subject of this paper is to provide audio and video call between two persons (two users of this application) in a real-time, therefore to support type of communication between physically separated (distant) places on planet that are not behind the same router. Way to do that is to use a WebRTC protocol and STUN/TURN servers that provide transfer of data to a distant place.

Keywords: WebRTC, STUN, TURN, Video chat application

1. UVOD

Usled trenutne situacije na globalnom nivou, sve je veća potražnja za softverskim rešenjima koja vrše neki vid komunikacije u realnom vremenu, naročito vizuelnog/audio tipa. Sve veći broj odnosa između ljudi kao što su zabava, obrazovanje pa čak i posao se digitalizuju i svima su nam dostupni na jedan pritisak tastera ili pokret prsta. Da bi to sve uspešno funkcionisalo i ljudi bili up-to-date sa stanjem u okruženju, potrebno je prenositi informacije u realnom vremenu.

Prenos informacija unutar jedne internet mreže ili jednog rutera je relativno jednostavno. Problem nastaje kada internet saobraćaj treba da se uspostavi između uređaja koji funkcionišu iza različitih rutera, tada se ceo proces komplikuje.

U zavisnosti od tipa rutera ceo taj proces može da iziskuje i velike troškove, stoga je potrebno izabrati korektno rešenje za specifičnu situaciju.

Celokupan sistem se sastoji iz tri dela: web aplikacije, STUN/TURN servera i baze podataka.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Milan Vidaković, red. prof.

Web aplikacija je implementirana u React/Node okruženju, STUN/TURN serveri su podignuti na AWS Linux server. Baza podataka je također podignuta na AWS u PostgreSQL tehnologiji.

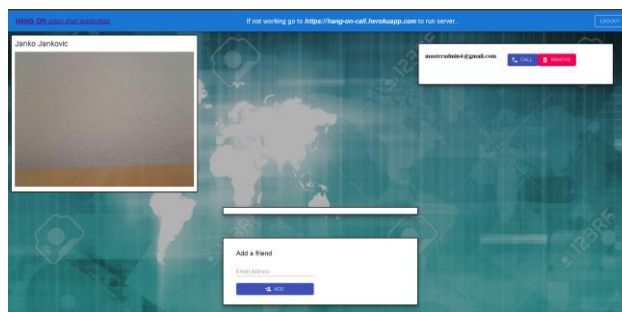
2. IMPLEMENTACIJA APLIKACIJE

Web aplikacija se sastoji iz klijentskog i serverskog dela. Klijentski deo je implementiran u React-u [1], a serverski u Node.js [2] okruženju. Serverski deo možemo podeliti na signaling server i serverski deo sa endpointima koji vrše operacije sa bazom podataka.

Celokupno rešenje je podignuto na poznate cloud servise. Klijentski deo je podignut na Netlify [3], a serverski deo (signaling server i endpointi) je podignut na Heroku [4].

2.1. Web aplikacija

Nakon prijave na sistem u gornjem levom uglu se nalazi video snimak ulogovanog korisnika u realnom vremenu. Iznad snimka se nalazi ime i prezime korisnika, ili ako nije uneseno onda nadimak (Slika 1).



Slika 1. Početna stranica nakon prijave na sistem

Nakon učitavanja snimka korisnika koji je ulogovan, opcija koja mu se dalje nameće jeste dodavanje prijatelja (Slika 2).

Add a friend

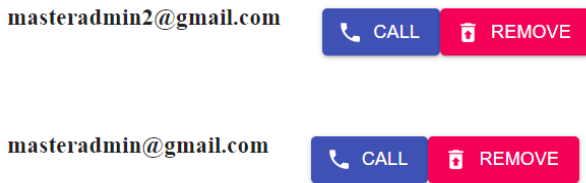
Email address



Slika 2. Forma za dodavanje prijatelja

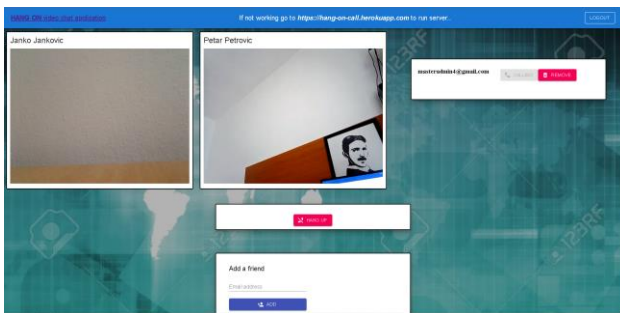
Korisnik unosi email adresu drugog korisnika koga želi da doda u listu prijatelja, nakon čega može da obavlja poziv sa njim.

Nakon što je korisnik uspešno dodat u prijatelje, pojaviće se njegov email u gornjoj desnoj strani ekrana (Slika 3). Korisnik ima opcije da pozove prijatelja (dugme CALL) ili da ga obriše iz liste (dugme REMOVE).



Slika 3. Lista sa spiskom prijatelja

U slučaju poziva, pored video snimka naše kamere, u sredini ekrana će se nakon što korisnik pozove drugog korisnika, a ovaj drugi prihvati poziv, pojaviti još jedan video stream korisnika koji je pozvan (Slika 4).



Slika 4. Početna strane nakon prihvatanja poziva

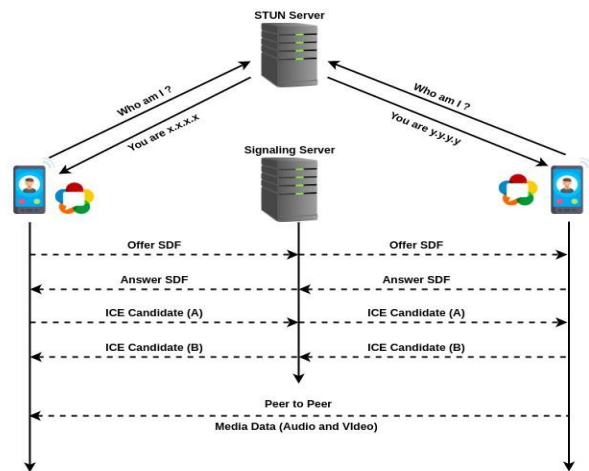
Nakon što je korisnik pozvan i pod uslovom da je prihvatio poziv, pojavljuje se dugme HANG UP. Klikom na to dugme konekcija sa drugim korisnikom se raskida i stranica se refrešuje. Klik na dugme REMOVE briše korisnika iz liste prijatelja, nakon čega se također radi refreš stranice. Poslednja preostala funkcionalnost je odjava korisnika sa sistema odnosno logout. Sve što radimo jeste brisanje email-a trenutnog korisnika iz sessionStorage-a iz browser-a i refrešovanje aplikacije na početnu stranicu.

2.2. STUN/TURN serveri

STUN/TURN su implementacije ICE [5] protokola. ICE je zapravo tehnologija pomoću koje ćemo zaobići NAT [6] ili Firewall [7]. Postoji mnogo verzija na internetu, github stranicama itd. Opcija koja je ovde implementirana jeste **coturn** server koji se jednostavno instalira na Linux okruženju i kao takav podigne na neku cloud platformu, ovde je to AWS (Amazon Web Services)[8].

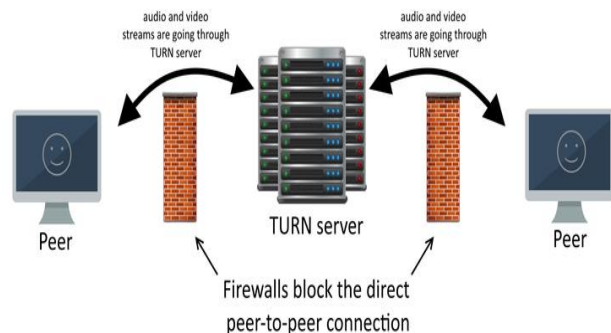
Nakon što je kreirana web aplikacija koja funkcioniše iza jednog rutera, jednog NAT-a ili Firewall-a, potrebno je omogućiti globalnu komunikaciju korisnika. To ćemo postići preko ICE protokola i njegovih implementacija. STUN server [9] služi da se njemu obrati naša aplikacija i da zatraži svoju javnu ip adresu. Dakle, kad korisnik započinje poziv web aplikacija se obraća STUN serveru, šalje mu zahtev, a STUN server kao odgovor vraća javnu IP adresu te web aplikacije odnosno računara/uređaja na kome se nalazi. Sa druge strane, korisnik koji prima poziv

takođe se obraća STUN serveru i ovaj mu kao odgovor vraća njegovu javnu IP adresu. Nakon toga može da se izvrši direktna konekcija peer-to-peer (Slika 5).



Slika 5. Funkcija STUN servera prilikom konekcije

Kad STUN server ne može da izvrši svoju namenu tada se koristi TURN [10]. Zajedno su podignuti, na istom serveru. TURN se uglavnom koristi kad se naide na simetrični NAT. Tada TURN server služi kao relej i celokupan sadržaj prenosi preko sebe do drugog peer-a. Ovaj postupak je skuplji (veći bandwidth) i dolazi do relativnog kašnjenja komunikacije jer postoji posrednik tako da se koristi u krajnjim slučajevima (Slika 6).



Slika 6. Uloga TURN servera prilikom komunikacije

Kako je podešen STUN/TURN? Prvo se na AWS kreira nalog. Nakon toga kreiramo EC2 [11] (Elastic Compute Cloud) virtuelnu mašinu. Tu se podese razni parametri. Konkretno, u svrhe projekta izabran je Ubuntu server sa ostalim predefinisanim vrednostima. Nakon što se EC2 instanca pokrene, potrebno je u Linux Ubuntu operativni sistem instalirati coturn paket i odraditi dodatnu konfiguraciju. Komande idu sledećim redosledom:

Prvo update-ujemo Ubuntu:

```
sudo apt-get -y update
```

Nakon toga instaliramo coturn paket

```
sudo apt-get install coturn
```

Nakon što se instalira coturn, proveravamo njegov trenutni status:

```
systemctl status coturn
```

Ako je pokrenut potrebno ga je zaustaviti

```
systemctl stop coturn
```

Nakon što smo zaustavili coturn otvaramo coturn fajl i u njemu obrišemo znak komentara (#) sa linije gde piše `TURN_SERVER_ENABLED=1`

```
sudo nano /etc/default/coturn
```

Potom treba da podešavamo glavnu konfiguraciju. Prvo kopiramo postojeću konfiguraciju u novi fajl u slučaju da nam zatreba backup.

```
sudo mv /etc/turnserver.conf  
/etc/turnserver.conf.original
```

Otvaramo original `turnserver.conf`

```
sudo nano /etc/turnserver.conf
```

Nakon što se fajl otvori, potrebno je podesiti konfiguraciju kao na listingu 1. Postavljamo portove koji će da slušaju na 3478 (običan tcp/udp) i 5349 (za tls). Podešavamo ime servera odnosno DNS naše cloud instance Linux servera (može i IP adresa) uz dodatnu konfiguraciju. User predstavlja `username:lozinka` par koji se podešava kao kredencijali za TURN server.

```
listening-port=3478  
tls-listening-port=5349
```

```
fingerprint  
lt-cred-mech
```

```
server-name=ec2-44-203-187-  
183.compute-1.amazonaws.com  
realm=ec2-44-203-187-183.compute-  
1.amazonaws.com  
user=admin:admin
```

```
total-quota=100  
stale-nonce=600
```

*Listing 1. Podešavanja u fajlu na putanji
/etc/turnserver.conf*

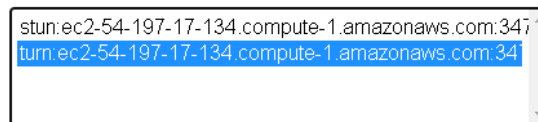
Nakon što smo uneli konfiguracione parametre, pokrećemo coturn:

```
systemctl start coturn
```

Nakon što smo završili sa Ubuntu komandama, potrebno je podesiti Security Groups, odnosno mehanizam koji omogućava/onemogućava pristup našem DNS-u sa strane. Da bi web aplikacija mogla da koristi podešeni coturn servis potrebno je postaviti inbound rules u okviru bezbednosne grupe. Potrebno je omogućiti pristup portu 3478 i za svaki slučaj spektru portova od 49152 do 65535. Nakon što je ovo sve podešeno, možemo testirati da li serveri rade ispravno. Postoji nekoliko klijenata na

internetu koji ovo testiraju, a jedan od poznatijih je Trickle Ice (Slika 7).

ICE servers



STUN or TURN URI:

TURN username:

TURN password:

Add Server

Remove Server

Reset to defaults

Slika 7. Unos STUN/TURN servera zbog testiranja

Dakle za STUN server je neophodno uneti `stun:<dns>:<port>`. Za TURN je pored URI-ja potrebno uneti kredencijale, dakle username i password koji smo prethodno konfigurisali u `turnserver.conf` fajlu. Kad smo uneli DNS servera možemo testirati. Klikom na dugme Gather candidates (Slika 8) pojaviće se spisak nekih adresa i protokola. Kako bi se uverili da naši serveri rade neophodno je da postoji tip `srflix` (za STUN server) i `relay` (za TURN server).

Time	Component	Type	Foundation	Protocol	Address	Port	Priority	Mid	MLIndex	Username	Flagment
0.006	no host		3111678107	udp	172.24.96.1	63107	128 (32128) / 255	0	0		noMm
0.007	no host		3022648116	udp	192.168.1.4	63108	128 (32128) / 255	0	0		noMm
0.007	no host		2997462051	udp	192.168.56.1	63109	128 (32128) / 255	0	0		noMm
0.110	no host		4104462251	tcp	172.24.96.1	9	90 (32128) / 255	0	0		noMm
0.110	no host		4205478112	tcp	192.168.1.4	9	90 (32128) / 255	0	0		noMm
0.110	no host		4233988103	tcp	192.168.56.1	9	90 (32128) / 255	0	0		noMm
0.210	no srflix		494279629	udp	178.227.222.201	50506	100 (32128) / 255	0	0		noMm
0.499	no relay		238844111	udp	172.31.92.228	50721	0 (32128) / 255	0	0		noMm
39.938											
39.940											

Gather candidates

Note: errors from `onicecandidateerror` above are not necessarily fatal. For example an IPv6 DNS lookup may fail but relay candidates can still be gathered via IPv4.
The server `stun:ec2-54-197-17-134.compute-1.amazonaws.com:3478` returned an error with code=701:
STUN binding request timed out.
The server `stun:ec2-54-197-17-134.compute-1.amazonaws.com:3478` returned an error with code=701:
STUN binding request timed out.

Slika 8. Prikaz uspešnosti konfigurisanih coturn servisa

2.3 Baza podataka

Baza koja je korišćena je PostgreSQL [12]. Nakon što se kreira instanca baze na RDS [13] (Relational Database Service) takođe na AWS-u, moramo podesiti bezbednosni mehanizam kao i za Ubuntu server. Ovde smo podesili da dolazni i odlazni saobraćaj bude dostupan za sve adrese i portove, dakle praktično nema bezbednosnog mehanizma. Nakon što je baza podešena, unosimo entitete. Prvi je `app_user`, entitet koji predstavlja korisnika i podatke o njemu (Listing 2), a drugi je `app_user_friends`, predstavlja tabelu koja sadrži veze između email adresa, odnosno koji korisnik je prijatelj sa kim (Listing 3).

```
CREATE TABLE app_user (  
  id bigint PRIMARY KEY,  
  email varchar NOT NULL,  
  password varchar NOT NULL,  
  first_name varchar,  
  last_name varchar,  
  nickname varchar NOT NULL,  
  date_created timestamptz,  
  current_socket_id varchar  
);
```

Listing 2. SQL kod za kreiranje app_user entiteta

```
CREATE TABLE app_user_friends2 (
  id bigint PRIMARY KEY,
  email_user varchar,
  email_friend varchar
);
```

Listing 3. SQL kod za kreiranje app_user_friends entitet

3. ZAKLJUČAK

Glavna svrha ovog rešenja jeste rešavanje problema komunikacije u realnom vremenu, dakle razmena audio i video podataka. Postoji dosta aplikacija koje rešavaju isti problem. Kao prednost ovog rešenja u odnosu na postojeća komercijalna rešenja može se navesti činjenica da je lako stupiti u video poziv sa drugim korisnikom, a opet ne dozvoliti da neko ko ne treba pristupi pozivu.

Mogući pravci daljeg proširenja ove aplikacije se podudaraju sa manama ove aplikacije, dakle sve ostalo što postojeće aplikacije imaju ovoj nedostaje tako da ima dosta mesta za usavršavanje.

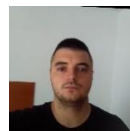
Kao neke glavne osobine koje bi se mogle dodati su pre svega real-time chat-ovanje, grupno chat-ovanje, zatim grupni pozivi, share screen opcija, deljenje datoteka i još mnogo toga.

4. LITERATURA

- [1] ReactJS
<https://reactjs.org/>
- [2] Node.js
<https://nodejs.org/en/about/>
- [3] Netlify
<https://www.netlify.com/>
- [4] Heroku
<https://www.heroku.com/>
- [5] ICE
https://en.wikipedia.org/wiki/Interactive_Connectivity_Establishment

- [6] NAT
https://en.wikipedia.org/wiki/Network_address_translation
- [7] Firewall
[https://en.wikipedia.org/wiki/Firewall_\(computing\)](https://en.wikipedia.org/wiki/Firewall_(computing))
- [8] AWS
<https://aws.amazon.com/what-is-aws/>
- [9] STUN
<https://en.wikipedia.org/wiki/STUN>
- [10] TURN
https://en.wikipedia.org/wiki/Traversal_Using_Relays_around_NAT
- [11] EC2
https://en.wikipedia.org/wiki/Amazon_Elastic_Compute_Cloud
- [12] PostgreSQL
<https://en.wikipedia.org/wiki/PostgreSQL>
- [13] RDS
https://en.wikipedia.org/wiki/Amazon_Relational_Database_Service

Kratka biografija:



Nikola Marić rođen je u Novom Sadu 1996-te godine. Osnovne akademske studije je završio na Fakultetu tehničkih nauka u Novom Sadu, smer Računarstvo i automatika, modul Primenjene računarske nauke i informatika. Kontakt: maricnikola752@gmail.com



Milan Vidaković rođen je u Novom Sadu 1971. Doktorirao je na Fakultetu tehničkih nauka 2003. god, a od 2014. je u zvanju redovni profesor iz oblasti Primenjene računarske nauke i informatika.