



UNIVERZITET U ZENICI
Politehnički fakultet
Softversko inženjerstvo
Multimedijalni sistemi i aplikacije

SIP protokol
Projekat

Studenti:

Belma Delilović, II-113
Ajla Brdarević, II-120
Denis Bičakčić, II-132

Profesor:
v. prof. dr. Goran Nermin

Zenica, januar 2026.

SADRŽAJ

1. UVOD.....	3
2. SIP protokol - Analiza SIP protokola u VOIP i IMS.....	4
2.1. Odnos SIP i RTP.....	4
2.2. SIP URI i request/response model.....	5
2.2.1. Struktura SIP URI-ja.....	5
2.2.2. Request/Response princip.....	5
2.2.3. Osnovne SIP metode.....	5
2.3. SIP transakcije i dijalozi.....	6
2.4. SIP call-flow.....	7
2.5. SIP u IMS arhitekturi.....	9
2.5.1. Uloge P-CSCF, I-CSCF i S-CSCF.....	9
2.5.2. Razlika između klasičnih VoIP sistema i IMS arhitekture.....	10
2.5.3. Sigurnost i autentikacija u IMS sistemima.....	10
3. Setup SIP servisa.....	11
3.1. Konfiguracija SIP korisnika.....	12
3.2. Dialplan konfiguracija.....	16
4. Softphone klijent A (User Agent 1).....	20
4.1. Inicijalizacija SIP stack-a.....	20
4.2. SIP registracija.....	23
4.3. Outgoing call.....	25
5. Softphone klijent B (User Agent 2).....	27
5.1. Inicijalizacija SIP stack-a.....	27
5.2. SIP registracija.....	30
5.3. Incoming call handling.....	33
6. Medij i sesija.....	37
6.1. SDP razmjena.....	37
6.1.1. Gdje se SDP pojavljuje u SIP komunikaciji.....	37
6.1.2. Kako je SDP provjeren.....	38
6.1.3. Primjer SDP ponude (offer) iz SIP INVITE poruke.....	38
6.1.4. Codec ponuda/odgovor (offer/answer).....	39
7. Testiranje i validacija.....	40
7.1. SIP logiranja.....	40
7.2. Wireshark analiza.....	40
8. ZAKLJUČAK.....	42
9. REFERENCE.....	43

1. UVOD

U okviru predmeta obrađuju se osnovni koncepti multimedijalnih sistema i aplikacija, s posebnim naglaskom na komunikacione protokole koji se koriste u savremenim IP mrežama. Jedan od najznačajnijih protokola u oblasti multimedijalnih komunikacija je SIP (Session Initiation Protocol), koji se koristi kao signalizacioni mehanizam u VoIP sistemima, ali i kao sastavni dio IMS (IP Multimedia Subsystem) arhitekture. Razumijevanje načina rada ovog protokola predstavlja osnovu za shvatanje kako se uspostavljaju, održavaju i prekidaju multimedijalne sesije u realnim sistemima.

Zadatak ovog rada je analizirati SIP protokol kao signalizacioni mehanizam u VoIP i IMS sistemima te realizovati praktičnu implementaciju SIP komunikacije između dva vlastita softphone klijenta. Implementacija se zasniva na korištenju open-source alata, pri čemu SIP server ima ulogu posrednika u registraciji korisnika i uspostavi poziva između klijenata.

U teorijskom dijelu rada obrađene su osnovne karakteristike SIP protokola, njegova uloga u komunikacionom procesu, struktura i tipovi SIP poruka, kao i odnos SIP-a sa protokolima za prenos medija. Posebna pažnja posvećena je objašnjenju razlika između klasičnih VoIP sistema i IMS arhitekture, kao i načinu na koji se SIP koristi unutar IMS okruženja.

Praktični dio rada obuhvata postavljanje SIP serverske infrastrukture, konfiguraciju korisničkih naloga i implementaciju dva softphone klijenta koji međusobno komuniciraju putem SIP protokola. Kroz implementaciju je demonstriran kompletan proces signalizacije, uključujući registraciju korisnika, uspostavu poziva i njegov prekid, kao i provjeru ispravnosti rada sistema kroz testiranje i analizu signalizacijskog toka.

Cilj rada je da se kroz kombinaciju teorijske analize i praktične realizacije prikaže način funkcionisanja SIP protokola u VoIP i IMS sistemima, te da se stečeno znanje poveže sa realnim primjerom primjene u obliku funkcionalne SIP komunikacije.

2. SIP protokol - Analiza SIP protokola u VOIP i IMS

SIP (Session Initiation Protocol) je signalizacijski protokol standardiziran od strane IETF-a (RFC 3261), koji se koristi u VoIP i multimedijским komunikacijama za upravljanje sesijama. Nalazi se u aplikacijskom sloju OSI modela (sloj 7), gdje djeluje iznad transportnog sloja (obično UDP ili TCP). [1][2]

SIP omogućuje stvaranje, modifikaciju i prekidanje komunikacijskih sesija između dvije ili više stranaka, uključujući glasovne/video pozive, instant poruke i multicast sesije. Protokol upravlja signalizacijom, lociranjem korisnika putem registrar servera i pregovara medijske parametre (npr. kodeke) putem SDP-a. Koristi se u sustavima poput IP telefonije i video konferencija. [2]

SIP ne prenosi stvarne medijske podatke (kao audio/video tokove) - za to se koriste protokoli poput RTP (Real-time Transport Protocol). Ne bavi se naplatom, osiguranjem kvalitete usluge (QoS), sigurnosno kodiranjem sadržaja niti rutinom paketa - te funkcije prepustio drugim protokolima (npr. RTCP za statistike, SRTP za sigurnost). [2]

U OSI modelu, SIP pripada aplikacijskom sloju (Layer 7), sličan HTTP-u ili FTP-u, gdje definira logiku za korisničke sesije. Transportni sloj (Layer 4, TCP/UDP) prenosi SIP poruke, dok niži slojevi (npr. IP na mrežnom sloju 3) rute pakete. [2][3]

2.1. Odnos SIP i RTP

SIP služi za uspostavu, upravljanje i prekidanje sesija u VoIP i multimedijским komunikacijama, RTP osigurava prijenos medijskih tokova (audio/video), a SDP djeluje kao most između signalizacije i medija opisujući parametre poput kodeka, portova i IP adresa. [4][5]

Uloge protokola

- SIP: Pokreće sesiju putem poruka poput INVITE i BYE, registruje korisnike i pregovara uvjete, ali ne prenosi medijske podatke. [4][6]

- RTP: Transportni protokol za realno-vremenski prijenos enkodiranih audio/video paketa preko UDP-a, bez signalizacije. [4][6]
- SDP: Format unutar SIP poruka koji specificira medijske detalje (vrste medija, formate, protokole), omogućavajući pregovor prije RTP prijenosa. [4][5]

SIP ugrađuje SDP u poruke (npr. INVITE) za dogovor parametara, nakon čega RTP započinje prijenos na dogovorenim portovima. SDP čini SIP sposobnim za medijske specifikacije, dok RTP brine samo o efikasnom toku bez upravljanja sesijom. [4][6]

2.2. SIP URI i request/response model

SIP URI identificira korisnike i resurse u SIP mrežama, a protokol koristi tekstualni request/response model sličan HTTP-u za komunikaciju. Osnovne metode poput REGISTER-a, INVITE-a i BYE-a omogućavaju registraciju, uspostavu i prekid sesije. [7]

2.2.1. Struktura SIP URI-ja

SIP URI slijedi format sip:korisnik@domena:port;parametri?headeri, gdje korisnik može biti telefon ili ime, domena označava server (npr. proxy ili registrar), port je opcionalan (zadano 5060/5061), parametri specificiraju transport (udp/tcp/tls), a headeri prosljeđuju dodatne informacije. Primjer: sip:korisnik@primjer.com;transport=tcp. Koristi se za adresiranje u pozivima i registracijama.

2.2.2. Request/Response princip

SIP djeluje na tekstualnim porukama: request (zahtjev) šalje User Agent Client (UAC) prema User Agent Serveru (UAS), koji vraća response (odgovor) – provisionalni (napredak, npr. 180 Ringing) ili finalni (rezultat, npr. 200 OK). Poruke su čitljive ASCII, sa start line-om (metoda/URI za request, status za response), headerima (From, To, Call-ID) i opcionalnim tijelom (SDP). Pouzdanost se postiže retransmisijama (UDP) ili TCP-om. [7]

2.2.3. Osnovne SIP metode

- REGISTER: Korisnik se registrira kod registrar servera šaljući svoj IP/contact info, server ažurira location service za doseg. [8]
- INVITE: Pokreće sesiju (poziv), sadrži SDP za pregovor medija; generira provisionalne odgovore (npr. 100 Trying) i finalni (200 OK). [7]

- BYE: Zatvara sesiju sa jedne strane, potvrđuje se 200 OK; omogućava graciozni prekid komunikacije.

2.3. SIP transakcije i dijalozi

SIP transakcije i dijalog predstavljaju ključne koncepte za upravljanje komunikacijom u SIP-u, gdje transakcija obuhvata jedan zahtjev i povezane odgovore, a dijalog niz takvih transakcija. [9]

2.3.1. Razlika transakcija vs dijalog

Transakcija je osnovna jedinica: sastoji se od jednog SIP zahtjeva (npr. INVITE) i svih odgovora (provisionalnih i finalnih), upravlja retransmisijama i timeoutovima na transakcijskom sloju. Dijalog je viši nivo – peer-to-peer asocijacija između dvije tačke koja traje dulje, obuhvaća više transakcija (npr. INVITE + BYE) s istim Call-ID, From i To headerima. [9][10][11]

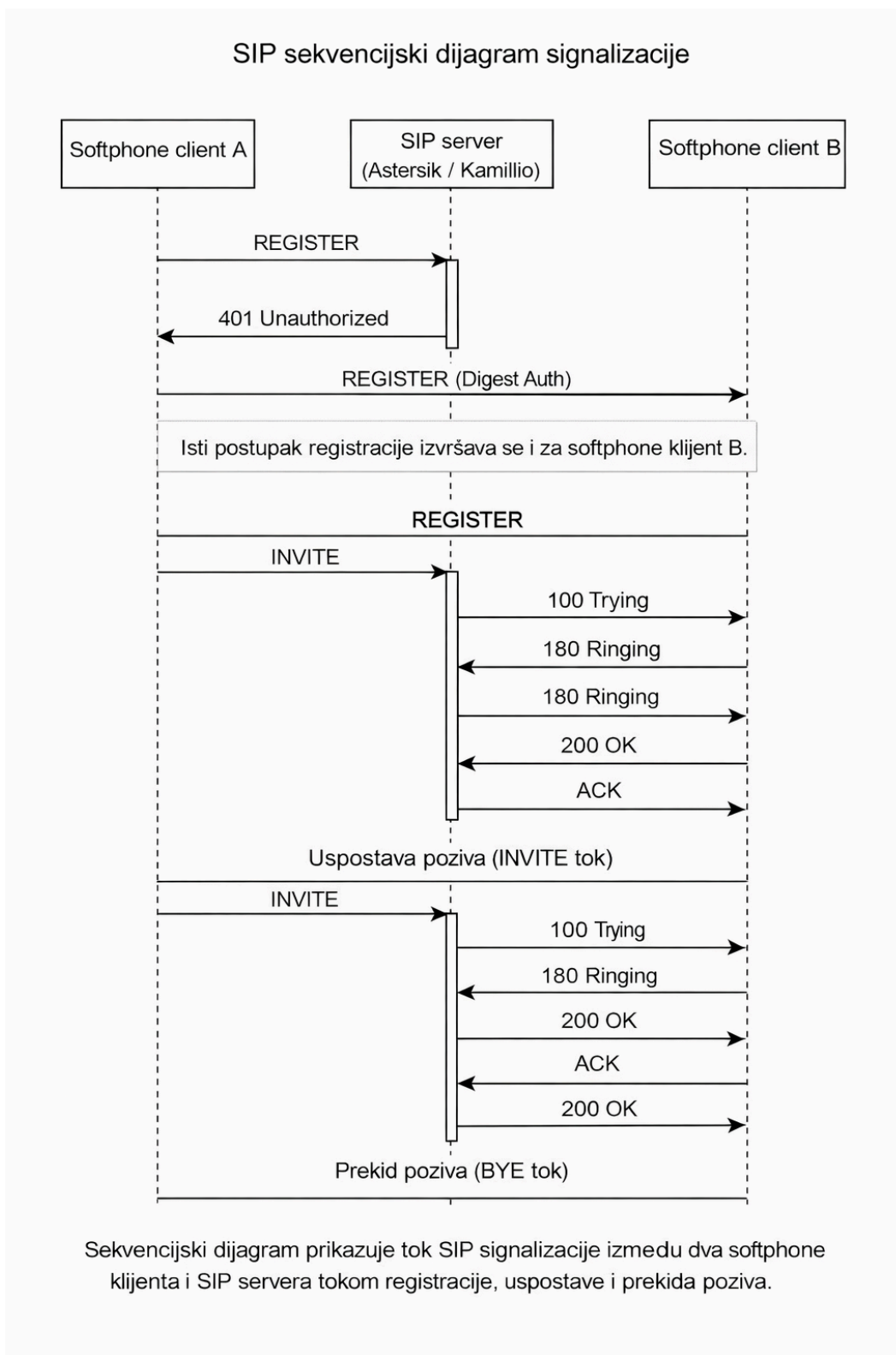
2.3.2. Kako nastaje dijalog

Dijalog se uspostavlja uspješnom INVITE transakcijom: nakon 200 OK odgovora i ACK-a, stvara se stanje dijaloga s jedinstvenim identifikatorom (Call-ID + From tag + To tag). Unutar njega mogu slijediti ACK, re-INVITE (za modifikaciju) ili BYE transakcije, održavajući kontinuitet sesije. [9][11]

2.3.3. Važnost za trajanje poziva

Dijalog održava stanje poziva (stateful), omogućavajući praćenje trajanja, modifikacije (npr. dodavanje medija) i graciozni prekid bez gubitka konteksta. Bez njega, svaka interakcija bila bi izolirana, što otežava upravljanje dužim sesijama poput konferencija ili transfera poziva. [9][10]

2.4. SIP call-flow



Slika 1. SIP call-flow sekvencijski dijagram

- REGISTER tok:

U SIP sistemima, prije uspostave poziva, korisnički agent se mora registrovati na SIP server kako bi bio dostupan drugim korisnicima. Registracija se ostvaruje slanjem REGISTER poruke prema registrar serveru, pri čemu klijent obavještava server o svojoj trenutnoj lokaciji (IP adresa i port).

Tok registracije započinje kada softphone klijent pošalje REGISTER zahtjev SIP serveru. Server obično odgovara sa 401 Unauthorized porukom, kojom zahtijeva autentikaciju. Klijent zatim ponovo šalje REGISTER poruku sa odgovarajućim autentikacijskim podacima (Digest autentikacija). Nakon uspješne provjere, server vraća odgovor 200 OK, čime je registracija uspješno završena.

Ovim postupkom SIP server ažurira svoju lokacijsku bazu podataka i zna gdje može proslijediti dolazne pozive za registrovanog korisnika.

- Uspostava poziva - INVITE tok:

Nakon uspješne registracije oba klijenta, moguće je uspostaviti SIP poziv. Proces započinje slanjem INVITE poruke od pozivajućeg klijenta prema SIP serveru, koja sadrži i SDP opis medijskih mogućnosti.

Po prijemu INVITE poruke, SIP server prosljeđuje zahtjev prema pozvanom klijentu. Tokom obrade poziva, klijent ili server šalju provisionalne odgovore kako bi se označio napredak poziva. Odgovor 100 Trying označava da je zahtjev zaprimljen i da se obrađuje, dok odgovor 180 Ringing signalizira da pozvani klijent zvoní.

Kada pozvani korisnik prihvati poziv, njegov klijent šalje odgovor 200 OK koji sadrži SDP odgovor. Pozivajući klijent zatim potvrđuje uspostavu sesije slanjem ACK poruke. Nakon razmjene ACK poruke, SIP signalizacija za uspostavu poziva je završena, a medijska komunikacija može započeti.

- Prekid poziva - BYE tok:

Prekid poziva u SIP sistemima realizuje se pomoću BYE poruke. Kada jedna strana odluči prekinuti poziv, njen klijent šalje BYE zahtjev drugoj strani putem SIP servera. Primalac BYE poruke odgovara sa 200 OK, čime se potvrđuje prekid sesije i oslobađaju se resursi povezani sa pozivom.

Ovim se SIP dijalog završava, a oba klijenta se vraćaju u stanje mirovanja, dok ostaju registrovani na SIP serveru.

2.5. SIP u IMS arhitekturi

IMS (IP Multimedia Subsystem) predstavlja standardizovanu arhitekturu za isporuku multimedijalnih usluga preko IP mreža, razvijenu prvenstveno za potrebe telekom operatera. [12][13] Za razliku od klasičnih VoIP rješenja, IMS integriše signalizaciju, autentikaciju, upravljanje sesijama i kontrolu kvaliteta usluge u jedinstven sistem. SIP protokol ima centralnu ulogu u IMS arhitekturi i koristi se kao osnovni signalizacioni mehanizam za upravljanje multimedijalnim sesijama. [12][14]

2.5.1. Uloge P-CSCF, I-CSCF i S-CSCF

U IMS arhitekturi SIP signalizacija se obrađuje kroz više funkcionalnih čvorova, od kojih su najvažniji P-CSCF, I-CSCF i S-CSCF. [13]

P-CSCF (Proxy Call Session Control Function) predstavlja prvu tačku kontakta korisničkog uređaja sa IMS mrežom. Svi SIP zahtjevi koje korisnik šalje prolaze kroz P-CSCF, koji ima ulogu SIP proxy servera. Ovaj čvor je zadužen za prosleđivanje SIP poruka, primjenu sigurnosnih mehanizama (npr. SIP over IPsec ili TLS) i interakciju sa sistemima za kontrolu politike i kvaliteta usluge. [13][15]

I-CSCF (Interrogating Call Session Control Function) služi kao ulazna tačka u IMS mrežu određenog operatora. Njegova osnovna uloga je da na osnovu informacija iz HSS-a (Home Subscriber Server) odredi koji S-CSCF je zadužen za konkretnog korisnika. I-CSCF ne održava stanje sesije, već djeluje kao inteligentni SIP router unutar IMS domene. [13]

S-CSCF (Serving Call Session Control Function) je centralni čvor IMS signalizacije. On upravlja SIP dijalozima, održava stanje sesija i primjenjuje servisnu logiku. S-CSCF je zadužen za registraciju korisnika, autentikaciju, autorizaciju i interakciju sa aplikacijskim serverima. [12][13] U praksi, S-CSCF ima sličnu ulogu kao SIP server u VoIP sistemima, ali sa znatno proširenim funkcijama i integracijom u operatorovu infrastrukturu.

2.5.2. Razlika između klasičnih VoIP sistema i IMS arhitekture

Klasični VoIP sistemi, poput onih zasnovanih na Asterisk ili Kamailio serverima, koriste SIP protokol za signalizaciju, ali su uglavnom fokusirani na osnovnu funkcionalnost uspostave i prekida poziva. [16] Takvi sistemi su fleksibilni, jednostavni za implementaciju i često se koriste u manjim ili privatnim mrežama.

IMS arhitektura, s druge strane, predstavlja operator-grade rješenje koje integriše SIP signalizaciju sa sistemima za autentikaciju pretplatnika, naplatu, kontrolu kvaliteta usluge i sigurnost. [12][13] IMS omogućava centralizovano upravljanje korisnicima i servisima, kao i konzistentno ponašanje sistema u velikim mrežama sa velikim brojem korisnika. Dok je SIP zajednički element i VoIP i IMS sistema, IMS uvodi dodatne mrežne komponente i strožije kontrolne mehanizme.

2.5.3. Sigurnost i autentikacija u IMS sistemima

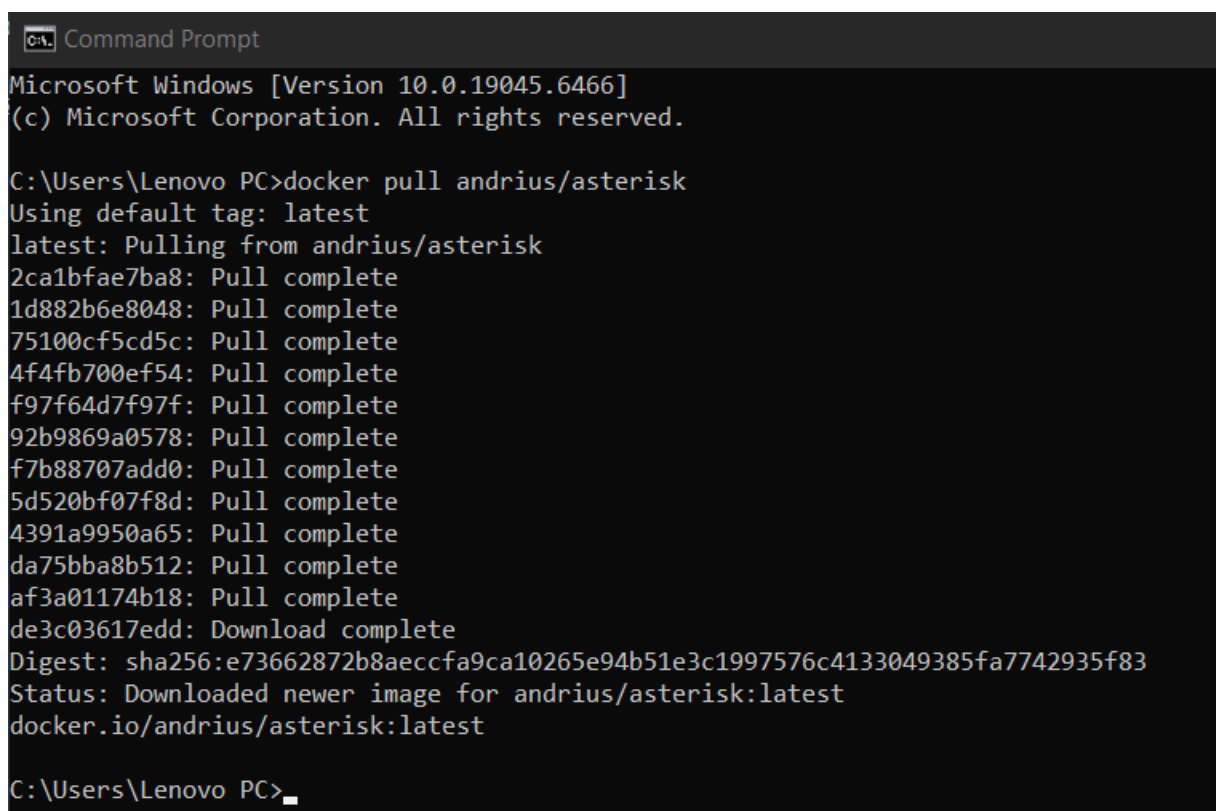
Sigurnost u IMS arhitekturi ima znatno veću ulogu nego u klasičnim VoIP implementacijama. Autentikacija korisnika u IMS-u zasniva se na integraciji sa HSS bazom podataka, gdje su pohranjeni podaci o pretplatnicima. [13] Okom SIP registracije, IMS koristi mehanizme kao što su SIP Digest autentikacija, često u kombinaciji sa kriptografskim protokolima poput IPSec-a ili TLS-a, kako bi se osigurala povjerljivost i integritet signalizacije. [14][15]

Pored autentikacije, IMS sistem omogućava i kontrolu autorizacije korisnika, odnosno provjeru da li korisnik ima pravo pristupa određenim servisima. Ovakav pristup osigurava viši nivo sigurnosti, pouzdanosti i skalabilnosti u odnosu na osnovne VoIP sisteme, što IMS čini pogodnim za primjenu u velikim komercijalnim mrežama. [12][13]

3. Setup SIP servisa

Za realizaciju SIP serverske infrastrukture u okviru ovog projekta odabran je Asterisk kao open-source SIP server, sa podrškom za PJSIP modul. Asterisk je izabran zbog široke upotrebe u VoIP sistemima, dobre dokumentacije i mogućnosti jednostavne integracije sa vlastitim softphone klijentima. Asterisk server je postavljen korištenjem Docker tehnologije, čime je omogućeno izolovano i jednostavno pokretanje serverskog okruženja bez potrebe za direktnom instalacijom i konfiguracijom zavisnosti na lokalnom operativnom sistemu. Korištenjem Docker kontejnera osigurana je prenosivost i ponovljivost okruženja, što je pogodno za testiranje i demonstraciju sistema.

Docker image za Asterisk je preuzet sa Docker Hub repozitorija i uspješno je pripremljen za pokretanje SIP servisa sa `docker pull andrius/asterisk`. Time je potvrđeno da je Asterisk serverska komponenta spremna za dalju konfiguraciju SIP korisnika i uspostavu komunikacije između softphone klijenata.



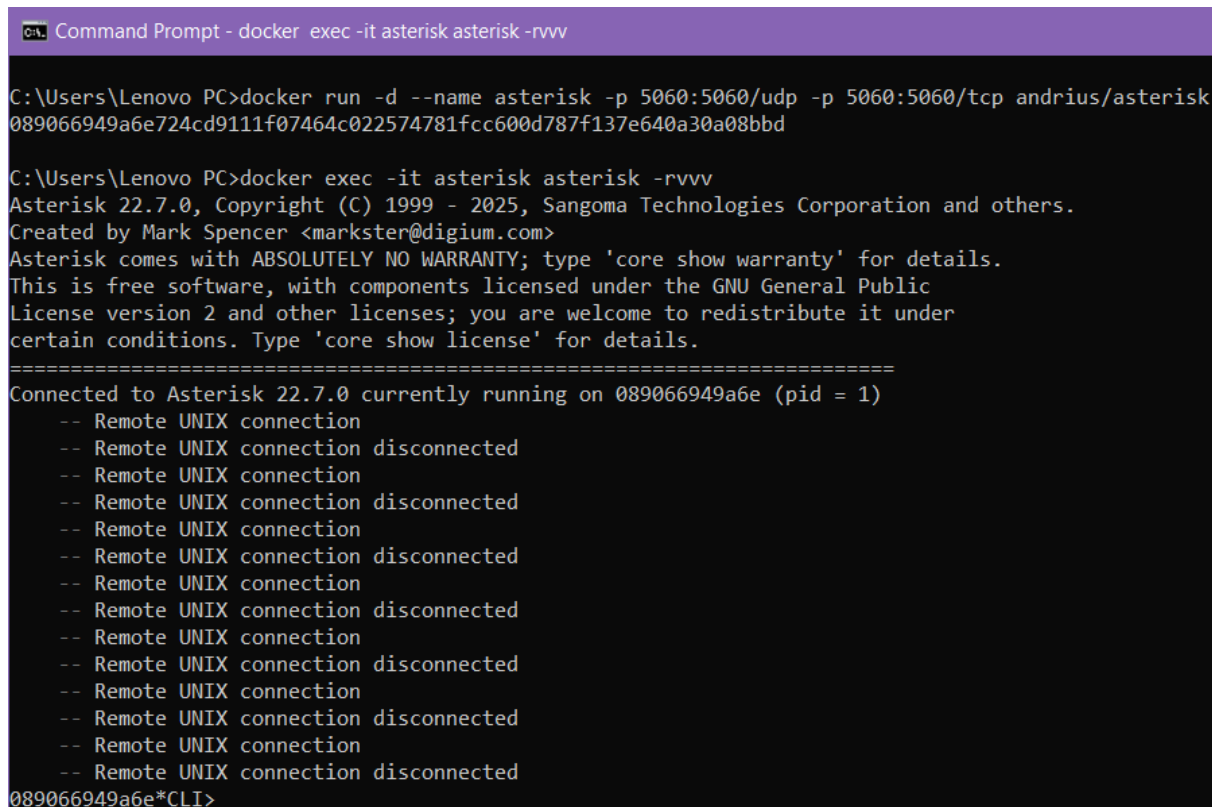
```
Command Prompt
Microsoft Windows [Version 10.0.19045.6466]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Lenovo PC>docker pull andrius/asterisk
Using default tag: latest
latest: Pulling from andrius/asterisk
2ca1bfae7ba8: Pull complete
1d882b6e8048: Pull complete
75100cf5cd5c: Pull complete
4f4fb700ef54: Pull complete
f97f64d7f97f: Pull complete
92b9869a0578: Pull complete
f7b88707add0: Pull complete
5d520bf07f8d: Pull complete
4391a9950a65: Pull complete
da75bba8b512: Pull complete
af3a01174b18: Pull complete
de3c03617edd: Download complete
Digest: sha256:e73662872b8aeccfa9ca10265e94b51e3c1997576c4133049385fa7742935f83
Status: Downloaded newer image for andrius/asterisk:latest
docker.io/andrius/asterisk:latest

C:\Users\Lenovo PC>
```

Slika 2. docker pull andrius/asterisk - image exists

Asterisk servis je uspješno pokrenut unutar Docker kontejnera, a ispravan rad servera potvrđen je pristupom Asterisk CLI interfejsu sa `docker run -d --name asterisk -p 5060:5060/udp -p 5060:5060/tcp andrius/asterisk` i `docker exec -it asterisk asterisk -rvvv`.



```
Command Prompt - docker exec -it asterisk asterisk -rvvv

C:\Users\Lenovo PC>docker run -d --name asterisk -p 5060:5060/udp -p 5060:5060/tcp andrius/asterisk
089066949a6e724cd9111f07464c022574781fcc600d787f137e640a30a08bbd

C:\Users\Lenovo PC>docker exec -it asterisk asterisk -rvvv
Asterisk 22.7.0, Copyright (C) 1999 - 2025, Sangoma Technologies Corporation and others.
Created by Mark Spencer <markster@digium.com>
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty' for details.
This is free software, with components licensed under the GNU General Public
License version 2 and other licenses; you are welcome to redistribute it under
certain conditions. Type 'core show license' for details.
=====
Connected to Asterisk 22.7.0 currently running on 089066949a6e (pid = 1)
-- Remote UNIX connection
-- Remote UNIX connection disconnected
-- Remote UNIX connection
-- Remote UNIX connection disconnected
-- Remote UNIX connection
-- Remote UNIX connection disconnected
-- Remote UNIX connection
-- Remote UNIX connection disconnected
-- Remote UNIX connection
-- Remote UNIX connection disconnected
-- Remote UNIX connection
-- Remote UNIX connection disconnected
089066949a6e*CLI>
```

Slika 2. `docker run -d --name asterisk -p 5060:5060/udp -p 5060:5060/tcp andrius/asterisk` i `docker exec -it asterisk asterisk -rvvv`

3.1. Konfiguracija SIP korisnika

U okviru ovog zadatka izvršena je konfiguracija SIP korisnika na Asterisk serveru korištenjem PJSIP modula. Definisane su dvije SIP ekstenzije, 1001 i 1002, koje predstavljaju dva softphone klijenta u sistemu. Za autentikaciju korisnika korišten je SIP Digest autentikacioni mehanizam, pri čemu svaki korisnik ima vlastito korisničko ime i lozinku. Konfiguracija je realizovana u fajlu `pjsip.conf`, gdje su za svaki korisnik definisani endpoint, autentikacijski parametri i AOR (Address of Record). Nakon ponovnog učitavanja PJSIP konfiguracije, uspješno su detektovani konfigurisani SIP endpointi na Asterisk serveru.

```
C:\> asterisk@089066949a6e: ~  
Microsoft Windows [Version 10.0.19045.6466]  
(c) Microsoft Corporation. All rights reserved.  
C:\Users\Lenovo PC>docker exec -it asterisk /bin/bash  
asterisk@089066949a6e:~$
```

Slika 3. docker exec -it asterisk /bin/bash - pristup Asterisk Docker kontejneru

Na slici 3. je prikazan pristup Asterisk serveru koji je pokrenut unutar Docker kontejnera. Korištenjem Command Prompt okruženja izvršena je komanda `docker exec -it asterisk /bin/bash`, kojom se ostvaruje interaktivni pristup shell okruženju unutar aktivnog Docker kontejnera nazvanog asterisk.

Ovom komandom omogućeno je direktno upravljanje fajlovima i servisima unutar kontejnera, što je neophodno za dalju konfiguraciju Asterisk SIP servera. Nakon izvršenja komande, vidljiv je prompt `asterisk@<container_id>`, što potvrđuje da je korisnik uspješno ušao u kontejner i da je Asterisk serversko okruženje spremno za konfiguraciju SIP parametara.

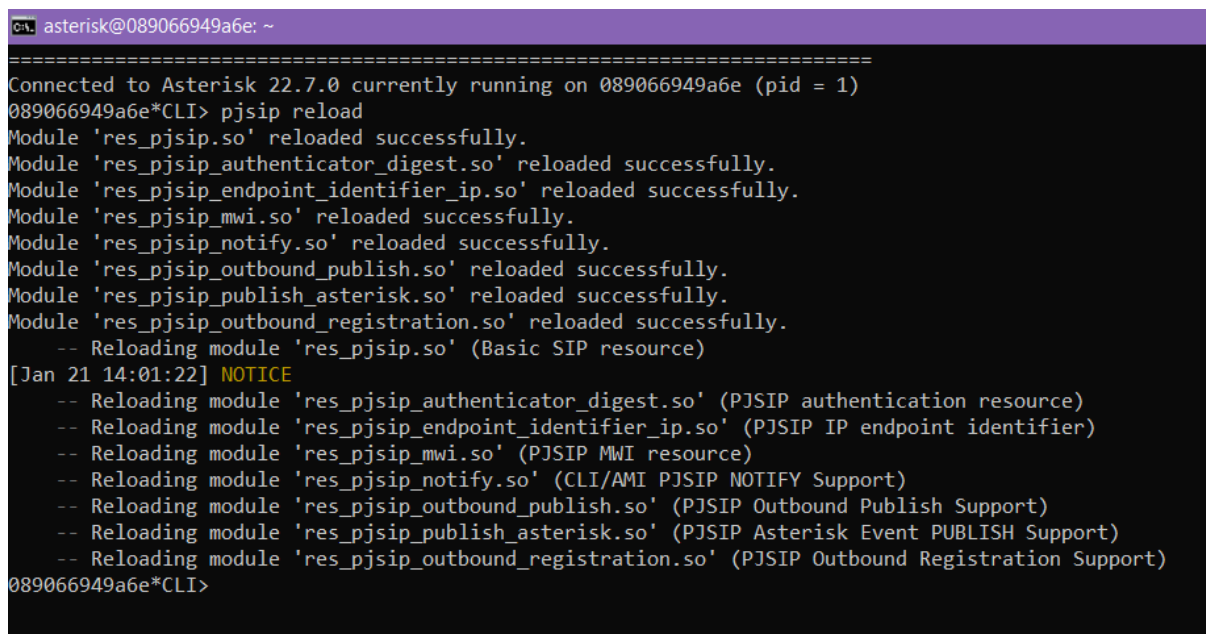
```
asterisk@089066949a6e: ~  
asterisk@089066949a6e:~$ cat <<EOF > /etc/asterisk/pjsip.conf  
> [transport-udp]  
> type=transport  
> protocol=udp  
> bind=0.0.0.0  
>  
> ; ----- USER 1001 -----  
> [1001]  
> type=endpoint  
> context=internal  
> disallow=all  
> allow=ulaw  
> auth=auth1001  
> aors=1001  
>  
> [auth1001]  
> type=auth  
> auth_type=userpass  
> username=1001  
> password=1001pass  
>  
> [1001]  
> type=aor  
> max_contacts=1  
>  
> ; ----- USER 1002 -----  
> [1002]  
> type=endpoint  
> context=internal  
> disallow=all  
> allow=ulaw  
> auth=auth1002  
> aors=1002  
>  
> [auth1002]  
> type=auth  
> auth_type=userpass  
> username=1002  
> password=1002pass  
>  
> [1002]  
> type=aor  
> max_contacts=1  
> EOF  
asterisk@089066949a6e:~$ cat /etc/asterisk/pjsip.conf  
[transport-udp]  
type=transport  
protocol=udp  
bind=0.0.0.0
```

Slika 4. Konfiguracija SIP korisnika u fajlu pjsip.conf

Na slici 4 je prikazan proces kreiranja i provjere konfiguracionog fajla pjsip.conf unutar Asterisk Docker kontejnera. Konfiguracija je unesena korištenjem shell komande `cat <<EOF > /etc/asterisk/pjsip.conf`, kojom je definisan sadržaj fajla bez korištenja dodatnih tekstualnih editora.

U fajlu su definisane dvije SIP ekstenzije, 1001 i 1002, koje predstavljaju softphone klijente u sistemu. Za svaki korisnik definisani su odgovarajući PJSIP elementi: endpoint, auth i AOR (Address of Record). Autentikacija korisnika realizovana je korištenjem SIP Digest autentikacije, gdje svaki korisnik ima jedinstveno korisničko ime i lozinku.

Na početku fajla definisan je transportni sloj (`transport-udp`), koji omogućava prijem SIP signalizacije putem UDP protokola. Nakon unosa konfiguracije, sadržaj fajla je ponovo prikazan korištenjem komande `cat /etc/asterisk/pjsip.conf`, čime je potvrđeno da je konfiguracija uspješno sačuvana i spremna za učitavanje u Asterisk server.



```
asterisk@089066949a6e: ~
=====
Connected to Asterisk 22.7.0 currently running on 089066949a6e (pid = 1)
089066949a6e*CLI> pjsip reload
Module 'res_pjsip.so' reloaded successfully.
Module 'res_pjsip_authenticator_digest.so' reloaded successfully.
Module 'res_pjsip_endpoint_identifier_ip.so' reloaded successfully.
Module 'res_pjsip_mwi.so' reloaded successfully.
Module 'res_pjsip_notify.so' reloaded successfully.
Module 'res_pjsip_outbound_publish.so' reloaded successfully.
Module 'res_pjsip_publish_asterisk.so' reloaded successfully.
Module 'res_pjsip_outbound_registration.so' reloaded successfully.
-- Reloading module 'res_pjsip.so' (Basic SIP resource)
[Jan 21 14:01:22] NOTICE
-- Reloading module 'res_pjsip_authenticator_digest.so' (PJSIP authentication resource)
-- Reloading module 'res_pjsip_endpoint_identifier_ip.so' (PJSIP IP endpoint identifier)
-- Reloading module 'res_pjsip_mwi.so' (PJSIP MWI resource)
-- Reloading module 'res_pjsip_notify.so' (CLI/AMI PJSIP NOTIFY Support)
-- Reloading module 'res_pjsip_outbound_publish.so' (PJSIP Outbound Publish Support)
-- Reloading module 'res_pjsip_publish_asterisk.so' (PJSIP Asterisk Event PUBLISH Support)
-- Reloading module 'res_pjsip_outbound_registration.so' (PJSIP Outbound Registration Support)
089066949a6e*CLI>
```

Slika 5. Učitavanje PJSIP konfiguracije na Asterisk serveru

Na slici je prikazan proces ponovnog učitavanja PJSIP konfiguracije na Asterisk SIP serveru korištenjem Asterisk CLI interfejsa. Nakon uspostavljanja veze sa serverom (`Connected to Asterisk ...`), izvršena je komanda `pjsip reload`, kojom se učitavaju izmjene napravljene u fajlu pjsip.conf bez potrebe za restartovanjem kompletnog Asterisk servisa.

Iz prikazanog ispisa vidljivo je da su PJSIP moduli uspješno ponovo učitani, uključujući module za autentikaciju (`res_pjsip_authenticator_digest`), identifikaciju endpointa i registraciju. Ovakav ispis potvrđuje da konfiguracija SIP korisnika nije sadržavala sintaksne greške i da je Asterisk server uspješno prihvatio definisane PJSIP parametre.

Ovim korakom potvrđeno je da su SIP ekstenzije 1001 i 1002 pravilno učitane u sistem i da je server spreman za dalju konfiguraciju dialplana i testiranje SIP poziva.

```
089066949a6e*CLI> pjsip show endpoints

Endpoint: <Endpoint/CID.....> <State.....> <Channels.>
  I/OAuth: <AuthId/UserName.....>
  Aor: <Aor.....> <MaxContact>
  Contact: <Aor/ContactUri.....> <Hash.....> <Status> <RTT(ms)..>
  Transport: <TransportId.....> <Type> <cos> <tos> <BindAddress.....>
  Identify: <Identify/Endpoint.....>
  Match: <criteria.....>
  Channel: <ChannelId.....> <State.....> <Time.....>
  Exten: <DialedExten.....> CLCID: <ConnectedLineCID.....>
=====

Endpoint: 1001                               Unavailable  0 of inf
  InAuth: auth1001/1001
  Aor: 1001                                   1

Endpoint: 1002                               Unavailable  0 of inf
  InAuth: auth1002/1002
  Aor: 1002                                   1

Objects found: 2

089066949a6e*CLI> _
```

Slika 6. Prikaz registrovanih PJSIP endpointa na Asterisk serveru

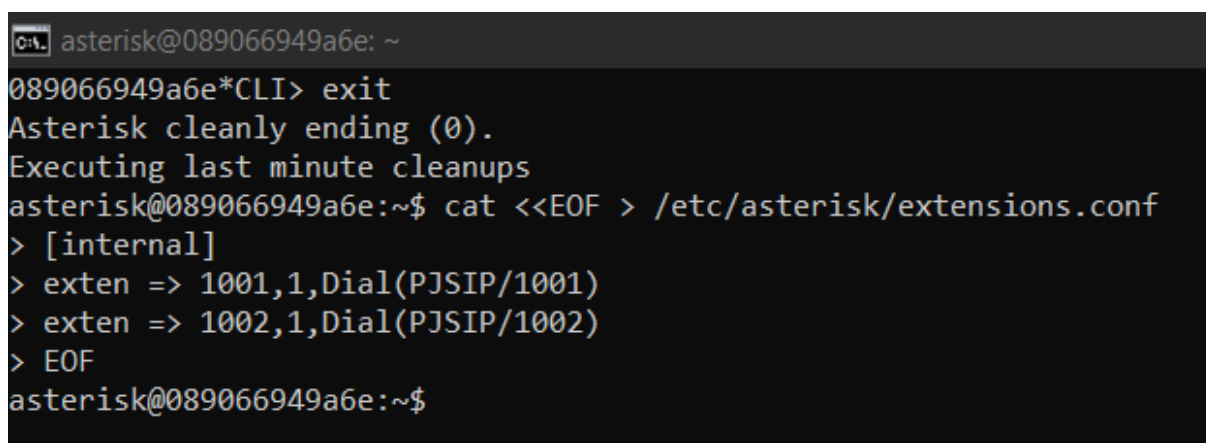
Na slici je prikazan rezultat izvršavanja Asterisk CLI komande `pjsip show endpoints`, koja se koristi za provjeru definisanih SIP endpointa u PJSIP konfiguraciji. Iz prikazanog ispisa vidljivo je da su na Asterisk serveru uspješno kreirane dvije SIP ekstenzije, 1001 i 1002, koje predstavljaju softphone klijente u sistemu.

Za svaki endpoint prikazani su pripadajući autentikacioni podaci (InAuth), kao i odgovarajući AOR (Address of Record), što potvrđuje da su korisnici pravilno definisani u fajlu `pjsip.conf`. Status Unavailable je očekivan u ovom trenutku, budući da se softphone klijenti još nisu registrovali na server.

Ovim korakom potvrđeno je da je konfiguracija SIP korisnika uspješno učitana i da je Asterisk server spreman za konfiguraciju dialplana i uspostavu SIP poziva između definisanih ekstenzija.

3.2. Dialplan konfiguracija

U ovom podpoglavlju prikazana je konfiguracija dialplana na Asterisk SIP serveru, čiji je cilj omogućiti rutiranje poziva između prethodno definisanih SIP korisnika. Dialplan predstavlja logiku obrade poziva u Asterisk sistemu i definiše kako se dolazni i odlazni pozivi mapiraju na odgovarajuće SIP ekstenzije. U okviru ovog zadatka konfigurisan je jednostavan dialplan koji omogućava međusobno pozivanje SIP ekstenzija 1001 i 1002, čime se obezbeđuje osnovna funkcionalnost VoIP komunikacije.



```
asterisk@089066949a6e: ~
089066949a6e*CLI> exit
Asterisk cleanly ending (0).
Executing last minute cleanups
asterisk@089066949a6e:~$ cat <<EOF > /etc/asterisk/extensions.conf
> [internal]
> exten => 1001,1,Dial(PJSIP/1001)
> exten => 1002,1,Dial(PJSIP/1002)
> EOF
asterisk@089066949a6e:~$
```

Slika 7. Kreiranje dialplan konfiguracije u Asterisk serveru

Na slici 7. prikazan je proces kreiranja dialplan konfiguracije u Asterisk SIP serveru koji je pokrenut unutar Docker kontejnera. Nakon izlaska iz Asterisk CLI okruženja, korištena je Linux shell komanda `cat <<EOF` za upis sadržaja u fajl `extensions.conf`. U okviru konteksta `[internal]` definisane su SIP ekstenzije 1001 i 1002, pri čemu je za svaku ekstenziju korištena `Dial(PJSIP/...)` instrukcija. Ovim korakom omogućeno je rutiranje poziva između definisanih SIP korisnika, čime je realizovana osnovna dialplan logika potrebna za SIP komunikaciju.


```

C:\ asterisk@089066949a6e: ~
asterisk@089066949a6e:~$ cat <<EOF > /etc/asterisk/extensions.conf
> [internal]
> exten => 1001,1,Dial(PJSIP/1001)
> exten => 1002,1,Dial(PJSIP/1002)
> EOF
asterisk@089066949a6e:~$ cat /etc/asterisk/extensions.conf
[internal]
exten => 1001,1,Dial(PJSIP/1001)
exten => 1002,1,Dial(PJSIP/1002)
asterisk@089066949a6e:~$ █

```

Slika 8. Provjera sadržaja dialplan konfiguracije

Na slici 8. prikazana je provjera sadržaja fajla `extensions.conf` nakon njegovog kreiranja. Korištenjem Linux shell komande `cat /etc/asterisk/extensions.conf` potvrđeno je da je dialplan konfiguracija ispravno sačuvana. U okviru konteksta `[internal]` vidljive su definicije SIP ekstenzija 1001 i 1002 sa odgovarajućim `Dial(PJSIP/...)` instrukcijama, čime je potvrđeno da je konfiguracija rutiranja poziva između SIP korisnika pravilno definisana i spremna za učitavanje u Asterisk sistem.

```

C:\ asterisk@089066949a6e: ~
asterisk@089066949a6e:~$ asterisk -rvvv
Asterisk 22.7.0, Copyright (C) 1999 - 2025, Sangoma Technologies Corporation and others.
Created by Mark Spencer <markster@digium.com>
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty' for details.
This is free software, with components licensed under the GNU General Public
License version 2 and other licenses; you are welcome to redistribute it under
certain conditions. Type 'core show license' for details.
=====
Connected to Asterisk 22.7.0 currently running on 089066949a6e (pid = 1)
089066949a6e*CLI> █

```

Slika 9. Pokretanje Asterisk CLI okruženja

Na slici 9. prikazan je ulazak u Asterisk Command Line Interface (CLI) korištenjem komande `asterisk -rvvv`. Ovom komandom uspostavljena je interaktivna veza sa aktivnim Asterisk serverom koji se izvršava unutar Docker kontejnera. Iz prikaza je vidljivo da je Asterisk verzije 22.7.0 uspješno pokrenut i spreman za dalju administraciju, uključujući učitavanje konfiguracija i upravljanje SIP signalizacijom putem komandne linije.

```
asterisk@089066949a6e: ~  
089066949a6e*CLI> dialplan reload  
Dialplan reloaded.  
-- Including switch 'DUNDi/e164' in context 'ael-dundi-e164-switch'  
-- Remote UNIX connection  
-- Remote UNIX connection disconnected  
-- Remote UNIX connection  
-- Remote UNIX connection disconnected  
-- Remote UNIX connection  
-- Remote UNIX connection disconnected  
-- Remote UNIX connection  
-- Remote UNIX connection disconnected  
-- Remote UNIX connection  
-- Remote UNIX connection disconnected  
-- Remote UNIX connection  
-- Remote UNIX connection disconnected  
089066949a6e*CLI>
```

Slika 10. Učitavanje dialplan konfiguracije

Na slici 10. prikazan je proces ponovnog učitavanja dialplan konfiguracije u Asterisk serveru korištenjem komande `dialplan reload`. Ovim korakom Asterisk server učitava izmjene iz fajla `extensions.conf` bez potrebe za restartovanjem servisa. Poruka o uspješnom učitavanju potvrđuje da je dialplan konfiguracija ispravno parsirana i aktivirana, čime su definisane SIP ekstenzije spremne za dalju obradu i rutiranje poziva unutar sistema.

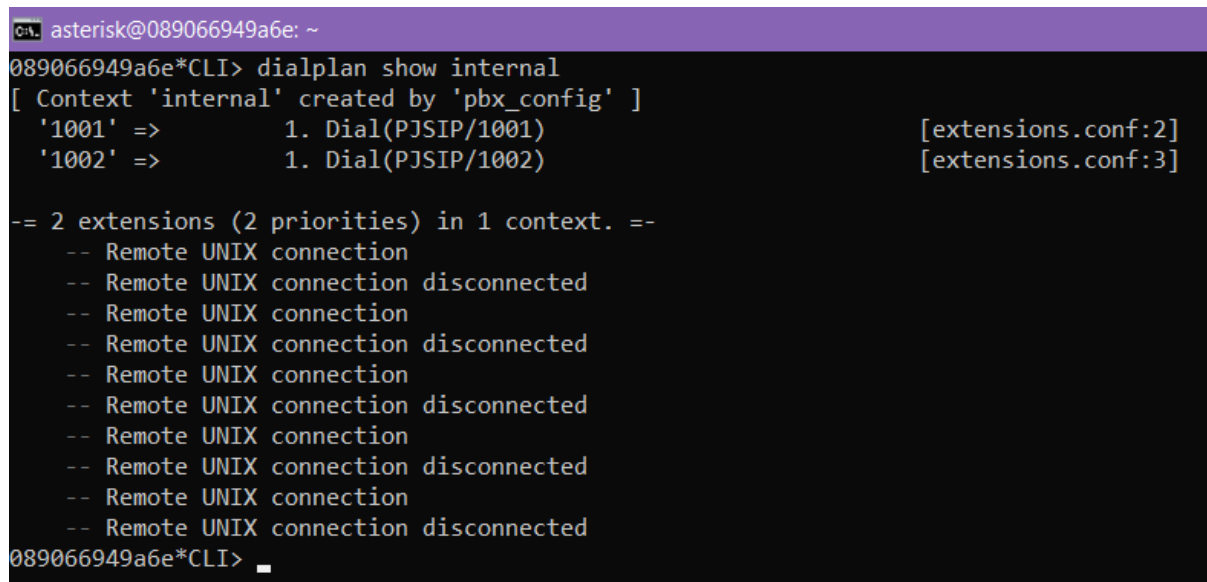
```
089066949a6e*CLI> channel originate PJSIP/1001 extension 1002@internal  
[Jan 21 15:04:26] ERROR  
ERROR
```

Slika 11. Pokušaj iniciranja testnog SIP poziva

Na slici 11 prikazan je pokušaj iniciranja testnog SIP poziva korištenjem komande `channel originate`, kojom se simulira uspostava poziva sa SIP ekstenzije 1001 prema ekstenziji 1002 unutar konteksta `internal`. Prilikom izvršavanja komande, Asterisk server vraća poruku o grešci, što ukazuje da u trenutku testiranja ne postoji aktivno registrovan SIP klijent koji bi mogao prihvatiti poziv.

Do pojave greške dolazi ne zbog pogrešne konfiguracije dialplana, već zbog činjenice da SIP korisnici (1001 i 1002) još uvijek nisu registrovani putem softphone klijenata. Komanda `channel originate` zahtijeva postojanje aktivnog PJSIP kanala, koji se uspostavlja tek nakon

uspješne registracije stvarnog klijentskog uređaja. U ovoj fazi projekta fokus je na konfiguraciji SIP servera i dialplana, dok će stvarna uspostava poziva biti realizovana u narednim fazama projekta kroz implementaciju softphone klijenata.



```
asterisk@089066949a6e: ~
089066949a6e*CLI> dialplan show internal
[ Context 'internal' created by 'pbx_config' ]
'1001' =>          1. Dial(PJSIP/1001)                [extensions.conf:2]
'1002' =>          1. Dial(PJSIP/1002)                [extensions.conf:3]

-- 2 extensions (2 priorities) in 1 context. --
-- Remote UNIX connection
-- Remote UNIX connection disconnected
-- Remote UNIX connection
-- Remote UNIX connection disconnected
-- Remote UNIX connection
-- Remote UNIX connection disconnected
-- Remote UNIX connection
-- Remote UNIX connection disconnected
-- Remote UNIX connection
-- Remote UNIX connection disconnected
089066949a6e*CLI> _
```

Slika 11. Pregled aktivnog dialplana u kontekstu internal

Na slici 11. prikazan je pregled aktivnog dialplana korištenjem Asterisk CLI komande `dialplan show internal`. Ovim prikazom potvrđeno je da je kontekst `internal` uspješno kreiran i učitao iz fajla `extensions.conf`. Vidljivo je da su u okviru navedenog konteksta definisane dvije SIP ekstenzije, 1001 i 1002, koje koriste `Dial(PJSIP/...)` instrukcije za usmjeravanje poziva prema odgovarajućim SIP endpointima. Prikaz potvrđuje da je dialplan konfiguracija ispravno parsirana i aktivna, te da je Asterisk server spreman za rutiranje poziva između definisanih SIP korisnika.

4. Softphone klijent A (User Agent 1)

U ovom poglavlju opisana je implementacija i testiranje softphone klijenta A, koji u okviru sistema ima ulogu SIP korisnika inicijatora poziva. Cilj ovog dijela projekta bio je demonstrirati osnovnu funkcionalnost SIP klijenta, uključujući inicijalizaciju SIP stacka, registraciju na SIP server i slanje odlaznog poziva prema drugom korisniku u sistemu.

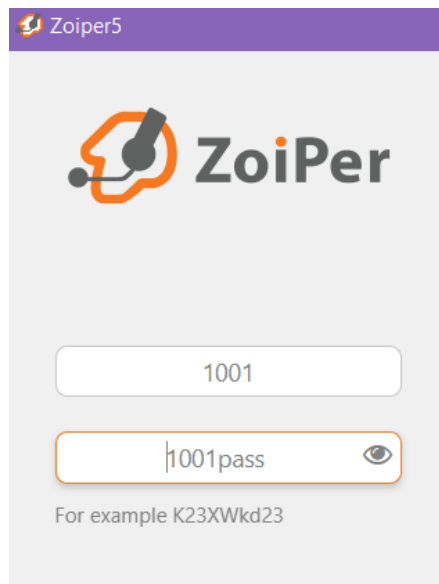
Softphone klijent je konfigurisan korištenjem SIP protokola i UDP transporta, a za praktičnu realizaciju korištena je Zoiper aplikacija povezana na Asterisk SIP server. Tokom implementacije, klijent se uspješno registrovao na server koristeći SIP REGISTER mehanizam sa Digest autentikacijom, čime je potvrđena ispravna konfiguracija korisničkog naloga i komunikacije sa SIP serverom.

Nakon uspješne registracije, testirana je funkcionalnost iniciranja odlaznog poziva prema drugoj SIP ekstenziji. Analiza SIP signalizacije izvršena je putem Asterisk CLI logova, gdje su jasno vidljive SIP poruke poput INVITE zahtjeva, odgovora servera i pokušaja rutiranja poziva kroz dialplan. Iako poziv nije u potpunosti uspostavljen zbog nedostupnosti ciljanog korisnika, potvrđeno je ispravno generisanje i obrada SIP signalizacije na strani softphone klijenta.

Ovim koracima potvrđena je funkcionalnost softphone klijenta A kao SIP User Agent-a, što predstavlja osnovu za daljnju implementaciju prijema i obrade dolaznih poziva u narednom poglavlju.

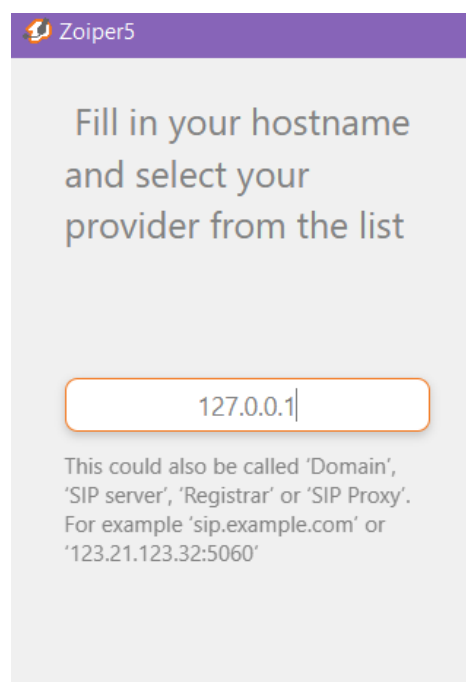
4.1. Inicijalizacija SIP stack-a

Ovdje je prikazan proces inicijalizacije SIP stacka za softphone klijent A. Korištenjem Zoiper softphone aplikacije izvršena je konfiguracija SIP naloga koji komunicira sa lokalnim Asterisk SIP serverom. Tokom konfiguracije definisani su osnovni parametri kao što su korisnički podaci, adresa SIP servera i transportni protokol. Uspješnom detekcijom SIP UDP transporta i aktivacijom softphone klijenta potvrđeno je da je SIP stack ispravno inicijalizovan i spreman za dalju signalizacionu komunikaciju.



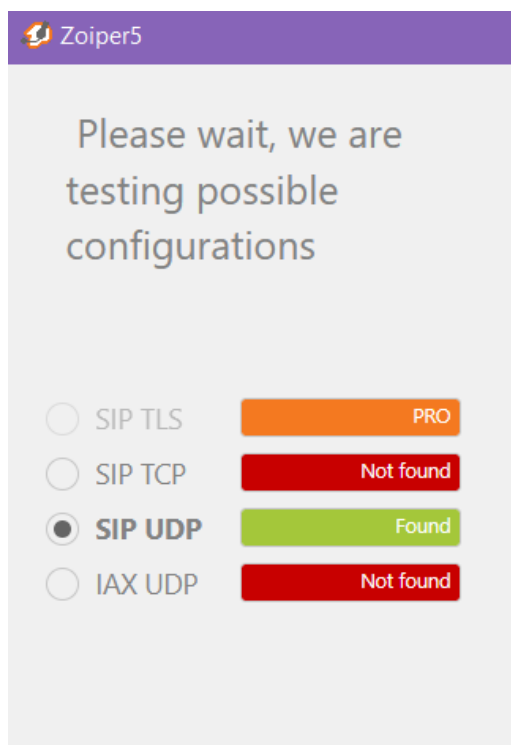
Slika 12. Unos SIP korisničkih podataka u Zoiper softphone aplikaciji

Na slici 12. prikazan je ekran Zoiper softphone aplikacije na kojem se vrši unos SIP korisničkih podataka za softphone klijent A. U polja za prijavu unijeti su korisničko ime 1001 i odgovarajuća lozinka, čime se definišu kredencijali SIP naloga koji je prethodno konfigurisan na Asterisk serveru. Ovim korakom započinje proces povezivanja softphone klijenta sa lokalnim SIP serverom, što predstavlja osnovu za inicijalizaciju SIP stacka i kasniju registraciju korisnika.



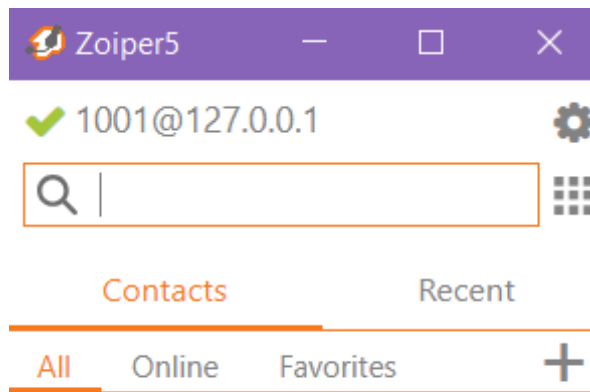
Slika 13. Unos adrese SIP servera u Zoiper softphone aplikaciji

Na slici 13. prikazan je korak u konfiguraciji Zoiper softphone aplikacije u kojem se unosi adresa SIP servera. Kao hostname (domena) postavljena je adresa 127.0.0.1, koja označava lokalni SIP server pokrenut na istom računaru putem Asterisk platforme. Ovim unosom softphone klijent se usmjerava na odgovarajući SIP registrar i proxy server, čime se omogućava uspostavljanje signalizacijske komunikacije između klijenta i servera.



Slika 14. Automatska detekcija SIP transportnog protokola u Zoiper aplikaciji

Na slici 14. prikazan je proces automatskog testiranja dostupnih SIP transportnih protokola u okviru Zoiper softphone aplikacije. Tokom ovog procesa aplikacija provjerava podršku za različite transportne mehanizme prilikom povezivanja sa definisanim SIP serverom. Prikaz potvrđuje da je SIP UDP transport uspješno detektovan i dostupan za komunikaciju, dok ostali transporti (SIP TCP, SIP TLS i IAX UDP) nisu pronađeni ili nisu korišteni. Ovim korakom potvrđena je ispravna inicijalizacija SIP stacka i komunikacija softphone klijenta sa Asterisk SIP serverom putem UDP transporta.



Slika 15. Uspješno inicijalizovan SIP softphone klijent A

Na slici 15. prikazan je glavni interfejs Zoiper softphone aplikacije nakon završene konfiguracije SIP naloga. Zeleni indikator pored naloga 1001@127.0.0.1 potvrđuje da je softphone klijent A uspješno povezan sa lokalnim SIP serverom i da je SIP stack pravilno inicijalizovan. Ovim korakom potvrđena je spremnost klijenta za dalju SIP komunikaciju, uključujući registraciju na server i uspostavu poziva u narednim fazama projekta.

4.2. SIP registracija

U okviru ovog zadatka demonstriran je proces SIP registracije softphone klijenta A na Asterisk SIP server korištenjem PJSIP modula i Digest autentikacije. Cilj ovog zadatka bio je potvrditi ispravno funkcionisanje signalizacionog toka REGISTER poruka, kao i uspješnu autentikaciju korisnika definisanog u konfiguraciji servera.

Aktiviranje SIP logiranja na Asterisk serveru: Kako bi se pratio tok SIP signalizacije u realnom vremenu, na Asterisk serveru je aktivirano detaljno SIP logiranje korištenjem Asterisk CLI okruženja. U tu svrhu korištena je komanda `pjsip set logger on`. Ovom komandom omogućeno je prikazivanje svih dolaznih i odlaznih SIP poruka između softphone klijenta i SIP servera, što omogućava detaljnu analizu procesa registracije.

- Pokretanje SIP registracije putem softphone klijenta

Nakon aktivacije SIP logiranja, izvršena je ponovna prijava (login) na SIP nalog u Zoiper softphone aplikaciji. Time je softphone klijent inicirao SIP REGISTER zahtjev prema Asterisk serveru koristeći prethodno konfigurisane korisničke podatke (ekstenzija 1001).

Analiza SIP REGISTER toka

- Početni REGISTER zahtjev (bez autentikacije)

Prvi korak u procesu registracije je slanje REGISTER zahtjeva bez autentikacijskih podataka:

REGISTER sip:127.0.0.1 SIP/2.0

CSeq: 1 REGISTER

- Odgovor servera - 401 Unauthorized

Na početni REGISTER zahtjev, SIP server odgovara porukom:

SIP/2.0 401 Unauthorized

WWW-Authenticate: Digest realm="asterisk"

Ovaj odgovor predstavlja standardni mehanizam SIP Digest autentikacije. Server odbija prvi zahtjev i šalje izazov (challenge) koji sadrži parametre potrebne za autentikaciju, kao što su realm i nonce vrijednost.

- REGISTER zahtjev sa Digest autentikacijom

Nakon prijema izazova, softphone klijent ponovo šalje REGISTER zahtjev, ovaj put sa Digest autentikacijskim podacima:

Authorization: Digest username="1001", realm="asterisk"

U ovom zahtjevu klijent koristi korisničko ime i lozinku kako bi izračunao Digest odgovor, čime dokazuje svoj identitet serveru bez slanja lozinke u otvorenom tekstu.

- Uspješna registracija - 200 OK

Nakon validacije autentikacijskih podataka, SIP server potvrđuje registraciju:

SIP/2.0 200 OK

-- Added contact 'sip:1001@...'

== Endpoint 1001 is now Reachable

Poruka 200 OK označava uspješno završenu SIP registraciju. Server dodaje kontakt korisnika u svoju lokacijsku bazu podataka i označava SIP endpoint kao dostupan (Reachable).

- Periodična obnova registracije (refresh)

Tokom rada softphone klijenta primijećeni su dodatni REGISTER zahtjevi sa istekom (Expires: 60), kao i povremeni odgovori 401 Unauthorized sa oznakom stale=true, nakon čega slijedi ponovna uspješna autentikacija:

Expires: 60

CSeq: 2 REGISTER

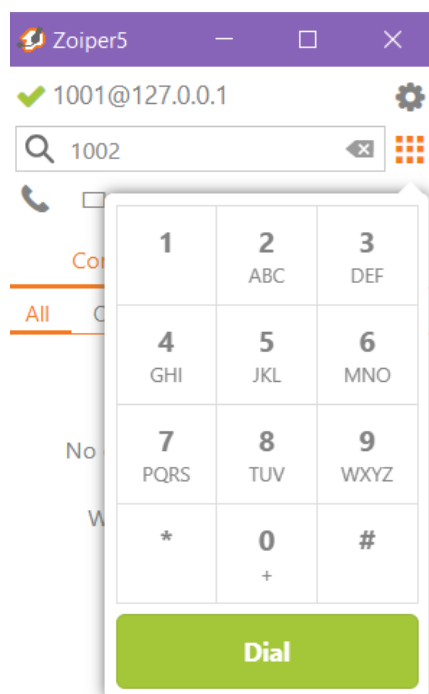
Ovo ponašanje predstavlja standardni mehanizam periodične obnove SIP registracije, kojim klijent održava aktivno stanje registracije na serveru. Navedeni odgovori ne predstavljaju grešku, već normalno ponašanje SIP protokola.

Isključivanje SIP logiranja: Nakon završene analize SIP registracije, detaljno SIP logiranje je isključeno komandom `pjsip set logger off`. Time je završeno praćenje signalizacije i vraćen Asterisk server u standardni režim rada.

4.3. Outgoing call

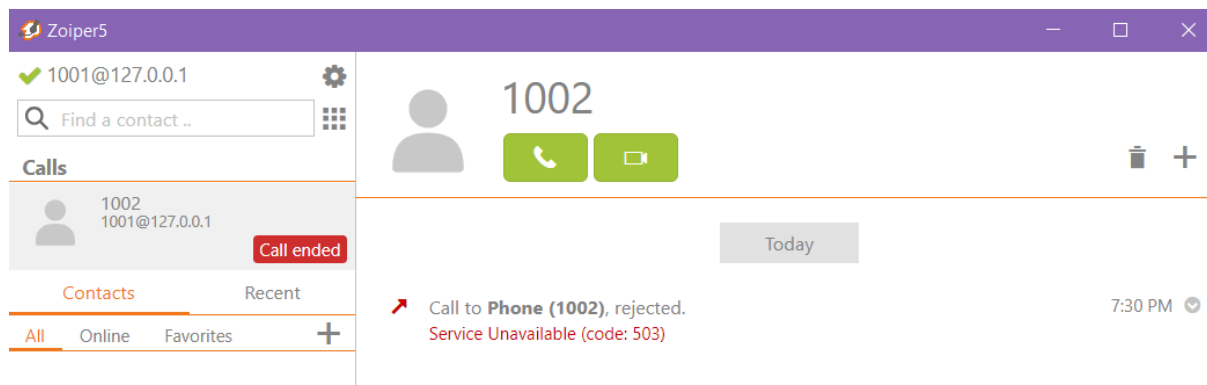
U okviru ovog zadatka testirano je iniciranje odlaznog SIP poziva sa softphone klijenta A (ekstenzija 1001) prema drugom korisniku u sistemu. Cilj zadatka bio je potvrditi da klijent ispravno generiše SIP INVITE poruku i da SIP server obrađuje signalizacioni tok poziva.

Radi analize signalizacije, na Asterisk serveru je aktivirano SIP logiranje pomoću komande `pjsip set logger on`. Nakon toga, iz Zoiper softphone aplikacije iniciran je poziv prema ekstenziji 1002.



Slika 15. Iniciranje poziva ka 1002

U SIP logovima zabilježen je dolazni INVITE zahtjev sa priloženim SDP opisom medija, čime je potvrđeno da je softphone klijent uspješno poslao pozivni zahtjev. Asterisk je započeo obradu poziva kroz dialplan, ali je poziv završen SIP odgovorom 503 Service Unavailable (slika 16), budući da ciljani korisnik (1002) u tom trenutku nije bio registrovan.



Slika 16. Poziv ka 1002 bez odgovora

Ovakvo ponašanje je očekivano i potvrđuje ispravno funkcionisanje SIP signalizacionog mehanizma za odlazne pozive. Na kraju testiranja, SIP logiranje je isključeno pomoću komande `pjsip set logger off`.

INVITE poruka: `INVITE sip:1002@127.0.0.1 SIP/2.0` potvrđuje da je softphone klijent 1001 inicirao outgoing poziv.

SDP u INVITE-u:

`m=audio 49706 RTP/AVP 106 9 98 101 0 8 3`

`a=rtpmap:106 opus/48000/2`

pokazuje da klijent nudi audio stream i kodeke, što znači da je SIP stack potpuno funkcionalan.

Dialplan se izvršava: `Executing [1002@internal:1] Dial("PJSIP/1001-00000000", "PJSIP/1002")`
Asterisk pokušava uspostaviti poziv prema 1002.

5. Softphone klijent B (User Agent 2)

U ovom poglavlju obrađena je implementacija i testiranje softphone klijenta B, koji u okviru sistema ima ulogu SIP User Agent Server-a (UAS), odnosno korisnika koji prima dolazne pozive. Ovo poglavlje predstavlja logičan nastavak prethodnog dijela, u kojem je obrađen softphone klijent A kao inicijator poziva.

Softphone klijent B koristi istu SIP infrastrukturu i server kao klijent A, pri čemu su osnovni koraci inicijalizacije SIP stacka i registracije na SIP server identični. Klijent B se registruje na Asterisk server korištenjem SIP REGISTER mehanizma uz Digest autentikaciju, čime postaje dostupan za prijem poziva unutar sistema.

Za razliku od klijenta A, čiji je fokus bio na slanju SIP INVITE poruka i iniciranju komunikacije, softphone klijent B je zadužen za obradu dolaznih SIP zahtjeva. Tokom testiranja, klijent B prima SIP INVITE poruku, generiše odgovore 180 Ringing i 200 OK, te time potvrđuje prihvatanje poziva i uspostavu SIP dijaloga.

Na kraju ovog poglavlja očekuje se da su oba softphone klijenta uspješno registrovana i da je omogućena dvosmjerna SIP komunikacija, čime se demonstrira kompletan tok uspostave poziva između dva korisnika u VoIP sistemu baziranom na SIP protokolu.

5.1. Inicijalizacija SIP stack-a

U okviru ovog zadatka izvršena je inicijalizacija SIP stacka za softphone klijent B (User Agent 2). Klijent B je realizovan na Linux virtualnoj mašini pokrenutoj sa eksternog SSD-a, dok je SIP server (Asterisk) ostao pokrenut na host računaru, čime je ostvareno distribuirano okruženje sa odvojenim korisničkim agentima.

Softphone klijent B koristi istu SIP infrastrukturu i PJSIP konfiguraciju kao i klijent A, ali se na SIP server povezuje preko njegove mrežne IP adrese. Nakon provjere mrežne povezanosti, instalirana je i pokrenuta Zoiper softphone aplikacija, te je konfigurisan SIP nalog sa ekstenzijom 1002 koristeći SIP preko UDP-a.

Uspješnim pokretanjem softphone aplikacije obezbijedena je inicijalna SIP funkcionalnost za klijenta B, čime su ispunjeni uslovi za nastavak procesa SIP registracije i testiranja dolaznih poziva.

```
C:\Users\Lenovo PC>ipconfig

Windows IP Configuration

Ethernet adapter vEthernet (Default Switch):

    Connection-specific DNS Suffix  . : 
    Link-local IPv6 Address . . . . . : fe80::6841:dc7d:3886:1258%47
    IPv4 Address. . . . . : 172.28.32.1
    Subnet Mask . . . . . : 255.255.240.0
    Default Gateway . . . . . : 

Ethernet adapter Ethernet:

    Connection-specific DNS Suffix  . : 
    Link-local IPv6 Address . . . . . : fe80::703e:fe05:bcbe:2d49%8
    IPv4 Address. . . . . : 192.168.0.15
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.0.1

Ethernet adapter Ethernet 2:

    Connection-specific DNS Suffix  . : 
    Link-local IPv6 Address . . . . . : fe80::4041:d21f:e34c:7ad6%20
    IPv4 Address. . . . . : 192.168.56.1
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 

Wireless LAN adapter Lokalna veza* 1:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . : 

Wireless LAN adapter Lokalna veza* 10:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . : 

Wireless LAN adapter Wi-Fi:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . : 

Ethernet adapter vEthernet (WSL):

    Connection-specific DNS Suffix  . : 
    Link-local IPv6 Address . . . . . : fe80::978b:96c8:5851:578a%59
    IPv4 Address. . . . . : 172.19.32.1
    Subnet Mask . . . . . : 255.255.240.0
```

Slika 17. Prikaz mrežnih interface-a i IP adresa host računara

Na slici 17. prikazan je izlaz komande ipconfig, koja je korištena za identifikaciju IP adrese host računara na kojem je pokrenut Asterisk SIP server. Cilj ovog koraka bio je utvrditi ispravnu IPv4 adresu dostupnu iz Linux virtualne mašine, kako bi se mogla provjeriti mrežna povezanost između softphone klijenta B i SIP servera. Identifikovana IP adresa korištena je u narednom koraku za testiranje mrežne konekcije putem ping komande i kasnije za konfiguraciju SIP server adrese u softphone aplikaciji.

```
kali@kali: ~  
File Actions Edit View Help  
(kali@kali)-[~]  
$ ping 192.168.0.15  
PING 192.168.0.15 (192.168.0.15) 56(84) bytes of data.  
64 bytes from 192.168.0.15: icmp_seq=1 ttl=255 time=4.18 ms  
64 bytes from 192.168.0.15: icmp_seq=2 ttl=255 time=1.43 ms  
64 bytes from 192.168.0.15: icmp_seq=3 ttl=255 time=1.37 ms  
64 bytes from 192.168.0.15: icmp_seq=4 ttl=255 time=1.12 ms  
64 bytes from 192.168.0.15: icmp_seq=5 ttl=255 time=2.80 ms  
64 bytes from 192.168.0.15: icmp_seq=6 ttl=255 time=1.27 ms  
64 bytes from 192.168.0.15: icmp_seq=7 ttl=255 time=1.10 ms  
64 bytes from 192.168.0.15: icmp_seq=8 ttl=255 time=1.22 ms  
64 bytes from 192.168.0.15: icmp_seq=9 ttl=255 time=0.888 ms  
64 bytes from 192.168.0.15: icmp_seq=10 ttl=255 time=1.23 ms  
64 bytes from 192.168.0.15: icmp_seq=11 ttl=255 time=1.23 ms  
64 bytes from 192.168.0.15: icmp_seq=12 ttl=255 time=0.871 ms  
64 bytes from 192.168.0.15: icmp_seq=13 ttl=255 time=0.982 ms  
64 bytes from 192.168.0.15: icmp_seq=14 ttl=255 time=1.02 ms  
64 bytes from 192.168.0.15: icmp_seq=15 ttl=255 time=1.33 ms  
64 bytes from 192.168.0.15: icmp_seq=16 ttl=255 time=2.04 ms  
64 bytes from 192.168.0.15: icmp_seq=17 ttl=255 time=1.03 ms  
64 bytes from 192.168.0.15: icmp_seq=18 ttl=255 time=1.41 ms  
64 bytes from 192.168.0.15: icmp_seq=19 ttl=255 time=2.23 ms  
64 bytes from 192.168.0.15: icmp_seq=20 ttl=255 time=3.05 ms  
64 bytes from 192.168.0.15: icmp_seq=21 ttl=255 time=1.10 ms  
64 bytes from 192.168.0.15: icmp_seq=22 ttl=255 time=1.42 ms  
64 bytes from 192.168.0.15: icmp_seq=23 ttl=255 time=1.60 ms  
64 bytes from 192.168.0.15: icmp_seq=24 ttl=255 time=1.24 ms  
64 bytes from 192.168.0.15: icmp_seq=25 ttl=255 time=1.67 ms
```

Slika 18. Provjera mrežne povezanosti Client B

Prije konfiguracije SIP klijenta B, izvršena je provjera mrežne dostupnosti Asterisk PBX servera sa Linux virtualne mašine. Korištenjem ping komande potvrđeno je da je IP adresa Asteriska (192.168.0.15) dostupna iz VM okruženja bez gubitka paketa i sa niskom latencijom. Time je potvrđeno da je mrežna konfiguracija virtualne mašine ispravna i da postoje svi preduslovi za inicijalizaciju SIP stacka na klijentu B.

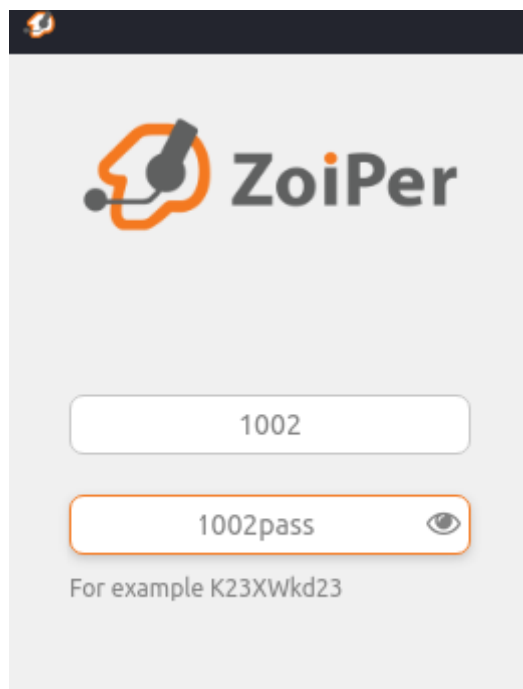
```
(kali@kali)-[~/Downloads]  
$ sudo dpkg -i Zoiper*.deb  
[sudo] password for kali:  
Selecting previously unselected package zoiper5.  
(Reading database ... 408445 files and directories currently installed.)  
Preparing to unpack Zoiper5_5.6.12_x86_64.deb ...  
Unpacking zoiper5 (5.6.12) ...  
Setting up zoiper5 (5.6.12) ...  
Processing triggers for kali-menu (2025.1.1) ...  
Processing triggers for desktop-file-utils (0.28-1) ...  
Processing triggers for mailcap (3.74) ...
```

Slika 19. Instalacija Zoiper aplikacije na Linux virtuelnoj mašini

Na slici 19. je prikazan proces instalacije Zoiper softphone aplikacije na Linux virtualnoj mašini korištenjem dpkg alata. Instalacija je izvršena iz .deb paketa preuzetog sa zvanične Zoiper stranice, pri čemu je komandom `sudo dpkg -i Zoiper*.deb` paket raspakovan i instaliran u sistem. Ovim korakom obezbijeđeno je softversko okruženje potrebno za pokretanje softphone klijenta B (User Agent 2), koji će se u narednim koracima registrovati na Asterisk SIP server i koristiti za prijem dolaznih poziva.

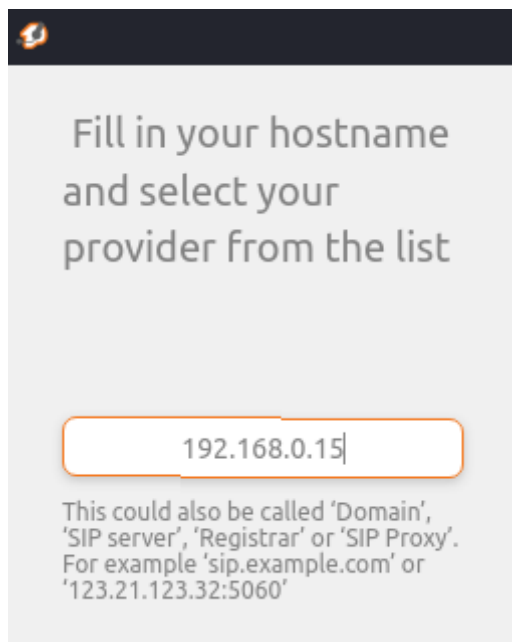
5.2. SIP registracija

U okviru ovog podpoglavlja izvršena je SIP registracija softphone klijenta B (ekstenzija 1002) na centralni SIP server baziran na Asterisk PBX-u. Nakon uspješne inicijalizacije SIP stacka i instalacije softphone aplikacije, klijent B je konfigurisan sa odgovarajućim korisničkim podacima i adresom SIP servera. Proces registracije realizovan je korištenjem SIP REGISTER mehanizma uz Digest autentikaciju, čime je omogućeno da klijent B postane dostupan unutar sistema za prijem dolaznih poziva. Uspješna registracija potvrđena je kroz Asterisk CLI izlaz, gdje je endpoint 1002 označen kao dostupan (reachable), što predstavlja preduslov za daljnje testiranje dolaznih poziva i kompletne SIP komunikacije.



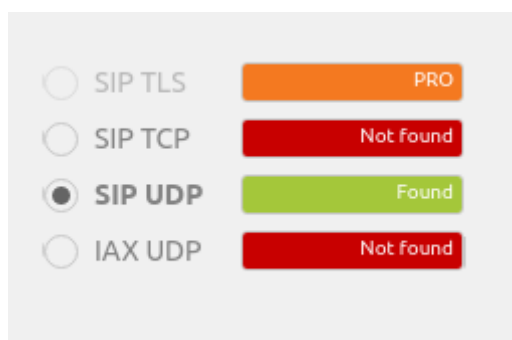
Slika 20. Unos korisničkih podataka za SIP nalog 1002

Na slici 20. prikazan je unos korisničkog imena i lozinke za softphone klijent B. Kao korisničko ime unesena je ekstenzija 1002, dok je lozinka definisana na Asterisk serveru - 1002pass. Ovim korakom započinje konfiguracija SIP registracije za drugi korisnički agent.



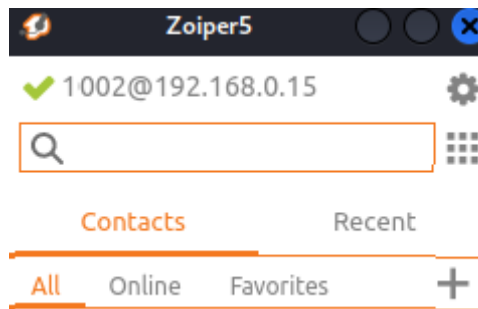
Slika 21. Definisanje SIP servera (Domain/Host)

Na slici 21. je prikazan unos IP adrese SIP servera. Kao domen je korištena IP adresa računara na kojem je pokrenut Asterisk PBX (192.168.0.15), čime se omogućava komunikacija između softphone klijenta B na Linux VM-u i centralnog SIP servera.



Slika 22. Automatska detekcija transportnog protokola

Zoiper aplikacija automatski testira dostupne SIP transportne protokole. Rezultat testiranja pokazuje da je SIP preko UDP-a dostupan i pronađen, dok ostali protokoli nisu aktivni. Time je potvrđeno da je SIP stack uspješno inicijalizovan koristeći UDP transport.



Slika 23. Uspješna SIP registracija B klijenta

Na slici 23. je prikazan glavni interfejs Zoiper aplikacije sa zelenom oznakom pored naloga 1002@192.168.0.15, što označava uspješnu SIP registraciju. Ovim je potvrđeno da je softphone klijent B aktivan i spreman za prijem dolaznih poziva.

```
089066949a6e*CLI> pjsip show endpoints

Endpoint: <Endpoint/CID.....> <State.....> <Channels.>
  I/OAuth: <AuthId/UserName.....>
  Aor: <Aor.....> <MaxContact>
  Contact: <Aor/ContactUri.....> <Hash.....> <Status> <RTT(ms)..>
  Transport: <TransportId.....> <Type> <cos> <tos> <BindAddress.....>
  Identify: <Identify/Endpoint.....>
  Match: <criteria.....>
  Channel: <ChannelId.....> <State.....> <Time.....>
  Exten: <DialedExten.....> CLCID: <ConnectedLineCID.....>
=====

Endpoint: 1001                                Not in use    0 of inf
  InAuth: auth1001/1001
  Aor: 1001                                    1
  Contact: 1001/sip:1001@172.17.0.1:32877;transport=U 1a9b4e4c40 NonQual -nan

Endpoint: 1002                                Not in use    0 of inf
  InAuth: auth1002/1002
  Aor: 1002                                    1
  Contact: 1002/sip:1002@172.17.0.1:47060;transport=U 5eccfa3de0 NonQual -nan

Objects found: 2
```

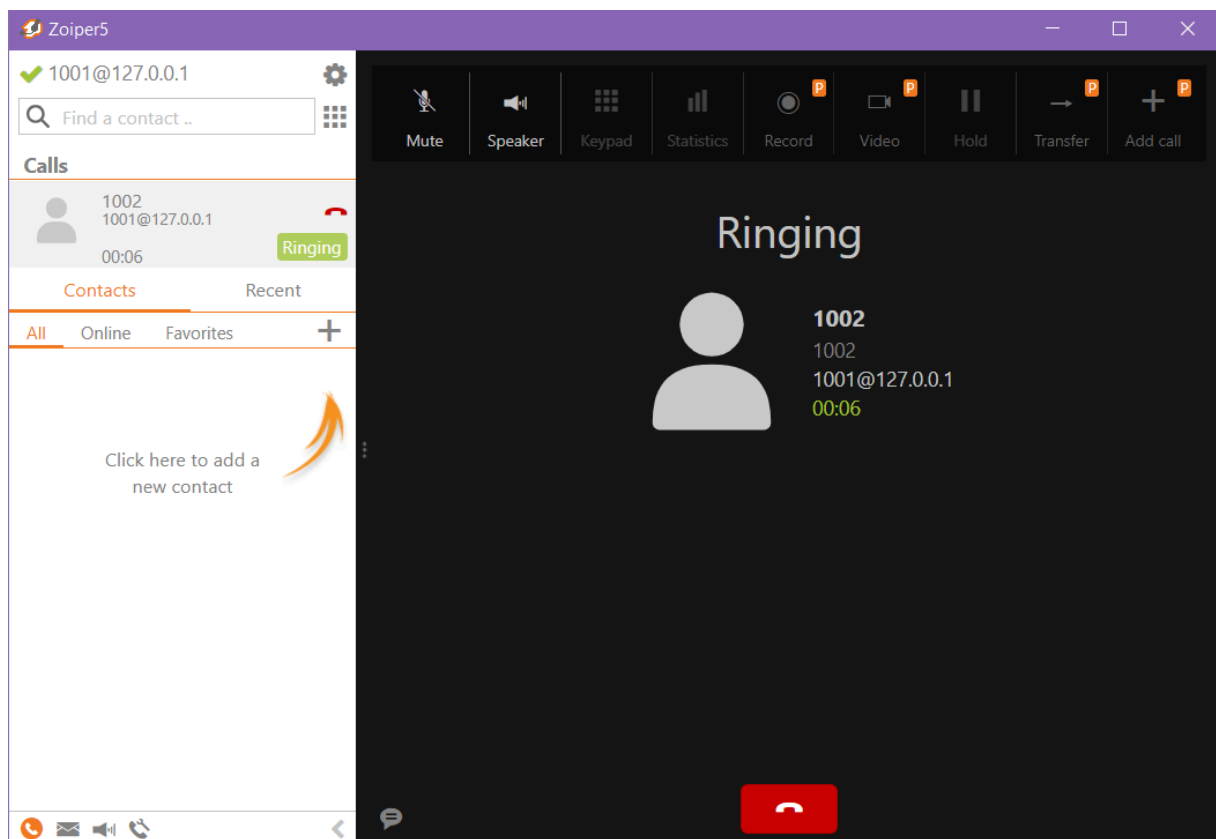
Slika 24. Provjera registracije n Asterisk serveru (host)

Prikaz rezultata komande `pjsip show endpoints` na Asterisk CLI-ju. Vidljivo je da su oba endpointa, 1001 i 1002, registrovana i dostupna. Endpoint 1002 ima aktivan kontakt, čime je potvrđena ispravna SIP registracija softphone klijenta B sa Linux VM-a.

5.3. Incoming call handling

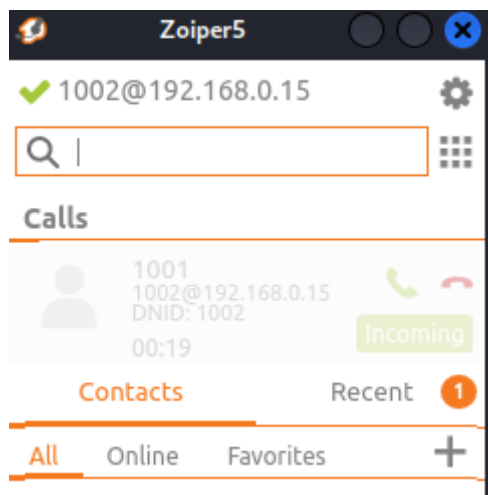
U okviru ovog zadatka testirana je obrada dolaznog SIP poziva na softphone klijentu B (User Agent 2). Nakon uspješne SIP registracije oba klijenta, klijent A inicira poziv prema ekstenziji 1002, dok klijent B preuzima ulogu SIP User Agent Server-a (UAS).

Tokom ovog procesa klijent B prima SIP INVITE poruku, generiše odgovor 180 Ringing, a zatim, nakon prihvatanja poziva, šalje 200 OK, čime se uspostavlja SIP dijalog i RTP komunikacija. Na ovaj način demonstriran je kompletan tok prijema i prihvatanja dolaznog poziva u VoIP sistemu baziranom na SIP protokolu.



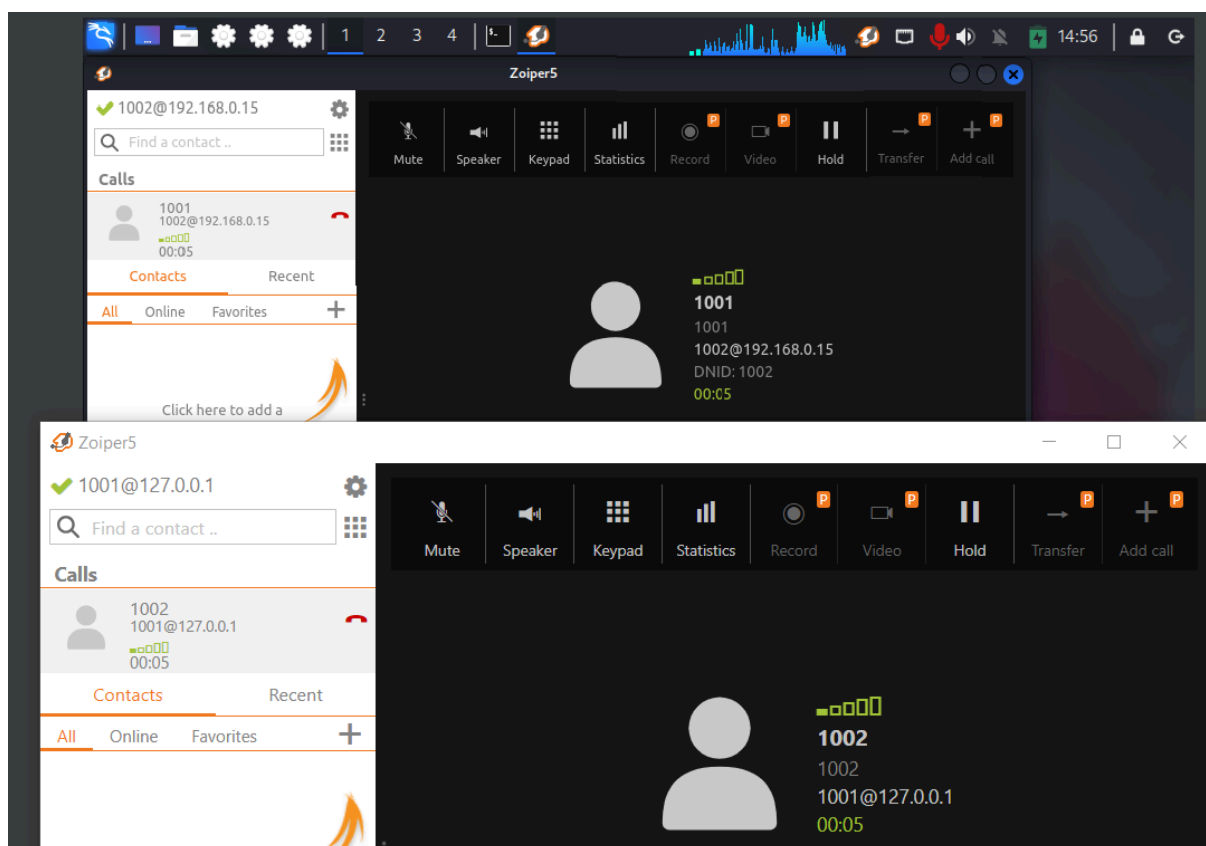
Slika 25. Odlazni poziv na klijentu A

Na slici 25. je prikazan aktivan poziv iniciran sa softphone klijenta A prema ekstenziji 1002. Klijent A šalje SIP INVITE poruku prema SIP serveru, nakon čega se poziv nalazi u stanju Ringing, čekajući odgovor klijenta B. Ovaj korak potvrđuje uspješno slanje poziva i ispravnu SIP signalizaciju sa strane pozivaoca.



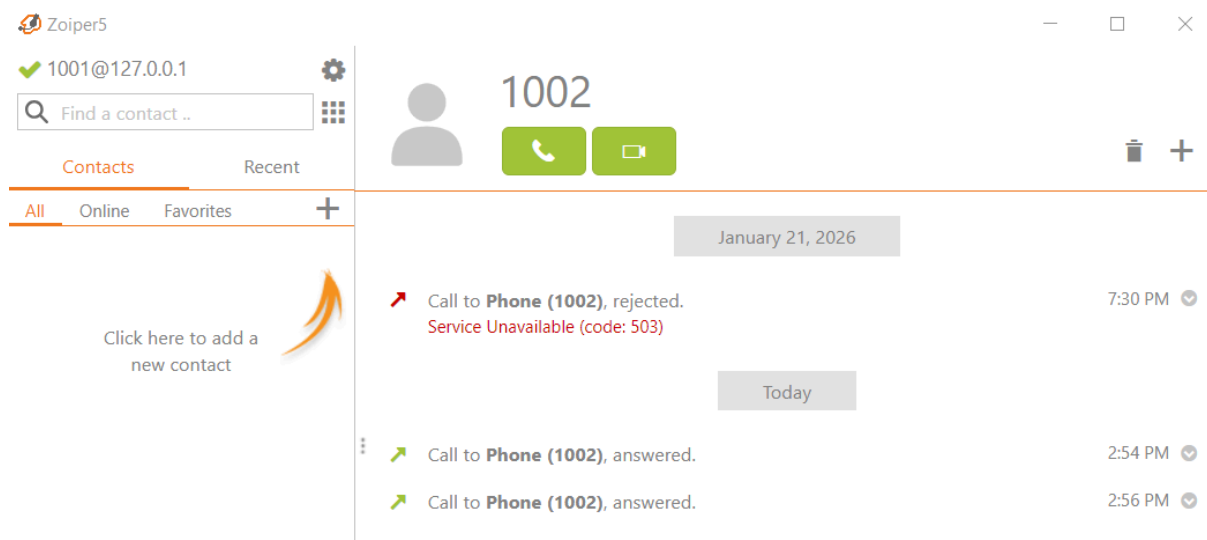
Slika 26. Prikaz dolaznog poziva na klijentu B

Slika 26. prikazuje dolazni poziv na softphone klijentu B (1002). U ovom trenutku klijent B je zaprimio SIP INVITE zahtjev i sistem je u fazi signalizacije dolaznog poziva. Ovo odgovara slanju SIP odgovora 180 Ringing, kojim se pozivaocu signalizira da krajnji korisnik prima poziv.



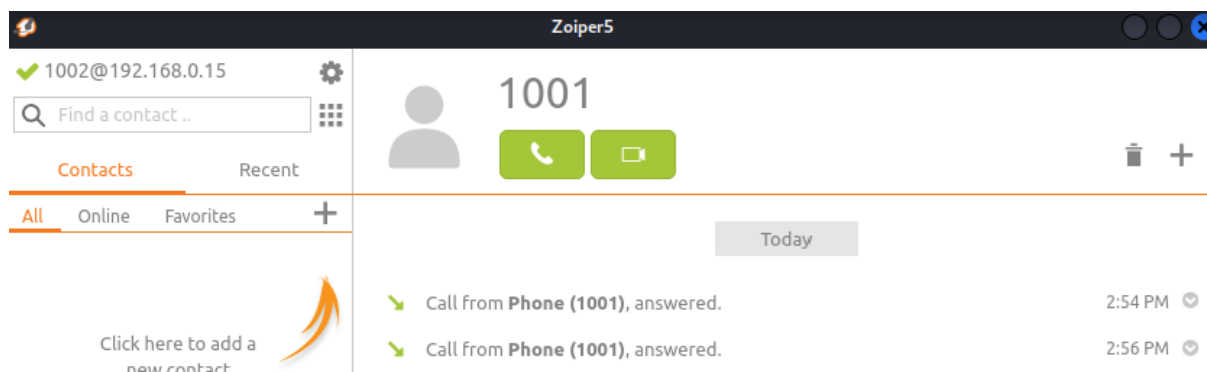
Slika 27. Uspostavljen poziv između klijenata A i B

Na slici 27. je prikazan uspješno uspostavljen poziv između softphone klijenata A (1001) i B (1002). Nakon prihvatanja poziva od strane klijenta B, poslan je SIP odgovor 200 OK, čime je uspostavljen SIP dijalog i započeta dvosmjerna RTP komunikacija. Oba klijenta prikazuju aktivno trajanje poziva, što potvrđuje uspješnu uspostavu veze.



Slika 28. Evidencija pokušaja uspostavljanja poziva na klijentu A

Slika 28. prikazuje historiju poziva na softphone klijentu A. Vidljiv je raniji neuspješan pokušaj poziva (Service Unavailable – 503), kao i kasniji uspješno realizovani pozivi prema ekstenziji 1002. Ovaj prikaz potvrđuje da je sistem pravilno reagovao na različita stanja SIP signalizacije i da je konačna konfiguracija ispravna.



Slika 29. Evidencija primljenih poziva na klijentu B

Na slici 29. je prikazana evidencija primljenih poziva na softphone klijentu B. Zabilježeni su uspješno prihvaćeni pozivi sa ekstenzije 1001, čime se potvrđuje da klijent B pravilno funkcionira kao SIP User Agent Server i da može primiti i obrađivati dolazne pozive.

Tokom uspješnog prijema i prihvatanja poziva, u Asterisk CLI-ju (uz uključen pjsip set logger on) očekuju se sljedeći ključni događaji:

- Prijem SIP INVITE poruke:

<--- Received SIP request --->

INVITE sip:1002@IP_ASTERISKA SIP/2.0

- Slanje odgovora 100 Trying:

SIP/2.0 100 Trying

- Slanje odgovora 180 Ringing:

SIP/2.0 180 Ringing

- Slanje odgovora 200 OK nakon prihvatanja poziva:

SIP/2.0 200 OK

- Razmjena ACK poruke i uspostava dijaloga:

ACK sip:1002@IP_ASTERISKA SIP/2.0

- Izvršavanje dial plana:

Executing [1002@internal:1] Dial("PJSIP/1001-...", "PJSIP/1002")

Prisustvo ovih poruka u logovima potvrđuje da je dolazni poziv pravilno obrađen, da je SIP dijalog uspješno uspostavljen i da je zadatak Incoming call handling uspješno realizovan.

6. Medij i sesija

U okviru ovog zadatka analizirana je SDP razmjena koja se odvija tokom uspostave SIP poziva između softphone klijenata. SDP informacije su prenesene unutar SIP INVITE i SIP 200 OK poruka i korištene su za dogovor medijskih parametara komunikacije.

SDP razmjenom definisani su podržani audio codec-i, kao i IP adresa i UDP port koji se koriste za RTP prenos audio signala. Klijent A je u INVITE poruci poslao SDP ponudu sa listom podržanih codec-a, dok je klijent B u odgovoru prihvatio kompatibilan codec, čime je omogućen RTP tok između krajnjih tačaka.

Prisustvo SDP sadržaja u SIP porukama potvrđeno je putem Asterisk PJSIP logera, čime je ispunjen zahtjev zadatka.

6.1. SDP razmjena

U okviru ovog zadatka analizirana je SDP razmjena (Session Description Protocol) koja se odvija prilikom uspostave SIP poziva između softphone klijenata. SDP se ne šalje kao poseban protokol, nego se prenosi kao sadržaj unutar SIP poruka i služi za dogovor medijskih parametara razgovora (audio).

6.1.1. Gdje se SDP pojavljuje u SIP komunikaciji

SDP se pojavljuje u sljedećim fazama uspostave poziva:

1. SIP INVITE (SDP offer / ponuda)

Pozivajući klijent (npr. 1001) šalje INVITE poruku koja u sebi sadrži SDP ponudu. U toj ponudi navodi:

- koje audio codec-e podržava
- na kojoj IP adresi očekuje RTP tok
- koji UDP port koristi za RTP audio

2. SIP 200 OK (SDP answer / odgovor)

Pozvani klijent (npr. 1002) vraća 200 OK, također sa SDP sadržajem, gdje potvrđuje:

- koji codec prihvata (iz liste ponuđenih)
- na kojoj IP adresi i portu će slati/prijemati RTP

Na ovaj način SIP signalizacija dogovara mediju, dok se sam audio prenosi odvojeno putem RTP-a.

6.1.2. Kako je SDP provjeren

Da bi SDP bio vidljiv u komunikaciji, u Asterisk CLI je uključen PJSIP logger `pjsip set logger on`. Nakon uključivanja logera, u CLI se prikazuju kompletne SIP poruke koje dolaze i odlaze, uključujući i SDP dio (Content-Type: application/sdp).

6.1.3. Primjer SDP ponude (offer) iz SIP INVITE poruke

SDP koji se pojavljuje u INVITE poruci izgleda ovako:

```
v=0
o=Z 0 19120803 IN IP4 127.0.0.1
s=Z
c=IN IP4 127.0.0.1
t=0 0
m=audio 49706 RTP/AVP 106 9 98 101 0 8 3
a=rtpmap:106 opus/48000/2
a=fmtp:106 sprop-maxcapture=16000; minptime=20; useinbandfec=1
a=rtpmap:98 telephone-event/48000
a=fmtp:98 0-16
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
a=sendrecv
a=rtcp-mux
```

1. v=0 - SDP verzija (standardno)
2. o=Z 0 19120803 IN IP4 127.0.0.1 - origin (ko je kreirao SDP); identifikacija izvora SDP-a (agent koji šalje poruku), te uključuje IP adresu izvora
3. s=Z - naziv sesije
4. c=IN IP4 127.0.0.1 - connection data, IP adresa na koju druga strana treba slati RTP audio

5. `t=0 0` - timing, sesija je aktivna bez definisanog vremena isteka (standardno u VoIP pozivima)
6. `m=audio 49706 RTP/AVP 106 9 98 101 0 8 3` - media opis (audio + RTP port + codec lista)

Ovo je ključni dio:

- audio → vrsta medije (audio)
 - 49706 → UDP port za RTP audio tok
 - RTP/AVP → RTP profil
 - 106 9 98 101 0 8 3 → lista ponuđenih codec payload tipova (među njima su:
 - 106 = Opus
 - 0 = PCMU (G.711 μ -law)
 - 9 = G.722
 - 98/101 = telephone-event (DTMF tonovi))
7. `a=rtpmap:106 opus/48000/2` - mapiranje codec-a, payload 106 je Opus, sample rate 48000 Hz, stereo (2 kanala)
 8. `a=fmtp:106 sprop-maxcapture=16000; minptime=20; useinbandfec=1`
 9. `a=rtpmap:98 telephone-event/48000` - telephone-event služi za DTMF (tipke), nije "glasovni" codec
 10. `a=rtpmap:101 telephone-event/8000` - telephone-event služi za DTMF (tipke), nije "glasovni" codec
 11. `a=sendrecv` - smjer medija, obje strane i šalju i primaju audio (dvosmjerni poziv).
 12. `a=rtcp-mux` - RTCP multiplexing, RTP i RTCP se koriste preko istog porta (često kod softphone klijenata).

6.1.4. Codec ponuda/odgovor (offer/answer)

- Klijent A (1001) u INVITE poruci šalje SDP ponudu sa listom codec-a koje podržava.
- Klijent B (1002) u 200 OK poruci šalje SDP odgovor gdje:
 - bira jedan codec iz ponuđenih (kompatibilan)
 - šalje svoj IP i port za RTP

Time je codec dogovoren prije nego što audio krene.

7. Testiranje i validacija

7.1. SIP logiranja

U okviru ovog zadatka izvršeno je SIP logiranje na strani SIP servera korištenjem Asterisk CLI okruženja. Cilj logiranja je bio detaljan uvid u SIP signalizaciju koja se razmjenjuje između softphone klijenata i SIP servera tokom registracije i uspostave poziva.

Za potrebe analize korišten je ugrađeni PJSIP logger, koji omogućava prikaz kompletnih SIP poruka u realnom vremenu, uključujući REGISTER, INVITE, ACK, kao i odgovore servera (100 Trying, 180 Ringing, 200 OK, 401 Unauthorized i sl.).

SIP logiranje je aktivirano u Asterisk CLI konzoli sljedećom komandom: `pjsip set logger on`.

Nakon aktivacije logera, u CLI su se počele prikazivati sve dolazne i odlazne SIP poruke. Tokom testiranja zabilježeni su:

- SIP REGISTER zahtjevi i odgovori (Digest autentikacija),
- SIP INVITE poruke za iniciranje poziva,
- SIP odgovori 180 Ringing i 200 OK,
- ACK poruke za potvrdu uspostave sesije,
- SDP sadržaj unutar SIP poruka.

Logovi su korišteni za potvrdu ispravnog toka SIP signalizacije i pravilne konfiguracije oba softphone klijenta. Nakon završetka analize, SIP logger je isključen sljedećom komandom: `pjsip set logger off`.

7.2. Wireshark analiza

U okviru ovog zadatka izvršena je analiza SIP signalizacije korištenjem Wireshark-a. Tokom aktivnog VoIP poziva između softphone klijenata 1001 i 1002 sniman je mrežni saobraćaj, uz primjenu odgovarajućeg filtera za SIP protokol. Cilj analize bio je potvrditi razmjenu SIP poruka koje učestvuju u upravljanju pozivom, kao i ispravan završetak komunikacije. Analizom snimljenog saobraćaja potvrđeno je da SIP signalizacija prolazi kroz centralni SIP server (Asterisk) i da se poziv pravilno zatvara slanjem odgovarajućih SIP poruka.

No.	Time	Source	Destination	Protocol	Length	Info
2931	50.739269	192.168.0.15	172.17.0.2	SIP	446	Request: ACK sip:172.17.0.2:5060
2940	50.964369	192.168.0.15	172.17.0.2	SIP	446	Request: ACK sip:172.17.0.2:5060
2991	51.928707	192.168.0.15	172.17.0.2	SIP	446	Request: ACK sip:172.17.0.2:5060
3410	53.926680	192.168.0.15	172.17.0.2	SIP	446	Request: ACK sip:172.17.0.2:5060
3611	57.608012	192.168.0.15	172.17.0.2	SIP	493	Request: BYE sip:asterisk@172.17.0.2:5060
3629	57.938309	192.168.0.15	172.17.0.2	SIP	446	Request: ACK sip:172.17.0.2:5060
3699	58.112094	192.168.0.15	172.17.0.2	SIP	493	Request: BYE sip:asterisk@172.17.0.2:5060
3753	59.108418	192.168.0.15	172.17.0.2	SIP	493	Request: BYE sip:asterisk@172.17.0.2:5060
3771	59.553259	192.168.0.15	172.17.0.2	SIP	727	Request: BYE sip:172.17.0.2:5060
3772	60.042526	192.168.0.15	172.17.0.2	SIP	727	Request: BYE sip:172.17.0.2:5060
4077	61.043440	192.168.0.15	172.17.0.2	SIP	727	Request: BYE sip:172.17.0.2:5060
4092	61.111047	192.168.0.15	172.17.0.2	SIP	493	Request: BYE sip:asterisk@172.17.0.2:5060
4093	61.928710	192.168.0.15	172.17.0.2	SIP	446	Request: ACK sip:172.17.0.2:5060
4103	63.043956	192.168.0.15	172.17.0.2	SIP	727	Request: BYE sip:172.17.0.2:5060
4108	65.113173	192.168.0.15	172.17.0.2	SIP	493	Request: BYE sip:asterisk@172.17.0.2:5060
4212	65.922596	192.168.0.15	172.17.0.2	SIP	446	Request: ACK sip:172.17.0.2:5060
4232	67.044584	192.168.0.15	172.17.0.2	SIP	727	Request: BYE sip:172.17.0.2:5060
4305	69.114187	192.168.0.15	172.17.0.2	SIP	493	Request: BYE sip:asterisk@172.17.0.2:5060
4310	69.924315	192.168.0.15	172.17.0.2	SIP	446	Request: ACK sip:172.17.0.2:5060
4317	71.049716	192.168.0.15	172.17.0.2	SIP	727	Request: BYE sip:172.17.0.2:5060
4608	73.115203	192.168.0.15	172.17.0.2	SIP	493	Request: BYE sip:asterisk@172.17.0.2:5060
4609	73.921347	192.168.0.15	172.17.0.2	SIP	446	Request: ACK sip:172.17.0.2:5060
4630	75.050120	192.168.0.15	172.17.0.2	SIP	727	Request: BYE sip:172.17.0.2:5060
4729	77.115671	192.168.0.15	172.17.0.2	SIP	493	Request: BYE sip:asterisk@172.17.0.2:5060
4743	77.923384	192.168.0.15	172.17.0.2	SIP	446	Request: ACK sip:172.17.0.2:5060
4862	79.051490	192.168.0.15	172.17.0.2	SIP	727	Request: BYE sip:172.17.0.2:5060
4869	81.116934	192.168.0.15	172.17.0.2	SIP	493	Request: BYE sip:asterisk@172.17.0.2:5060
4878	81.920862	192.168.0.15	172.17.0.2	SIP	446	Request: ACK sip:172.17.0.2:5060
4879	83.051336	192.168.0.15	172.17.0.2	SIP	727	Request: BYE sip:172.17.0.2:5060
5147	85.116920	192.168.0.15	172.17.0.2	SIP	493	Request: BYE sip:asterisk@172.17.0.2:5060
5272	87.052774	192.168.0.15	172.17.0.2	SIP	727	Request: BYE sip:172.17.0.2:5060
5348	89.118650	192.168.0.15	172.17.0.2	SIP	493	Request: BYE sip:asterisk@172.17.0.2:5060
5360	91.052619	192.168.0.15	172.17.0.2	SIP	727	Request: BYE sip:172.17.0.2:5060

Slika 30. Wireshark prikaz SIP signalizacije tokom VoIP poziva

Na slici 30. je prikazan Wireshark snimak mrežnog saobraćaja filtriran pomoću izraza sip || rtp. Vidljive su SIP poruke razmijenjene između softphone klijenta i Asterisk SIP servera, uključujući ACK poruke koje potvrđuju uspostavljeni SIP dijalog, kao i BYE poruke koje označavaju korektan završetak poziva. Prisutnost ovih poruka potvrđuje da je SIP signalizacija ispravno funkcionisala tokom trajanja VoIP sesije.

8. ZAKLJUČAK

U okviru ovog projekta realizovana je kompletna implementacija i analiza VoIP sistema zasnovanog na SIP protokolu, koristeći Asterisk kao centralni SIP server i softphone klijente kao krajnje korisničke agente. Kroz niz zadataka obuhvaćeni su svi ključni koraci potrebni za uspostavu SIP komunikacije, od inicijalne konfiguracije servera i korisnika, preko SIP registracije i uspostave poziva, do detaljne analize signalizacije i medijskog toka.

Konfiguracijom SIP ekstenzija i dialplana omogućena je komunikacija između dva korisnika (1001 i 1002), pri čemu je uspješno demonstrirana primjena SIP REGISTER mehanizma uz Digest autentikaciju. Testiranjem dolaznih i odlaznih poziva potvrđena je ispravna razmjena SIP poruka (INVITE, 180 Ringing, 200 OK, ACK i BYE), kao i pravilno upravljanje životnim ciklusom poziva.

Dodatno, izvršena je analiza SDP razmjene, čime je potvrđena ispravna razmjena informacija o kodecima, IP adresama i RTP portovima, što je ključno za uspostavu audio komunikacije. Korištenjem Asterisk CLI logiranja i alata Wireshark omogućena je detaljna analiza SIP signalizacije na mrežnom nivou, čime je potvrđena transparentnost i ispravnost kompletnog komunikacijskog toka.

Implementacijom softphone klijenata na različitim sistemima i korištenjem virtualne mašine demonstrirana je realistična arhitektura VoIP sistema sa centralizovanim SIP serverom i distribuiranim korisničkim agentima. Time je projekat uspješno ispunio sve definisane ciljeve i pružio praktičan uvid u funkcionisanje SIP/VoIP sistema u realnom okruženju.

9. REFERENCE

- [1] DNAKE Global: What is SIP Intercom & Why You Need It. Dostupno na: <https://www.dnake-global.com/bs/news/what-is-sip-intercom-why-you-need-it/> Pristupano: 21. 1. 2026.
- [2] D. Androić: Session Initiation Protocol. Dostupno na: https://www.phy.pmf.unizg.hr/~dandroic/nastava/mr/session_initiation_protocol.pdf Pristupano: 21. 1. 2026.
- [3] CARNET SysPortal: Session Initiation Protocol (SIP). Dostupno na: <https://sysportal.carnet.hr/node/352> Pristupano: 21. 1. 2026.
- [4] S. Gros: Session Initiation Protocol (SIP). Dostupno na: <https://www.zemris.fer.hr/~sgros/courses/cn/2002-2003/seminari/finished/pdf/sip.pdf> Pristupano: 21. 1. 2026.
- [5] K. Horvat: Prijenos zvuka preko Internet protokola (AoIP), završni rad, Veleučilište Velika Gorica, Velika Gorica.
- [6] Repozitorij ETfOS: Rad na temu VoIP/SIP. Dostupno na: <https://repozitorij.etfos.hr/object/etfos:3443/FILE0> Pristupano: 21. 1. 2026.
- [7] P. Zave: Under-specification and SIP. Dostupno na: <https://pamelazave.com/under2.pdf> Pristupano: 21. 1. 2026.
- [8] M. Handley et al.: Analysis of SIP-based communication. Dostupno na: <https://arxiv.org/pdf/1205.0480> Pristupano: 21. 1. 2026.
- [9] M. Basiccevic: PhD Thesis – SIP and IMS Systems. Dostupno na: https://www.rt-rk.uns.ac.rs/sites/default/files/Basicevic_phd_thesis.pdf Pristupano: 21. 1. 2026.
- [10] Visoka škola elektrotehnike i računarstva Beograd: Lekcija 4 – SIP. Dostupno na: [https://www.viser.edu.rs/uploads/2018/03/Lekcija4%20\(2\).pdf](https://www.viser.edu.rs/uploads/2018/03/Lekcija4%20(2).pdf) Pristupano: 21. 1. 2026.
- [11] Fakultet elektrotehnike i računarstva, Univerzitet u Zagrebu: VoIP – SIP i H.323. Dostupno na: https://www.fer.unizg.hr/_download/repository/VMK_2018_14_VoIP_SIP_H323.pdf Pristupano: 21. 1. 2026.
- [12] IETF: RFC 3261 – Session Initiation Protocol. Dostupno na: <http://www.faqs.org/rfcs/rfc3261.html> Pristupano: 21. 1. 2026.

- [13] ETSI: IP Multimedia Subsystem (IMS); SIP-based Interfaces. Dostupno na: https://www.etsi.org/deliver/etsi_ts/123200_123299/123278/18.00.00_60/ts_123278v180000p.pdf Pristupano: 21. 1. 2026.
- [14] G. Camarillo, M. Garcia-Martin: The 3G IP Multimedia Subsystem (IMS): Merging the Internet and the Cellular Worlds, Wiley, 2006.
- [15] E. Rosenberg: SIP Security, Wiley, 2009. Dostupno na: <https://books.google.ba/books?id=2ul9Xtc13IIC> Pristupano: 21. 1. 2026.
- [16] J. Rosenberg: SIP: Understanding the Session Initiation Protocol. Dostupno na: <https://archive.org/details/sipunderstanding0000john> Pristupano: 21. 1. 2026.