

**ALFA BK UNIVERZITET**  
**FAKULTET INFORMACIONIH TEHNOLOGIJA**



**RAZVOJ SIMULACIONOG SISTEMA ZA VOIP I  
OBRADU SIGNALA ZASNOVANOG NA AI  
ALGORITMIMA PRIMENOM AGILNOG PRISTUPA  
RAZVOJU**

**- doktorska disertacija -**

**KANDIDAT**

Mario Stojanović 2022/5903

**MENTOR**

prof. dr Milan Gligorijević

**Beograd, 2026.**

**ALFA BK UNIVERSITY**  
**FACULTY OF INFORMATION TECHNOLOGIES**



**DEVELOPMENT OF A SIMULATION SYSTEM FOR  
VOIP AND SIGNAL PROCESSING BASED ON AI  
ALGORITHMS USING AN AGILE DEVELOPMENT  
METHODOLOGY**

**- Doctoral Dissertation -**

**PHD CANDIDATE**

Mario Stojanović 2022/5903

**PHD SUPERVISOR**

Milan Gligorijević, PhD Professor

**Belgrade, 2026.**

**ALFA BK UNIVERZITET**  
**FAKULTET INFORMACIONIH TEHNOLOGIJA**



Datum \_\_\_\_\_

Komisija za ocenu i javnu odbranu doktorske disertacije:

prof. dr Milan Gligorijević - predsednik Komisije, Fakultet informacionih tehnologija, Alfa BK Univerzitet

\_\_\_\_\_

prof. dr Vladimir Mladenović - član Komisije, Fakultet tehničkih nauka Čačak, Univerzitet u Kragujevcu

\_\_\_\_\_

prof. dr Dejan Viduka - Fakultet informacionih tehnologija, Alfa BK Univerzitet

\_\_\_\_\_

Student je odbranio doktorsku disertaciju s ocenom \_\_\_\_\_(\_\_\_\_\_)

## **Zahvalnica**

*Želim da izrazim iskrenu i duboku zahvalnost svima koji su tokom mog doktorskog puta bili uz mene, pružali podršku, razumevanje i svojim znanjem, savetima i ohrabrenjem doprineli nastanku ove disertacije. Njihova nesebična pomoć i vera u mene predstavljali su dragocen oslonac tokom svih godina istraživanja i rada.*

*Posebnu zahvalnost dugujem svom mentoru, prof. dr Milanu Gligorijeviću, na izuzetnoj posvećenosti, strpljenju, stručnom vođenju i dragocnim savetima. Njegova podrška, jasne smernice i poverenje bili su od ključnog značaja za oblikovanje i uspešno završavanje ove disertacije.*

*Zahvaljujem dr Ani Matović i Mariji Matović na stručnoj pomoći, korisnim sugestijama i smernicama koje su značajno doprinele kvalitetu ovog rada.*

*Iskrenu zahvalnost dugujem dr Edisu Mekiću na vrednim savetima, inspirativnim idejama i podršci tokom izrade disertacije.*

*Ovu disertaciju posvećujem svojoj porodici – supruzi Anamariji, ćerki Hani i sinu Novaku. Njihova безусловna ljubav, strpljenje, razumevanje i stalna podrška bili su moja najveća snaga i motivacija. Oni su moj oslonac i pokretačka energija na putu ostvarivanja snova i ciljeva, a ova disertacija predstavlja jedan od njihovih zajedničkih uspeha.*

*Mario Stojanović*

## Sadržaj

1. Uvod.....	10
2. Dosadašnja istraživanja.....	11
3. Predmet i problem istraživanja .....	14
4. Cilj i zadaci istraživanja.....	15
5. Hipoteza istraživanja.....	18
6. Metodologija naučnog istraživanja .....	21
7. Implementacija istraživanjaRezultati istraž .....	26
8. Rezultati istraživanja.....	115
9. Interpretacija i diskusija rezultata istraživanja.....	155
10. Značaj istraživanja za teorijsku praksu .....	163
11. Zaključak.....	165
12. Literatura.....	166
13. Biografija autora .....	173

## **Abstract**

This doctoral dissertation explores the role of simulation and artificial intelligence (AI) in the development of computer networks, VoIP systems, and digital signal processing (DSP). Simulation enables the analysis of network performance, modeling of telecommunication protocols, and optimization of signal processing in controlled environments. By utilizing advanced simulation techniques, it allows for a detailed examination of network congestion, latency, and throughput.

In VoIP technology, simulations facilitate the study of key parameters such as delay, jitter, and packet loss, which are crucial for improving service quality and communication security. Furthermore, the integration of AI into DSP has revolutionized signal processing, enhancing speech recognition and image processing. Deep learning techniques and neural networks improve accuracy and adaptability in real-world applications.

Despite challenges in modeling complex systems and high computational demands, this research contributes to the advancement of simulation methods and AI techniques, opening new possibilities for more efficient, scalable, and intelligent network and communication systems.

### **Keywords:**

- VoIP (Voice over Internet Protocol)
- IP Internet protocols
- AI artificial intelligence
- DSP digital signal processing
- latency
- jitter
- packet loss
- Agile
- DevOps
- Action research
- GLMP Good Laboratory and Manufacturing Practice
- Scrum
- PO Product owner
- PB product backlog
- SPrint
- SIP
- RFC 3261
- H.323

- Skinny protocol
- RTP Real-time Transport Protocol
- Image processing
- CNN convolutional neuron networks
- Image classification
- Image segmentation
- Generative neuron networks
- Image generation

## Apstrakt

Ovaj doktorski rad istražuje ulogu simulacije i veštačke inteligencije (AI) u razvoju računarskih mreža, VoIP sistema i digitalne obrade signala (DSP). Simulacija omogućava analizu mrežnih performansi, modelovanje telekomunikacionih protokola i optimizaciju obrade signala u kontrolisanim uslovima. Korišćenjem naprednih simulacionih omogućava se detaljno ispitivanje mrežne zagušenosti, latencije i propusnosti.

U VoIP tehnologiji, simulacije omogućavaju istraživanje ključnih parametara kao što su kašnjenje, džiter i gubitak paketa, što je od suštinske važnosti za poboljšanje kvaliteta usluge i bezbednosti komunikacije. Nadalje, integracija AI u DSP revolucionisala je obradu signala, unapređujući prepoznavanje govora i obradu slike. Tehnike dubokog učenja i neuronskih mreža poboljšavaju preciznost i prilagodljivost u realnim aplikacijama.

Uprkos izazovima u modeliranju složenih sistema i velikim računarskim zahtevima, ovaj rad doprinosi unapređenju simulacionih metoda i AI tehnika, otvarajući nove mogućnosti za efikasnije, skalabilne i inteligentne mrežne i komunikacione sisteme.

### Ključne reči:

- VoIP ( Protokoli za prenos zvuka preko Interneta)
- IP inetrenet protokoli
- AI veštačka inteligencija
- DSP digitalbna obrada signala
- kašnjenje
- džiter
- gubitak paketa
- Agilne metodologije
- Devops
- Akciono istraživanje
- Pravilima dobrog razvoja laboratorija i proizvodnje
- Vlasnik proizvoda
- Lista radova
- Sprint
- SIP protocol
- RFC 3261
- H.323
- Skinny prtokol
- RTP protocol za prenos u realnom vremenu

- Obrada slike
- Obrada slike
- CNN konvolucione neuronske mreže
- Kalsifikacija slike
- Segmentacija slike
- Genreativne neuronske mreže
- Generisanje slike

## 1. UVOD

VoIP (Voice over Internet Protocol) tehnologija predstavlja jedno od ključnih rešenja u modernim telekomunikacijama, koje omogućava prenos govora preko mreža koje koriste internet protokole (IP) uz značajne prednosti u fleksibilnosti, troškovima i kvalitetu usluge. Sa sve većim rastom potražnje za visokokvalitetnim VoIP servisima, azvija se i potreba za razvojem simulacionih modela u kojima možemo precizno analizirati performanse VoIP sistema u različitim mrežnim uslovima.

Cilj ovog istraživanja je analiza i optimizacija performansi VoIP sistema kroz simulaciju u kontrolisanim okruženjima. Istraživanje se fokusira na identifikaciju ključnih faktora koji utiču na kvalitet VoIP komunikacije, uključujući latenciju, jitter, gubitak paketa, izbor kodeka, enkripciju i QoS mehanizme. Pored tehničkih aspekata VoIP simulacije, istraživanje uključuje i primenu agilnih metodologija u razvoju simulacionih modela, kako bi se poboljšala efikasnost planiranja, razvoja i implementacije softverskih simulacija. Posebno se razmatra razvoj simulacija koje primenjuju modern algoritme veštačke inteligencije (AI artificial intelligence), za obradu signala prenetog kroz mreže.

Primenom naprednih simulacionih tehnika i integracijom agilnih pristupa, ovaj rad pruža precizne preporuke za optimizaciju VoIP sistema, omogućavajući poboljšanja u mrežnim podešavanjima, sigurnosti i kvalitetu usluge, što će doprineti razvoju efikasnijih i pouzdanijih VoIP komunikacionih rešenja. Sva ova rešenja su implemntirana u simuliranim sistemiima, i validirana u realnoj implementaciji. Ovime je pokazana opravdanost korišćenja simulacija za modelovanje skupih sistema i razvoja rešenja na jeftinijim simulacijama i njihovom kasnijom primenom u realnom svetu.

## 2. DOSADAŠNJA ISTRAŽIVANJA

Simulacija je postala osnovni alat u proučavanju i razvoju računarskih mreža, VoIP sistema i aplikacija veštačke inteligencije (AI) u oblasti digitalne obrade signala (DSP) [1]. Mogućnost kreiranja kontrolisanih okruženja za modelovanje i analizu ponašanja mreža, telekomunikacionih sistema i metoda obrade signala omogućila je istraživačima da steknu dragocene uvide ne remeteći rad realnih sistema [2]. Tokom godina razvijene su različite simulacione tehnike i alati kako bi se odgovorilo na sve veću složenost i kompleksnost računarskih mreža, što je sa druge strane dovelo do jednostavnijeg procesa optimizovanja performansi, smanjenja troškova i poboljšanja tehnoloških mogućnosti mreža [3]. Ovaj rad istražuje ulogu simulacije u ovim domenima, ključne metodologije koje se koriste, kao i uticaj modernih tehnika za DSP zasnovanih na AI za unapređenje ovih oblasti.

Razvoj alata za simulaciju mreža odigrao je ključnu ulogu u razumevanju velikih mrežnih sistema. Nacionalna fizička laboratorija (NPL) bila je pionir u ranim studijama simulacije radi procene performansi paketno-komutiranih mreža, fokusirajući se na datagrame i mrežnu zagušenost, što je bilo od suštinskog značaja za razvoj velikih komunikacionih mreža. Ove rane simulacije pružile su ključne uvide u paketno-komutirane mreže, koje su kasnije postale osnova za moderne internet komunikacione protokole. Uvođenje alata kao što je QualNet dodatno je unapredilo simulaciju mreža korišćenjem tehnologije paralelne simulacije diskretnih događaja, omogućavajući istraživačima da efikasno modeluju bežične komunikacione protokole. Bežične ad hoc mreže, kao ad hoc mreže za vozila (VANETs) takođe su imale koristi od simulacionih studija mreža, posebno u analizi dinamičkih topologija i ponašanja mreža pod ograničenim uslovima mobilnosti [4]. Simulacije VANET mreža uzimaju u obzir ograničene putne topologije, slabljenje signala koje se javlja prenosom istog preko više različitih putanja, uticaj na signale prepreka pored puta i različite brzine vozila, čime se osigurava visok nivo tačnosti u testiranju komunikacionih protokola za inteligentne transportne sisteme. Simulacioni alati poput OPNET-a, NetSim-a i ns2 omogućili su opsežnu optimizaciju parametara i analizu scenarija „šta ako“ kako bi se predvideo učinak mreže u različitim realnim situacijama [5]. Emulacioni testni okviri razvijeni u institucijama poput Istraživačke laboratorije američke vojske (ARL) i Istraživačke laboratorije mornarice (NRL) dodatno su omogućili validaciju algoritama i aplikacija u realnim mrežnim uslovima [6], premošćavajući jaz između teorijskih modela i praktičnih implementacija [7]. Kontinuirani razvoj alata za simulaciju mreža omogućio je bolju skalabilnost i tačnost u evaluaciji složenih komunikacionih sistema [8,9].

Slično tome, u oblasti glasovne komunikacije preko interneta (VoIP), simulacija je obezbedila ključne metode za procenu performansi mreže u različitim uslovima. VoIP tehnologija omogućava prenos glasa preko IP mreža, nudeći isplativa i fleksibilna komunikaciona rešenja [10].

Međutim, obezbeđivanje visokog kvaliteta prenosa glasa zahteva obimna istraživanja parametara prenosa signala poput kašnjenja, džitera (jitter) i gubitka paketa, koji direktno utiču na korisničko iskustvo [11]. Razni simulacioni alati, uključujući OPNET Modeler, OMNeT++ i OverSim, korišćeni su za modelovanje VoIP sistema. Ovi alati omogućavaju istraživačima analizu uticaja mrežnih uslova na kvalitet glasa i pomažu u optimizaciji strategija za obezbeđenje kvaliteta usluge (QoS) [12]. Procene performansi VoIP-a oslanjaju se na ključne metrike poput latencije, koja meri vreme potrebno da glasovni paketi prođu kroz mrežu, džitera, koji se odnosi na varijacije u vremenu dolaska paketa, i gubitka paketa, koji utiče na kontinuitet zvuka [13]. Povećana mrežna zagušenja, ograničenja propusnog opsega i neefikasnosti rutiranja dodatno otežavaju ove izazove, što zahteva robusne simulacione tehnike za modelovanje scenarija stvarnog saobraćaja. Analizom ovih faktora istraživači su uspeli da dizajniraju adaptivne VoIP protokole koji poboljšavaju korisničko iskustvo i pouzdanost komunikacije [14]. Pored toga, bezbednost VoIP-a postaje sve značajnije pitanje, jer sajber pretnje poput prisluškivanja, DoS napada i glasovnog spama (SPIT) mogu degradirati kvalitet usluge [15]. Simulacije pružaju dragocene uvide u ranjivosti VoIP sistema, omogućavajući razvoj tehnika enkripcije, sistema za detekciju upada i sigurnih protokola [16].

Primena veštačke inteligencije u digitalnoj obradi signala transformisala je različite oblasti, uključujući telekomunikacije, obradu multimedije, biomedicinski inženjering i finansijsku analizu. Integracija AI i DSP značajno je poboljšala kapacitete za obradu podataka, omogućavajući sistemima da se dinamički prilagođavaju složenim signalima i poboljšavaju tačnost u aplikacijama u realnom vremenu [17]. AI tehnike poput dubokog učenja i neuronskih mreža integrisane su u DSP sisteme kako bi poboljšale analizu signala i prepoznavanje obrazaca [18]. U obradi slika i video zapisa, DSP tehnike zasnovane na AI omogućile su napredne funkcionalnosti poput detekcije objekata, prepoznavanja lica i poboljšanja slike, uglavnom kroz modele dubokog učenja poput konvolucionih neuronskih mreža (CNN). Implementacija AI u video obradi dodatno je poboljšala interpolaciju kadrova, poboljšanje super-rezolucije i smanjenje šuma, čineći moderne tehnologije striminga i emitovanja efikasnijima [19].

U biomedicinskom sektoru, DSP tehnike zasnovane na AI metode omogućile su efikasniju analizu fizioloških signala, poput elektroencefalograma (EEG) i elektrokardiograma (ECG), za ranu dijagnostiku neuroloških i kardiovaskularnih poremećaja. AI zasnovani dijagnostički alati koriste DSP algoritme za otkrivanje obrazaca u moždanim talasima radi ranog prepoznavanja epilepsije, Alchajmerove bolesti i drugih kognitivnih poremećaja. Slično tome, AI unapređena ECG analiza omogućila je rano otkrivanje aritmija i srčanih oboljenja, značajno poboljšavajući ishode za pacijente [20].

Uprkos značajnom napretku koji je omogućen kombinacijom simulacije i DSP tehnologijom zasnovanom na AI, i dalje postoje brojni izazovi za rešavanje ovih problema [21].

Složenost modeliranja realnih sistema [22], računarski zahtevi i potreba za visoko preciznim simulacionim okruženjima i dalje predstavljaju prepreke širokoj primeni [23]. Buduća istraživanja moraju se fokusirati na poboljšanje tačnosti simulacija [24], optimizaciju AI algoritama za DSP aplikacije i razvoj hibridnih modela koji kombinuju empirijske podatke sa simulacionim okruženjima [25]. Interdisciplinarna saradnja između računarskih naučnika, mrežnih inženjera i stručnjaka iz različitih domena biće ključna za dalji napredak u integraciji AI, simulacionih tehnika i DSP-a [26]. Rešavanjem ovih izazova, AI, simulacija i DSP nastaviće da unapređuju tehnološki napredak u računarskim mrežama, telekomunikacijama i digitalnoj obradi signala, omogućavajući efikasnije i inteligentnije sisteme komunikacije i obrade informacija.

### 3. PREDMET I PROBLEM ISTRAŽIVANJA

#### Predmet istraživanja

Predmet ovog istraživanja je analiza i optimizacija performansi VoIP sistema kroz simulaciju u kontrolisanim uslovima. Fokus istraživanja je na identifikaciji ključnih faktora koji utiču na kvalitet VoIP komunikacije, uključujući latenciju, jitter, gubitak paketa, izbor kodeka, enkripciju i QoS mehanizme. Dodatno, istraživanje obuhvata primenu agilnih metodologija u razvoju simulacionih modela VoIP sistema, sa ciljem poboljšanja efikasnosti planiranja, razvoja i isporuke softverskih simulacija.

#### Problem istraživanja

Glavni problem istraživanja odnosi se na optimizaciju performansi VoIP sistema u realnim uslovima, bez narušavanja korisničkog iskustva. VoIP komunikacija zavisi od prenosa u realnom vremenu, te je izazov pronaći optimalan balans između kvaliteta poziva, sigurnosti, propusnog opsega i mrežnih resursa. Ključni aspekti problema uključuju:

Mogućnost postizanja realistične simulacije VoIP sistema koja verno odražava performanse u stvarnom okruženju.

Uticaj povećanja broja korisnika na mrežnu zagušenost i kvalitet usluge.

Balans između jasnoće zvuka i efikasnosti resursa u zavisnosti od izbora kodeka i enkripcionih protokola.

Efikasnost QoS mehanizama u poboljšanju stabilnosti i kvaliteta VoIP poziva.

Uticaj cloud okruženja na latenciju i skalabilnost VoIP simulacije.

Uloga agilnih metodologija u unapređenju procesa razvoja i implementacije VoIP simulacionih modela.

Ovim istraživanjem nastoji se pronaći optimalna mrežna podešavanja i razvojne metodologije koje omogućavaju poboljšanje kvaliteta VoIP komunikacije i povećanje efikasnosti njenog razvoja.

#### 4. CILJ I ZADACI ISTRAŽIVANJA

Cilj ovog istraživanja je da se kroz razvoj i testiranje simuliranog VoIP sistema u kontrolisanim uslovima ispita mogućnost optimizacije mrežnih podešavanja, čime bi se postigla niža latencija i viši kvalitet poziva. Istraživanje će omogućiti eksperimentisanje sa parametrima mreže koji u realnim uslovima ne mogu biti menjani bez uticaja na korisničko iskustvo. Nakon validacije optimizovanih podešavanja u simulatoru, ista će moći da budu primenjena u realnoj mreži uz minimalne rizike. Ovim se stvara efikasan metod za unapređenje performansi VoIP sistema kroz detaljnu analizu uticaja različitih faktora na kvalitet usluge.

Ciljevi istraživanja prema postavljenim istraživačkim pitanjima

##### **Verifikacija mogućnosti virtuelnih sistema da simuliraju realne VoIP procese**

Utvrđiti koji su ključni tehnički faktori potrebni za tačnu simulaciju realnih VoIP sistema.

Evaluirati u kojoj meri simulirani sistemi mogu reprodukovati performanse stvarnih mreža u različitim scenarijima.

Identifikovati ograničenja simuliranih VoIP sistema i predložiti načine za njihovo prevazilaženje.

##### **Analiza uticaja povećanja broja istovremenih VoIP poziva na performanse mreže**

Ispitati kako povećanje broja aktivnih poziva utiče na korišćenje propusnog opsega, jitter i gubitak paketa.

Identifikovati kritične tačke mrežnog zagušenja pri velikom broju istovremenih korisnika.

Predložiti metode za minimizaciju negativnih efekata mrežne preopterećenosti na VoIP uslugu.

##### **Ispitivanje uticaja izbora audio kodeka na performanse VoIP sistema**

Uporediti kvalitet zvuka, potrošnju propusnog opsega i opterećenje procesora za različite kodeke (G.711, G.729, Opus).

Analizirati kompromis između jasnoće zvuka i efikasnosti mrežnih resursa u zavisnosti od izbora kodeka.

Odrediti optimalne kodeke za specifične scenarije korišćenja VoIP sistema.

##### **Evaluacija uticaja enkripcionih protokola na sigurnost i performanse VoIP sistema**

Analizirati kako različiti enkripcioni protokoli (SRTP, TLS) utiču na latenciju VoIP komunikacije.

Meriti dodatno opterećenje procesora pri primeni enkripcije u realnim i simuliranim uslovima.

Utvrđiti optimalan balans između bezbednosti i efikasnosti komunikacije u realnom vremenu.

### **Unapređenje stabilnosti VoIP sistema kroz primenu QoS mehanizama**

Ispitati koji QoS mehanizmi imaju najveći uticaj na smanjenje jittera i gubitka paketa. Analizirati ponašanje različitih prioritetnih algoritama u uslovima mrežnog zagušenja. Testirati efikasnost simuliranih QoS algoritama u odnosu na realne mrežne uslove.

### **Procena skalabilnosti VoIP simulacije u cloud okruženju**

Uporediti latenciju VoIP sistema u lokalnim i cloud okruženjima.

Ispitati glavne prednosti i nedostatke implementacije VoIP simulacije u cloud infrastrukturi.

Analizirati mogućnosti optimizacije rutiranja kako bi se smanjila latencija u cloud VoIP sistemima.

Ciljevi istraživanja vezani za primenu agilnih metodologija

Drugi deo istraživanja bavi se analizom primene agilnih metodologija u razvoju simulacionih modela VoIP sistema. Cilj je da se ispita kako fleksibilni pristupi, poput DevOps-a i Scrum-a, utiču na efikasnost razvoja i isporuke softverskih simulacija.

### **Povećanje efikasnosti planiranja i razvoja simulacionih sistema kroz agilne metodologije**

Istražiti kako primena agilnih metoda može poboljšati organizaciju i razvoj simulacionih sistema.

Evaluirati na koji način korišćenje DevOps tehnologija poboljšava efikasnost procesa isporuke gotovih simulacija.

### **Analiza uticaja agilnih metodologija na kvalitet i efikasnost razvoja softvera**

Uporediti efikasnost timova koji koriste agilne metodologije sa onima koji ih ne koriste. Ispitati uticaj agilnih pristupa na kvalitet i stabilnost isporučenih softverskih rešenja.

Analizirati povećanje broja isporučenih zahteva i uspešnost implementacije softverskih komponenti pri upotrebi agilnih metoda.

### **Unapređenje timskog rada kroz iterativne i kolaborativne metodologije**

Ispitati kako integracija Scrum metodologije poboljšava komunikaciju i timsku produktivnost.

Evaluirati značaj iterativnog razvoja i povratnih informacija u agilnom procesu rada.

Utvrđiti kako akciono istraživanje doprinosi fleksibilnosti i kontinuiranom unapređenju softverskog razvoja.

Zaključak

Ovo istraživanje ima za cilj unapređenje performansi VoIP sistema kroz simulaciju realnih mrežnih uslova i testiranje različitih faktora koji utiču na kvalitet poziva. Dodatno, primena

agilnih metodologija u razvoju softverskih simulacija biće analizirana kako bi se utvrdilo u kojoj meri poboljšavaju efikasnost i kvalitet krajnjeg proizvoda. Kombinovanjem ovih pristupa, istraživanje nastoji da pruži preporuke za optimizaciju VoIP sistema i metodologija razvoja simulacionih modela.

## 5. HIPOTEZE ISTRAŽIVANJA

Razvojem VoIP simulatora koji može biti testiran u kontrolisanim uslovima radi optimizovanja mrežnih podešavanjima, očekuje da se primenom simuliranih sistema u realnim uslovima postigne niža latencije i veći kvalitet poziva. Ova hipoteza pretpostavlja da idealno podešavanje mreže u simulaciji i primenom istog u realnim okvirima minimizuje kašnjenje, omogućavajući gotovo trenutni prenos zvuka i jasnu komunikaciju. Osim toga, simulirani sistem omogućava eksperimentisanje sa parametrima veze koji u realnim uslovima ne bi mogli biti menjani bez rizika po stabilnost mreže. Nakon testiranja optimizovanih parametara u simulatoru, iste postavke mogu biti implementirane u realnoj mreži bez ugrožavanja korisničkog iskustva. Postavka ovakve osnovne hipoteze postavlja set idućih istraživačkih pitanja.

1. Virtuelni sistemi mogu u potpunosti simulirati realne procese u VoIP sistemima.
2. Koji su ključni tehnički faktori potrebni za vernu simulaciju realnih VoIP sistema?
3. U kojoj meri simulirani VoIP sistemi reprodukuju performanse stvarnih mreža u različitim scenarijima?
4. Koja su ograničenja simuliranih VoIP sistema u poređenju sa realnim okruženjima?
5. Povećanje broja istovremenih VoIP poziva značajno utiče na korišćenje propusnog opsega, izaziva jitter i dovodi do gubitka paketa.
6. Kako povećanje broja istovremenih poziva utiče na kvalitet usluge u simuliranoj VoIP mreži?
7. Koje su kritične tačke mrežnog zagušenja pri povećanom broju VoIP korisnika?
8. Na koji način se jitter i gubitak paketa mogu minimizirati u uslovima visoke mrežne opterećenosti?
9. Izbor audio kodeka (npr. G.711, G.729, Opus) utiče na kvalitet poziva, potrošnju propusnog opsega i opterećenje procesora.
10. Kako različiti audio kodeci utiču na balans između kvaliteta zvuka i potrošnje propusnog opsega?
11. Koji su kompromisi između jasnoće zvuka i mrežnih resursa pri upotrebi različitih kodeka?
12. Kako izbor kodeka utiče na performanse procesora u VoIP komunikaciji?
13. Implementacija enkripcionih protokola (npr. SRTP, TLS) poboljšava sigurnost VoIP komunikacije, ali može povećati latenciju i opterećenje procesora.
14. Kako različiti enkripcioni protokoli utiču na latenciju VoIP komunikacije?

15. Koliko dodatnog opterećenja enkripcija stvara na procesorskim resursima u realnim i simuliranim uslovima?
16. Na koji način se može postići optimalan balans između sigurnosti i performansi VoIP sistema?
17. Omogućavanje QoS mehanizama u simuliranoj VoIP mreži smanjuje gubitak paketa i jitter, poboljšavajući stabilnost poziva.
18. Koji QoS mehanizmi imaju najveći uticaj na poboljšanje kvaliteta VoIP poziva?
19. Kako se različiti prioritetni algoritmi unutar QoS mehanizama ponašaju u uslovima mrežnog zagušenja?
20. Koliko efikasno simulirani QoS algoritmi mogu predvideti i rešiti probleme u realnim VoIP mrežama?
21. Pokretanje VoIP simulatora u cloud okruženju omogućava bolju skalabilnost, ali može povećati latenciju zbog dodatnih mrežnih skokova i neefikasnosti internet rutiranja.
22. Kako se latencija VoIP sistema menja pri migraciji sa lokalnog na cloud okruženje?
23. Koje su glavne prednosti i nedostaci pokretanja VoIP simulacije u cloud infrastrukturi?
24. Da li je moguće optimizovati rutiranje kako bi se smanjila latencija u cloud VoIP sistemima?

Kako bismo dokazali ove pretpostavke i dali odgovore na set postavljenih pitanja, testiraćemo sledeće pojedinačne hipoteze:

#### Hipoteze vezane za VoIP simulaciju

1. Virtuelni sistemi mogu u potpunosti simulirati realne procese u VoIP sistemima.
2. Povećanje broja istovremenih VoIP poziva značajno utiče na korišćenje propusnog opsega, izaziva jitter i dovodi do gubitka paketa. Pošto VoIP zavisi od prenosa u realnom vremenu, očekuje se da mrežno zagušenje degradira kvalitet usluge.
3. Izbor audio kodeka (npr. G.711, G.729, Opus) utiče na kvalitet poziva, potrošnju propusnog opsega i opterećenje procesora. Kodeci sa većom kompresijom troše manje propusnog opsega, ali mogu kompromitovati jasnoću zvuka.
4. Implementacija enkripcionih protokola (npr. SRTP, TLS) poboljšava sigurnost VoIP komunikacije, ali može povećati latenciju i opterećenje procesora. Fokus istraživanja je na kompromisu između bezbednosti i efikasnosti komunikacije u realnom vremenu.

5. Omogućavanje QoS mehanizama u simuliranoj VoIP mreži smanjuje gubitak paketa i jitter, poboljšavajući stabilnost poziva. QoS daje prioritet VoIP paketima u odnosu na ostali saobraćaj, osiguravajući neometanu komunikaciju.
6. Pokretanje VoIP simulatora u cloud okruženju omogućava bolju skalabilnost, ali može povećati latenciju zbog dodatnih mrežnih skokova i neefikasnosti internet rutiranja.

#### Primena agilnih metodologija u razvoju simulacionih modela

Drugi deo istraživanja odnosi se na primenu agilnih metodologija u procesu razvoja simulacionih modela za obradu digitalnih signala DSP. Agilne metodologije u razvoju softverskih proizvoda imaju za cilj poboljšanje efikasnosti razvoja kroz integraciju programskih alata koji poboljšavaju prilagodljivost tima na promene i olakšavaju međusobnu komunikaciju.

#### Istraživačka pitanja

1. Kako se primena agilnih metoda može koristiti za povećanje efikasnosti u planiranju i razvoju simulacionih sistema?
2. Kako upotreba DevOps tehnologija i drugih alata povezanih sa agilnim metodologijama može doprineti poboljšanju efikasnosti procesa isporuke gotovih simulacija?

#### Hipoteze vezane za agilne metodologije

1. Efikasnost projektnog tima veća je kada se u pripremnoj fazi razvoja simulacionih sistema koriste agilne metodologije i DevOps.
2. Efikasnost i kvalitet isporučenog softverskog proizvoda viši su kada se koriste agilne metodologije.
3. Nivo isporučenih zahteva i procenat uspešno razvijenih softverskih komponenti znatno je veći kada se koriste agilne metodologije.
4. Integracija agilnih metodologija, kao što je Scrum, poboljšava kvalitet timskog rada kroz iterativnu, kolaborativnu i fleksibilnu primenu
5. Akciono istraživanje pruža fleksibilan i iterativan okvir kompatibilan sa agilnim metodama, omogućavajući kontinuirano unapređenje postupka razvoja softverskih proizvoda kroz povratne informacije učesnika i usklađivanje sa industrijskim standardima

Istraživanjem simuliranog VoIP sistema i primenom agilnih metodologija u njegovom razvoju, očekuje se dokazivanje postavljenih hipoteza, pri čemu će eksperimentalna evaluacija dati uvid u optimizaciju mrežnih podešavanja, uticaj različitih faktora na kvalitet VoIP poziva i značaj agilnih pristupa u razvoju složenih simulacionih modela.

## 6. METODOLOGIJA NAUČNOG ISTRAŽIVANJA

Tradicionalne metode analize i proučavanja sistema i problema, kao što su strukturisana analiza i analiza podataka, fokusiraju se prvenstveno na „tvrde“ aspekte problema koje proučavamo – one koji su sigurni i precizni. Ovi rigidni pristupi analizi problema prate strogo definisani model i mogu se relativno ujednačeno primeniti u različitim modelima. Međutim, istraživači takođe moraju da uzmu u obzir dvosmisleni, složenu prirodu svakog modela izrade softvera, gde međusobno povezani uticaji na dizajn i implementaciju softvera i simulacije mogu biti izazovni za snalženje. Složenost ovih procesa proizilazi iz ljudskog faktora, pošto se ljudi značajno razlikuju od podataka i procesa, koji su bez ovog faktora jednostavniji za analizu. Pojedinci imaju različite i ponekad suprotstavljene ciljeve, percepcije i stavove, koji se vremenom razvijaju. Zanemarivanje ovih ljudskih elemenata može dovesti do neadekvatnih rezultata pri korišćenju tradicionalnih metodologija analize informacionih sistema, jer one često ne odražavaju realnost stvarnih organizacija, odnosno ne uzimaju u obzir ljudski faktor.

Akciono istraživanje (Action research - AR) premošćuje teoriju i praksu tako što uključuje i u analizu problema i uticaj istraživača i inženjera u rešavanju izazova u stvarnom svetu kroz promenu i refleksiju međuljudskog faktora [27]. Ovaj iterativni pristup sastoji se od nekoliko bitnih ciklusa: dijagnostikovanja problema, implementacije intervencija i refleksivne analize problema. Upravo je to razlog zašto softversko inženjerstvo i nauka koja se bavi informacionim tehnologijama takođe često primenjuju ovakav metodološki pristup, pošto se bave problemima koji nastaju u stvarnim organizacionim kontekstima [28].

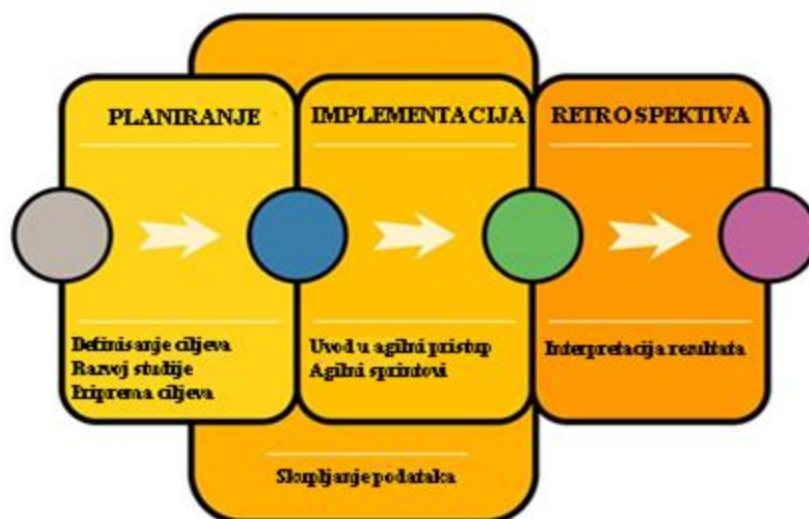
U akcionom istraživanju, istraživači saraduju sa inženjerima u stvarnim okruženjima kako bi testirali teorije, prikupili povratne informacije iz praktičnog iskustva, usavršili teoriju na osnovu ovih inputa i ponovo je primenili. Svaka iteracija ciklusa akcionog istraživanja poboljšava teoriju, čineći je prilagodljivijom za različite scenarije. U slučaju razvoja informacionih sistema, ovaj proces doprinosi stvaranju relevantnijeg i efektivnijeg okvira [29].

Akciono istraživanje je participativna i iterativna metoda koja se fokusira na rešavanje praktičnih problema kroz metodološki okvir koji se sastoji od sledećih koraka: ciklusa planiranja, implementacije, posmatranja i refleksije [30].

Prvi korak akcije uključuje identifikaciju praktičnog problema u kontekstu neophodnom za njegovu realizaciju i zajedničko definisanje njegovog obima. Ova faza zahteva angažovanje svih učesnika (istraživača i drugih zainteresovanih strana) kako bi se obezbedilo međusobno razumevanje i saglasnost. Na primer, problem se ne samo identifikuje već se i duboko analizira kako bi se napravio plan koji može da se primeni [31]. Okvir akcionog istraživanja naglašava inkluzivno učešće u definisanju ciljeva i prikupljanju osnovnih podataka za procenu trenutnog stanja [32].

U drugoj fazi se izvode planirane intervencije ili radnje. Ovaj korak karakteriše aktivno učešće i saradnju svih zainteresovanih strana. Ova faza uključuje strukturirane aktivnosti sa jasno definisanim ulogama učesnika i istraživača. Praćenje u realnom vremenu je od ključnog značaja za dokumentovanje promena, a prilagođavanja akcionog plana mogu biti izvršena ako je potrebno da bi se rešila nova pitanja. Iterativni karakter ove faze obezbeđuje da intervencije ostanu dinamične i da odgovaraju na potrebe identifikovane tokom planiranja [33].

Završna faza se fokusira na posmatranje i evaluaciju ishoda sprovedenih akcija. Učesnici se bave refleksivnim praksama kako bi procenili efikasnost intervencija i njihov uticaj na problem. Metodologija akcionog istraživanja, uključuje i refleksiju zajednice i samoevaluaciju kako bi se osiguralo da istraživanje ostane validno i inkluzivno. Refleksija često vodi ka identifikaciji novih pitanja ili oblasti za dalja istraživanja, čime se pokreće još jedan ciklus akcionog istraživanja [34].



Slika 1 Faze akcionog istraživanja

Ovaj metodološki pristup smo modifikovali i primenili prvo na opšti okvir agilnog pristupa razvoja simulacije, koji će biti u skladu sa pravilima dobrog razvoja laboratorija i proizvodnje (Good Laboratory and Manufacturing Practice (GLMP)). Njegovom primenom ćemo pratiti razvoj simulacije VoIP sistema, a zatim ćemo ovu metodologiju porimeniti na analizu jednog od najčešće korišćenih pristupa razvoja proizvoda takozvanom Scrum pristupu.

Sve metodologije zasnovane na Agile pristupu, kao što su Scrum, Kanban, XP i ostale [35], sadrže zajedničku osnovu.

Naime, svi se odnose na implementaciju i prilagođavanje ciklusa razvoja proizvoda. Glavna ideja je da se projekat podeli na manje delove ili inkremente, koji se dalje dele na manje

iteracije za implementaciju. Na kraju svake iteracije, koja se takođe naziva sprint, dobije se nova verzija proizvoda koja se naziva inkrement.

Svaka od gore pomenutih metodologija se sastoji od sledećih faza: planiranje, dizajn, razvoj, implementacija i pregled što je dato na Slici 2. Faza pregleda takođe funkcioniše istovremeno kao uvod u planiranje sledeće iteracije, jer je to faza iz koje proizilazi ceo iteracijski ciklus [36].

Na Slici 1 je dat pregled celog ciklusa iteracije. Svaki Agilni proces razvoja softvera počinje sa fazom planiranja. U ovoj fazi, vođa Agilnog tima ili Product owner (PO) komunicira sa zainteresovanim stranama, poslovnim timom, programerima i budućim korisnici aplikacija. On definiše viziju projekta koja u potpunosti dobija oblik. Njegova uloga je da se utvrdi ono što projekat ima za cilj da postigne, da dokumentuje poslovne i korisničke zahteve, prioritet izvršavanja zadataka i efikasno alokira resurse.

Dizajn je faza od velikog značaja za sve metodologije zasnovane na Agilnom pristupu, koji prethodi razvoju. U ovom trenutku, tim se uglavnom fokusira na skiciranju arhitekture, korisničkog interfejsa i korisničkog iskustva softverskog proizvoda. Odluke o dizajnu su postavljene kao korisničke priče koje predstavljaju prioritete zahteve definisane u prethodnoj fazi planiranja. Reagovanje na promene potrebe i zahteve, i sve ukupnu laku prilagodljivost je glavna karakteristika faze projektovanja u svim Agilnim metodologijama. Upravo ovo čini razvoj efikasnim da bi dobili pravilnu strukturu softvera i efikasan process razvoja istog u skladu sa vizijom projekta. Pošto je po prirodi iterativna i kroz kontinuirano dobijanje povratnih informacija podložna brzim promenama, faza dizajna omogućava fleksibilnost kako promene nastaju tokom procesa razvoja.

Faza razvoja dolazi nakon dizajna, i u ovoj fazi timovi nastavljaju da rade na trenutnoj iteraciji softvera. Ova faza uključuje sve programerske aktivnosti od UKS/UI dizajna i arhitekture do kodiranja. Po završetku procesa razvoja softvera, Agilni tim instalira aplikaciju i to je prilika za prikupljanje vrednih povratnih informacija koje omogućavaju dalja poboljšanja. Ova iterativna povratna sprega informiše tim kako da razvije iduće verzije. Te verzije su doradene i poboljšane verzije softvera u prethodnim iteracijama.



Slika 2 Struktura Agilne metodologije za single sprint

Nakon razvoja simulacionog modela za prenos slike i zvuka bilo je neophodno razviti sistem za obradu digitalnog signala. Sam sistem je takođe razvijen primenom agilne metodologije,

samo što smo u ovom slučaju sa obzirom na to da smo želeli da isprogramiramo simulacioni sistem koristili modifikovani Scrum pristup izradi sistema koji podržavaju obradu slike korišćenjem najnovih tehnologija veštačke inteligencija.

Tokom istraživanja implementiran je Scrum pristup jer većina kompanija u implementaciji koristi ovu agilnu metodologiju. Scrum je implementacija okvira za upravljanje projektima, prilagođena za iterativni proces razvoja softvera. Pošto je ovaj pristup relativno jednostavan i fokusiran na upravljanje softverom, a ne na tehnički razvoj softvera, može se koristiti kao okosnica za izradu simulacionog sistema [37].

Scrum, kao metodologija, ima tri faze: početnu, razvojnu i završnu. U početnoj fazi zacrtani su opšti ciljevi planiranog zadatka. Takođe u njoj se definišu članove projektnih timova, kao i potrebni alati i resursi. Rezultati ove faze se nazivaju iteracijama. U ovoj fazi se definiše zaostatak proizvoda za dokumentovanje kao korisnička priča. Korisničke priče su zasnovane na jednostavnom pseudokodu:

<Korisnik> želi da <ima> tako da <Benefit>.

Zahteve zatim analizira i daje im prioritet PO. Dokument u kome se ovo arhivira naziva se Product backlog (PB). Ovaj dokument koji se redovno ažurira.

Druga faza Scrum-a se sastoji od sprinteva. Ova faza uključuje seriju ciklusa, svaki od 4 nedelje, svaki sprint daje jedan proizvod na osnovu zadatka. Svaki sprint uključuje sve faze razvoja softvera, od analize zahteva, preko dizajna, do završne faze isporuke nakon pregleda.

Sprint počinje početnim sastankom između PO i članova. Na ovom sastanku se definišu korisničke priče, a zatim počinje sprint. Tokom sprinta, članovi tima imaju 15-minutne Scrum sastanke dnevno. Nakon sprinta, sledi sastanak za pregled sprinta i retrospektiva sprinta [38].

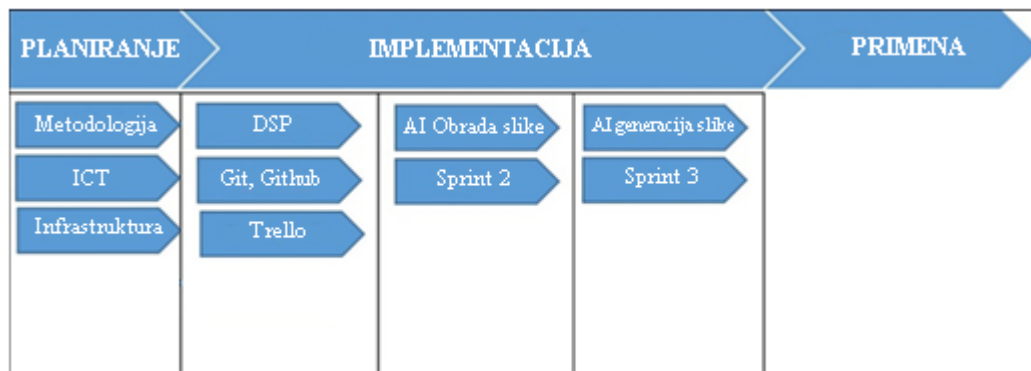
Kada se ispune zahtevi i isporuči softver u skladu sa potrebama vlasnika proizvoda, najnovija verzija proizvoda je spremna za pregled od istog.

Ukupan ciklus Scrum metodologije dat je na slici 3.



Slika 3 Faze Scrum metodologije

Na osnovu ovoga vidimo da je i Scrum metodologija u skladu sa opštim okvirom implementacije akcionog istraživanja, tako da je mesto Scrum metodologije unutar metodologije akcionog istraživanja dato na slici 4.



Slika 4 Opšti okvir implementacije agilne metodologije simulacionog razvoja

## 7. IMPLEMENTACIJA ISTRAŽIVANJA

### 7.1 Razvoj VoIP sistema u skladu sa pravilima dobrog razvoja laboratorija i proizvodnje

Tokom izrade projekta podelili smo zadatke četveročlanim timovima. Članovi tima se na početku upoznaju sa važnim teorijskim osnovama, protokolima i procedurama u projektovanju VoIP sistema; kao i različitim pristupima kako pripremiti simulaciju sistema. Osnovna ideja je kreiranje kompletnog izvodljivog realnog sistema na jednostavan i funkcionalan način, umesto razvoja sistema sa složenom funkcionalnošću koja verovatno ne može biti implementirana tokom ograničenog vremena trajanja istraživanja (otprilike 15 nedelja). Kao krajnji rezultat projekta, članovi istraživačkih timova trebali su da budu osposobljeni da koriste simulaciju i sistem razvijen u realnom vremenu, da analiziraju detalje protokola koji se koriste u VoIP komunikacionim sistemima kako bi se dublje upoznali sa organizacijom i mehanizmima rada VoIP protokola i sistema.

Pre početka razvoja članovi timova su upoznati sa osnovama GLP/GMP pravila. GMP propisi su razvijeni kako bi se osiguralo da se dobiju proizvodi u skladu sa prethodno definisanim standardima kvaliteta [39]. S druge strane, pored ovoga oni moraju da obezbede validnost i pouzdanost studija sprovedenih u laboratorijama i da obezbede kvantifikovano merenje performansi laboratorija u kojima su primenjeni GLP propisi. Prvobitno, GMP/GLP regulativa je razvijena za farmaceutske istraživačke laboratorije, ali njihova opšta priroda omogućava primenu u svim naučnim disciplinama [40].

Prvi korak je bio rešavanje organizaciona pitanja, a u skladu sa GLP/GMP, odnosno uspostavljaju se ključni upravljački uslovi za razvoj simulacije. Ovi uslovi će u ovom slučaju biti: održavanje Direktora studije (SD) (koji će biti odgovoran za ukupan proces i pregled svih aspekata projekta), uspostavljanje jedinica za osiguranje kvaliteta (QAU) (koja će nezavisno vršiti reviziju vođenja procesa), razvoj i usvajanje Standardne operativne procedure (SOP) (u skladu sa dogovorenim i pisanim politikama, rad opreme, administracija), odabir osoblja (biće organizovano, obuka ili obuka definisana u skladu sa njihovim obrazovanjem i iskustvom) [41,42].

Uloga direktora studije je da bude odgovoran za sveukupno sprovođenje svih aspekata projekta, od početka projekta do završene instalacije virtuelnog i realnog vremena VoIP sistema. On je za početak pripremio organizacionu šemu sa jasno definisanim ulogama i odgovornosti tokom planiranog procesa, kao i kanale komunikacije i izveštavanja između različitih subjekata uključenih u proces. Nakon što je organizaciona šema postavljena, SD je pripremio glavni raspored, u kojem je jasno naznačio početne i očekivane datume završetka za definisane delove projekta, radno opterećenje, članove tima, očekivane rezultate za svaku fazu projekta i prekretnice.

Jedinica za osiguranje kvaliteta imala je kritičnu ulogu u implementaciji GLP/GMP, ova jedinica je osigurala da su sve strane uključene u svim fazama projekta kao i da su njihovi dobijeni rezultati u skladu sa GLP/GMP. Osnovni preduslov za kvalitetan rad, je da QAU ne sme biti uključen u razvoj i implementaciju projekta, da bi imao nezavistan pregled procesa realizacije. U klasičnom GLP/GMP pristupu, QAU pokriva samo pitanja upravljanja i procedura, ali u specifičnom pristupu kao što je instalacija VoIP sistema, dodaju joj se dodatne obaveze koje pokrivaju kvalitet i standard opreme i softvera.

Nakon postavljanja QAU -a, SD je definisao Standardne operativne procedure (SOP), definisao kako bi se specifični zadaci trebali implementirati i sprovoditi. Pošto je definisano kreiranje virtuelnog modela i, kasnije, implementacija na odabranom hardversku. U sklopu SOP-a su definisan načini na koji timovi tokom procesa uvek treba da dokumentuju i/ili prate implementaciju. U našem slučaju on je imao sledeću strukturu: naziv i opis korišćene softverske aplikacije, potpuna tehnička specifikacija hardvera na kom je softver instaliran, operativni sistem i preduslovi ako ih ima za instalaciju softvera, glavne funkcionalnosti koje softver pruža, konfiguracija i komunikacione veze između aplikacije i ručne procedure za instaliranje hardvera i ispravnu instalaciju softvera. Cilj SOP-a je da novo razvijena implementirana rešenja treba da budu propisno evidentirana, kako bi ih drugi korisnici lako replicirali. Na osnovu ovog pregleda, QAU treba da bude u stanju da instalira i validira razvijena rešenja.

### **7.1.1 Faza planiranja**

#### **Softverski paketi za implementaciju**

Pitanja složenosti realnih komunikacionih i informacionih sistema postavljaju ograničenje za sprovođenje inovativnih i eksperimentalnih laboratorijskih sistema. Virtuelne laboratorije i simulacije pružaju rešenje za njih probleme. Razlog za predlaganje VirtualVoIP-a za razvoj je da se inženjerima pruži uvid u rad kompleksnog sistema u jednostavnom rešenju sa niskom cenom, a koji će sadržati pun skup funkcionalnosti skupljeg realnog sistema [43]. U ovoj fazi članovi istraživačkog tima su dobili projekat i odlučili su se koji softverski alati mogu da se koriste za odgovarajuću simulaciju sistema. Ceo proces upravljanja timom kao i odluka o tome koji aspekt projekta treba da pokrije je ostavljena njima samima. Pošto je važan aspekt GLP/GMP standarda je obučenosť i znanje kadra koji bi obavljaju ove aktivnosti, SD je podelio sve istraživače u četiri tima, i upoznao ih sa SOP.

Pre predlaganja softverskih i hardverskih rešenja za dati problem koji je definisao SD, delegirali smo Tim\_A za dizajn virtuelnog i realnog VoIP sistem što je u skladu sa DK fazom GLP/GMP. Ova faza kao rezultat daje svu neophodnu opremu u skladu sa zahtevima GLP/GMP. Oni su definisali operativne i funkcionalne specifikacije softvera i hardvera potrebnog za uspešnu

implementaciju projekta. Sva predložena rešenja morala su da imaju potrebne funkcije i performanse potrebne da implementirati VoIP sistem. Predlog je uključio analizu problema, postavljanje topologije mrežnog okruženja, definisanje protokola o softverskoj i hardverskoj podršci, opis korišćenja softvera, izbor paketa na osnovu o njihove funkcionalne i operativne specifikacije, kao i opise aktuelnih operativnih sistema.

Za instalaciju virtuelnih mašina predložena je VMware Workstation Ver.10.0.2, pošto na ovoj verziji postoji mogućnost instaliranja do sto virtuelnih mašina po jednom hostu. Operativni sistem Serverske virtuelne mašine Centos 7 OS sa IP adresom 192.168.10.2, a na klijentskim mašinama (User\_1 do User\_7) je instaliran Lubuntu 14.10 OS, sa IP adresama 192.168.2.2 do 192.168.8.2. Pošto su softverski telefoni koji imaju male zahteve za resursima, a Lubuntu kao lagana distribucija Linuka, ne prekoračavaju resurse ponuđene za VM.

Nakon analize dostupnih softverskih rešenja, dat je predlog da simulacija VoIP sistema bude zasnovana na: GNS3+ mrežnom simulatoru, Asterisk PBKS, VMware i Wireshark alatom za kontrolu i praćenje saobraćaja. Na realnim i virtuelnim klijentskim mašinama, određeni instaliran je Lubuntu i GSKS softphone.

Jedna od glavnih funkcionalnosti GNS3+ paketa je mogućnost kreiranja virtuelnih mreža učitavanjem ISO fajlova postojećih mašina za rutiranje, tako da možemo da kreiramo virtuelne mrežne laboratorije u okruženjima gde nedostaje skupa oprema.

Asterisk je open source softver koji računarski hardver bilo kog računara stavlja u funkciju komunikacionog servera. Naziv je dobio po asterisk (\*) simbolu, a PBX predstavlja skraćenicu od Private Branch eXchange. Inicijalna verzija ovog softvera pojavila se 1999. godine. Tada je tvorac Asteriska, sada već dobro poznati Mark Spenser (engl. Mark Spencer) iz kompanije Digium, objavio inicijalni kod pod GPL licencom. Od tada je u njegov razvoj uključeno mnogo programera širom sveta, preciznije rečeno, čitava Asterisk komuna. Softver je napisan u C programskom jeziku i trenutno je aktuelna verzija 12. Iako je prvobitno dizajniran za linux operativni sistem, Asterisk takođe može da radi i pod NetBSD, OpenBSD, FreeBSD, MAC OS X, i Solaris [44].

Na internetu možete naići na podatak da sistemi zasnovani na Asterisku čine 18% od ukupnog svetskog komunikacionog tržišta. Ono što je Asterisk učinilo liderom je činjenica da su pre njega telefonski sistemi bili poprilično skupi i predstavljali su velike investicije za preduzeća. Promena koju je Asterisk doneo kada se pojavio, oslikana je kroz podatak da je jedini preduslov za njegovo pokretanje da posedujete računar sa linux operativnim sistemom. Zavisno od namene budućeg sistema moguće da će biti neophodan specijalizovani hardver, što je u praksi najčešće neka kartica PCI standarda koja linux sistemima proširuje operativnost i na oblast telefonije. Tu dolazimo do firme Digium, koja danas predstavlja glavnog sponzora Asterisk projekta i koja prodaje i razvija uređaje koji su podržani od strane ovog softvera. Digium takođe obezbeđuje

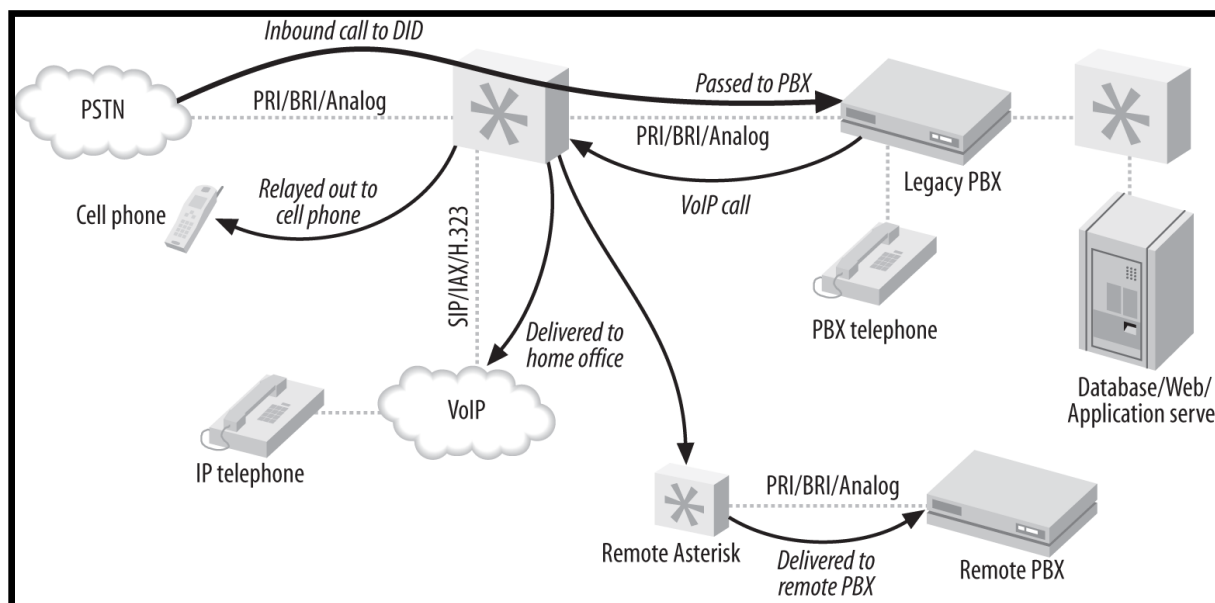
treninge i podršku za Asterisk sisteme ili gotova rešenja zasnovana na njima. Podržano je mnoštvo VoIP protokola od kojih su najpoznatiji SIP, IAX2 i H.323 protokoli, IAX protokol razvio je Mark Spenser za potrebe Asterisk sistema (IAX2 je verzija 2 IAX protokola). Padi sa svim SIP telefonima, pri čemu predstavlja registar ili vezu između IP telefona i PSTN. Pored toga Asterisk podržava i kombinaciju klasične i VoIP telefonije, što je osnovni preduslov za nadgradnju postojećih sistema i prelazak na nove tehnologije. Asterisk se koristi kao zamena za klasične PBX ili za njihovu nadogradnju. Bitno je napomenuti da ukoliko Asterisk sistem planirate da koristiti samo kao VoIP gateway, sa namenom da obezbedi isključivo međusobnu komunikaciju VoIP klijenata, nikakav dodatni hardver vam nije neophodan.

Call centar ili kancelarijski telefonski sistemi su najčešći primeri upotrebe ovog proizvoda. Asterisk korisnicima potpuno besplatno omogućuje sledeće funkcionalnosti: glasovnu poštu, direktno prosleđivanje poziva (engl. automated attendant), pozive na čekanju, snimanje poziva, email integraciju, konferencijske razgovore, a pri tome je vrlo fleksibilan i možemo ga modelovati prema našim potrebama. Skalabilan je i ukoliko se ukaže potreba može biti operativan i na više hardverskih instanci, a podržava rad i pod virtualnim okruženjem. Tipičan server sa Asteriskom sposoban je da podnese stotine konkurentnih poziva i hiljade korisnika [45].

Asterisk možete preuzeti sa adrese [www.asterisk.org/downloads](http://www.asterisk.org/downloads). Moguće je preuzeti fajl u tar.gz formatu i naknadno ga kompajlirati. Za neke Linux distribucije već postoje predefinisani paketi, međutim baš zbog toga što proces instalacije ume da bude komplikovan i pošto ume da se razlikuje od Linux distribucije do distribucije, sa date adrese moguće je preuzeti AsteriskNOW. Kao praktično i upotrebljivo olakšanje posla za korisnike, AsteriskNOW predstavlja ISO imidž već predefinisane Linux distribucije sa podešenim Asteriskom i instaliranim Web admin interfejsom (FreePBX GUI). Može se besplatno preuzeti za 32-bitnu i za 64-bitnu platformu. Instalater će vam postaviti par jednostavnih pitanja, a sva naknadna Asterisk podešavanja se vrše pomoću Web admin interfejsa.

Ukoliko se odlučite za instalaciju na već postojeću Linux distribuciju, sam Asterisk podrazumevano dolazi bez grafičkog interfejsa, te se sva podešavanja vrše konzolno. Konfiguracija se sastoji iz skupa konfiguracionih fajlova, koje je potrebno posle instalacije podesiti prema specifičnim potrebama, te stoga možemo reći da i proces instalacije, a posebno konfiguracije zahtevaju visok stepen poznavanja što Linux-a, kao i samog Asteriska . Više reči o instalaciji i podešavanju Asteriska biće u posebnom poglavlju.

Asterisk može da se koristi kao veoma uspešna veza između starog PBX i novijih verzija. Može se postaviti ispred PBX kao kapija (i koji dozvoljava korisnicima da migriraju van PBX u zavisnosti od toga kako potrebe nalažu), ili se može postaviti iza PBX kao periferni aplikativni server. Čak može oboje u isto vreme, kao što je prikazano na slici 5.



Slika 5 Asterisk kao PBX mrežni prolaz

Sve ono što nam pruža Internet dovelo nas je da ga jednostavno zavolimo, kako zbog toga što je toliko bogat sadržajem i, što je možda još važnije, jer nam omogućava da definišemo kako komuniciramo. Znajući da razvoj i napredovanje Internet tehnologija nisu dostigli svoj maksimum, to će nas „naterati“ da ga koristimo sve više i više. Prenošenje glasa Internetom je sve više zastupljeno i ovu mogućnost koristi značajna populacija u Svetu što, imajući u vidu da kvalitet glasa ne odstupa od kvaliteta u PSTN, može dovesti do minimalne zastupljenosti PSTN u celokupnom procentu prenosa glasa ili čak i do potpunog gašenja ove tehnologije. Naravno, teško je zamisliti da se PSTN ugasi, u pravom smislu te reči, kako zbog svoje dobro razvijene infrastrukture tako i zbog prednosti koje poseduje, već će to gašenje nastati tako što će PSTN tehnologija postati sastavni deo Interneta kao još jedna od njegovih tehnologija. Kao i kod većine ostatka Interneta, open source tehnologije će dovesti do ove transformacije .

Sve ono što je generalno potrebno Internet tehnologijama iz oblasti obrade i prenosa govora, tiče se i razvoja i unapređenja Asteriska.

VMware Workstation je softver koji pruža mogućnost korisniku/korisnicima da kreiraju neograničen broj virtuelnih mašina na računarima sa x86 procesorima. Svaka virtuelna mašina može imati svoj „gostujući“ operativni sistem kao što su windows, linux i druge slične varijante. Jednostavno govoreći, VMware Workstation dopušta jednoj fizičkoj mašini da pokrene neograničeno mnogo operativnih sistema istovremeno. Ostali VMware proizvodi pomažu rukovođenjem (VMware Player) i prenošenjem (VMware Server) sa određenim prednostima ali i manama u odnosu na VMware Workstation.

Osim premošćavanja mrežnih adaptera, CD-ROM čitača, HDD diskova i USB uređaja, VMware Workstation dopušta i simulacije sa hardverom. Na primer, moguće je kreirati ISO fajl za CD-ROM i .vmdk fajlove za HDD; mrežni adapter drajver se može konfigurirati da koristi mrežnu adresu sa prolazom kroz host računar dosta efikasnije nego kada bi koristio premošćenu varijantu (koja bi zahtevalo IP adresu za svaki gostujući operativni sistem na host mreži)

VMware Workstation Pro korišćen u istraživanju (poznat kao VMware Workstation do izdavanja VMware Workstation 12 2015.) je hipervizor sa hostom (Tip 2) koji radi na k64 verzijama Windows i Linux operativnih sistema. On omogućava korisnicima da podese virtuelne mašine (VM) na jednoj fizičkoj mašini i koriste ih istovremeno zajedno sa glavnom mašinom na kojoj je instaliran ovaj programski paket. Svaka virtuelna mašina može da izvršava sopstveni operativni sistem nezavisno od drugih virtuelnih mašina, uključujući sopstvene verzije Microsoft Windows, Linux, BSD ili MS-DOS. VMware Workstation razvija i prodaje VMware, koji je u vlasništvu Broadcoma od novembra 2023. U maju 2024. Workstation Pro je postao besplatan za ličnu upotrebu, sa plaćenim pretplatama dostupnim za komercijalnu upotrebu, dok je besplatni ograničeni VMware Workstation Player (poznat kao VMware Player) odbačen. U novembru 2024. VMware Workstation je postao besplatna za komercijalnu upotrebu, a plaćene pretplate i podrška više nisu postojeće opcije.

VMware Workstation omogućava premošćavanje postojećih mrežnih adaptera domaćina i deljenje fizičkih diskova i USB uređaja sa virtuelnom mašinom. Može da simulira disk jedinice; ISO image datoteka se može mountovati kao virtuelni optički disk, a virtuelni hard diskovi se implementiraju kao \*.vmdk datoteke.

VMware Workstation Pro može da sačuva stanje virtuelne mašine („snimak“) u bilo kom trenutku. Ovi snimci se kasnije mogu vratiti, efektivno vraćajući virtuelnu mašinu u sačuvano stanje, kakvo je bilo i bez bilo kakvog oštećenja VM-a nakon snimanja.

VMware Workstation uključuje mogućnost grupisanja više virtuelnih mašina u fasciklu inventara. Mašine u takvoj fascikli se zatim mogu uključiti i isključiti kao jedan objekat, koristan za testiranje složenih klijent-server okruženja [46].

VMware Tools, paket sa drajverima i drugim softverom koji je dostupan za različite operativne sisteme koje VMware proizvodi podržavaju, instaliraju se na instalirane operativne sisteme radi dodavanja novi funkcionalnosti. Alati se ažuriraju s vremena na vreme, sa imaju sledeće bitne elemente neophodne za implementaciju istraživanja: Drajveri za emulirani hardver; VESA kompatibilna grafika za virtuelnu mašinu za pristup visokim rezolucijama ekrana i/ili specijalnim efektima prozora kao što je Windows Aero/Desktop Windows Manager; Mrežni drajveri za vmknet2 i vmknet3 NIC; Ensoniq AudioPCI audio; Integracija miša; Podrška za deljene fascikle i prevuci i ispusti prenos datoteka između virtuelne mašine i mašine na kojoj je instaliran paket. (Ova funkcionalnost je opisana kao HGFS (Host Guest File System) i može biti

onemogućena po podrazumevanoj vrednosti radi bezbednosti; može se omogućiti promenama u .VMKS konfiguracionoj datoteci); Deljenje među memorije između domaćina i gosta; Mogućnosti sinhronizacije rada procesora (virtuelna mašina se sinhronizuje sa satom host mašine); Podrška za Unity, funkciju koja omogućava besprekornu integraciju aplikacija sa radnom površinom domaćina, skrivanjem monitora virtuelne mašine i crtanjem prozora aplikacija koje rade u virtuelnoj mašini na hostu (Podrška za Unity je dodata za Windows 10 i uklonjena za Linux u Workstation 12 verziji) [47].

Da bi razvili adekvatnu simulaciju koja će u potpunosti podržavati sve aspekte realne mreže neophodno je imati odgovarajući simulacioni paket koji će podržavati kako mrežne elemente tako i virtuelne mašine. Odabran je paket GNS3 nam omogućava da prilagodimo svoje virtuelne mrežne laboratorije tako da tačno odgovaraju specifičnim potrebama. Paket podržava kreiranje neograničenog broja projekata zasnovanih na Cisco tehnologijama ili na tehnologijama drugih proizvođača mrežne opreme. Može se dodavati neograničen broj objekata projektima i pristupati im bilo kada, bez obzira da li postoji internet konekcija. GNS3 pruža maksimalnu fleksibilnost za projektovanje mreža kroz kombinaciju emuliranih hardverskih uređaja koji koriste realne mrežne operativne sisteme kao što je Cisco IOS, simulirane operativne sisteme kao što je NKS-OSv i mogućnost deljenja resursa na više realnih ili emuliranih računarskih mašina.

Osim toga implementirani grafički interfejs GNS3 nam omogućava da kreiramo virtuelizovane mrežne laboratorije sa raznim ruterima, prekidačima i računarima. Ali njegova najjača strana je uparenost sa IOS. Za razliku od sličnih aplikacija, GNS3 ne oponaša samo Cisco IOS komande ili funkcije. Umesto toga, koristi pozadinsku hipervizorsku aplikaciju da emulira hardver koji pokreće Cisco IOS. Pošto se emulira samo hardver, na računaru pokrećete stvarnu IOS datoteku odnosno sliku. Sve konfiguracione komande i izlaz se koriste iz kernela IOS-a, pa su nam tako dostupni svi protokoli ili funkcije koje podržava verzija IOS-a za projektovanje naše mreže. Ova funkcionalnost razlikuje GNS3 programima kao što su RouterSim, Boson NetSim ili VIRL, koji simuliraju celokupno iskustvo i pružaju samo ograničena okruženja, komande i scenarije sa kojima možete da radite.

Pored ove viskosofisticirane simulacije mrežnog hardvera, GNS3 integriše i podršku za simulirane operativne sisteme. Odnosno integriše QuickEmulator (QEMU) i VirtualBox virtuelne mašine na kojima mogu biti instalirani standardni računarski operativni sistemi kao Linux ili Windows.

GNS3 je zasnovan na klijent-server tehnologiji; slično kao što se veb pretraživač povezuje sa veb serverom da bi pristupio i prikazao veb stranice, program GNS3 preko grafičkog korisničkog interfejsa (GUI) pristupa GNS3 serveru, omogućavajući mu da pokrene, zaustavi i na drugi način kontroliše GNS3 uređaje. Ovo omogućava našim projektima da se skaliraju jer nisu ograničeni na pokretanje na jednom računaru. Ako radimo sa velikim ili složenim topologijama,

takođe možete pokrenuti GNS3 serverski program na drugom računaru. Ako bi imali pristup vrhunskom serveru sa puno memorije i procesorske snage, možemo instalirati GNS3 serverski program na hardver serveru, i istovremeno kontrolisati sve uređaje iz GNS3 topologije instalirane na slabijim klijent računarima.

Ovo je moguće jer ovaj paket omogućava virtuelno povezivanje. To jest možemo da umrežimo virtuelne uređaje zajedno, koristeći protokole kao što su Internet Protocol verzija 4 (IPv4) i Internet Protocol verzija 6 (IPv6). Tako možemo kreirati jednostavne laboratorije koje mogu da rade na samo jednom računaru [48].

Takođe GNS3 ima mogućnost da poveže virtuelne interfejsu u laboratorijskim uređajima sa jednim ili više fizičkih Ethernet interfejsa na našem računaru. Ovo nam omogućava da povežemo svoje virtuelne mreže sa stvarnim hardverom kao što su ruteri, prekidači i drugi računari. Na primer, možete da pokrenete dve ili više GNS3 mreže koristeći više računara i da povežete realne računare zajedno pomoću Ethernet ukrštenog kabla ili sviča. To vam daje mogućnost da povežete sve virtuelne GNS3 uređaje sa bilo kojom realnom mrežom i računarima na njoj.

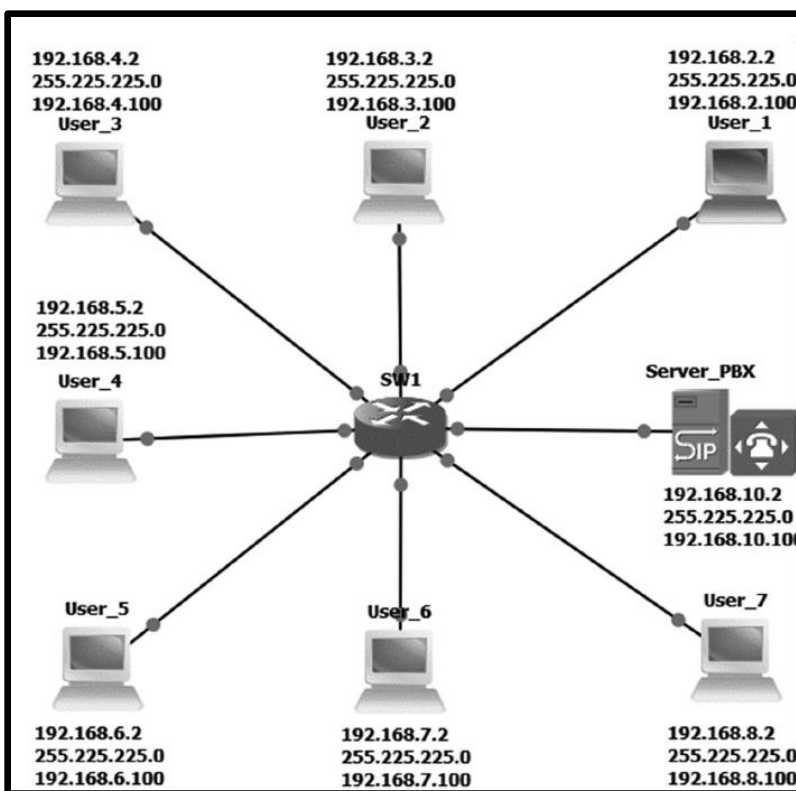
Konačno GNS3 koristi tehnologije otvorenog koda, uključujući Dinamips, QEMU i VirtualBok. Može da pokreće Juniper, Arista i mnoge druge mrežne operativne sisteme na isti način kao Cisco IOS. To znači da možemo dodati QEMU i VirtualBok virtuelne mašine u GNS3 i koristiti ih u svojim projektima. Ovi uređaji se mogu povezati sa drugim GNS3 uređajima kako bi se formirala kompletna mreža od kraja do kraja. Na primer, možete povezati VirtualBok host sa nizom svičeva i rutera i dozvoliti mu da pristupi resursima na drugom VirtualBok hostu na toj mreži. U ovom scenariju, možemo da konfiguriramo i testiramo sve vrste protokola za rutiranje, kao i funkcije kao što su prevođenje mrežnih adresa (NAT), liste kontrole pristupa (ACL) i virtuelne privatne mreže (VPN).

Sve sjajne stvari imaju svoju cenu, a GNS3 nije izuzetak. Najvažnije, GNS3 zahteva jednu ili više Cisco IOS slika za pokretanje na vašim virtuelnim Dinamips ruterima, a GNS3 ih ne obezbeđuje kao deo paketa. Naime slike se moraju kopirati sa rutera koji su nam u vlasništvu posedujete ili preko naloga Cisco veze na mreži (CCO), ako imamo ugovor sa Cisco [49].

Ovaj predlog je upućen QAU koji je proverio da li su svi predloženi protokoli standardizovani. On je takođe proverio da li predviđeni Softverski paketi nude podršku u vidu dokumentacije proizvoda, tehničkog papira, papira o kompatibilnosti i dokumentacije izvornog koda za slobodni softver. Svi projektni timovi su na osnovu poslatog odgovora predložili rešenje za ispunjavanje principa kvaliteta, kojih će se držati tokom implementacije istraživanja.

### **Standardi i topologije**

U skladu sa predloženom metodologijom Tim\_A je predložio razvoj sledeće topologije mreže date na slici 6. Topologija se sastoji od serverske mašine na kojoj je instaliran softver za PBX (Private Bench Exchange). Zatim od rutera koji će podeliti statičke IP adrese računarskim klijent mašinama na kojima će biti instalirani soft telefoni. Sistem će biti otvorenog tipa, tako da u svakom momentu možemo da dodamo dodatne mašine, dodatne IP adrese kao i dodatne brojeve soft telefonima. Pošto GLP/GMP pravila nalažu standardizaciju vršena je analiza VoIP protokola i za razvoj je usvojem SIP protokol. Ovaj protocol je standardizovan u RFC 3261 [50]. Ovaj protokol podržavaju svi VoIP sistemi na tržištu, ko i ostali signalni protokoli u on se oslanja na Realtime Transport Protocol (RTP) za prenos paketa izmedju izvora i prijemnika. Takodje u standardu RFC 3261 su definisani i svi pomoćni protokoli a koji mogu biti deo SIP-a.



Slika 6 Topologija mreže realnog i simuliranog VoIP sistema

## VOIP PROTOKOLI

Postoji nekoliko VoIP – specifičnih protokola, ali svi su oni svrstani, odnosno pripadaju dvema kategorijama, i to:

- Signalni protokoli,
- Transportni protokoli.

Iako VoIP arhitektura je potpuno drugačija od one koja se koristi od strane tradicionalne telefonije, i dalje postoji osnovna potreba za signalizacijom. Nekako, telefoni moraju da zvone, brojevi se moraju saopštiti, a putanje moraju biti definisane, i tim funkcijama upravljaju signalni protokoli. Tri najčešća tipa signalnih protokola su H.323, Skinny i Session Initiation Protocol, ili SIP.

### **H.323 protokol**

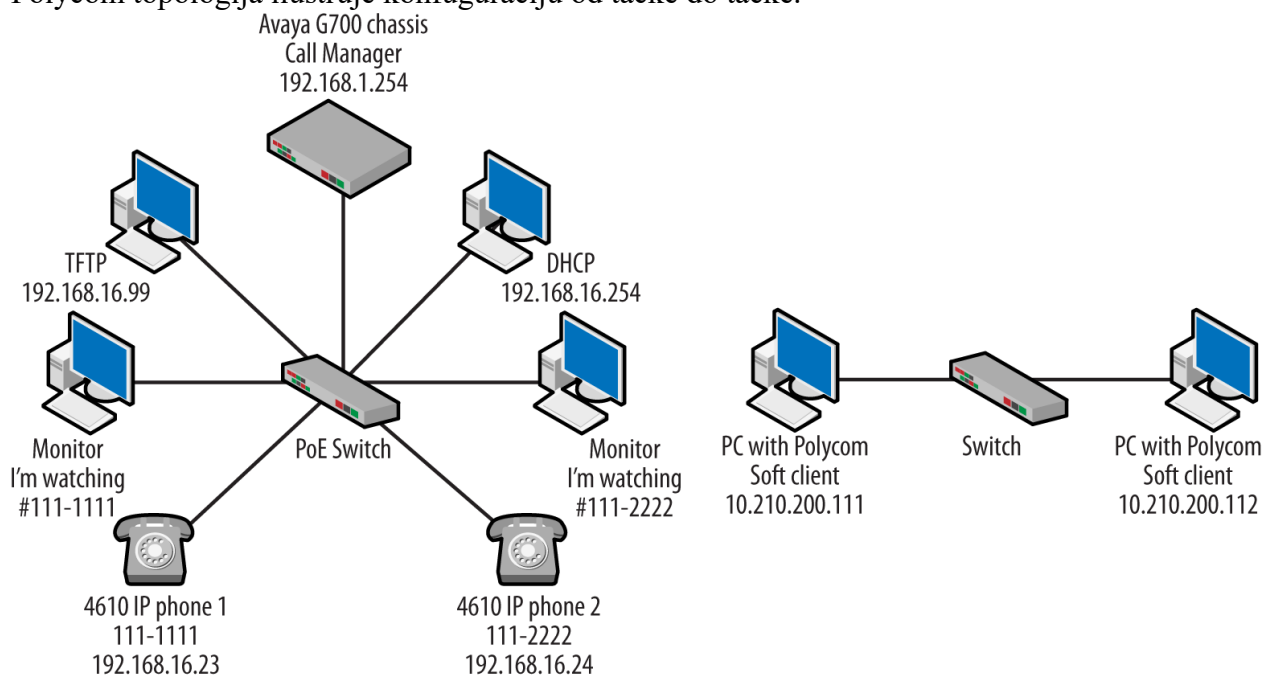
Devedesete godine prošlog veka iznedrile su mnoge promene načina komunikacije. Među njima su video konferencije, a i ideja da glas može biti inkapsuliran u paketima podataka. ITU-T (International Telecommunications Union–Telecom) H.323 standard datira iz tog vremena. Iako je ažuriran mnogo puta, većine ideja koje su nakon toga implementirane su ga samo unapređivale. Osnovni problem je konverzija signalizacije koja koristi tradicionalnu telekomunikacionu mrežu zasnovanu na IP protokolu. Umesto izmišljanja nove tehnike i protokola, mnoge ideje su zadržane, ali pakovanje je tokom vremena menjano. Tako H.323 koristi kao svoju osnovu mnoge protokole i standarde koji su povezani sa javnom mrežom, ili PSTN (Public Switched Telephone Network), koja je inkapsulirana u protokolima kao što su TCP, UDP ili IP. Pored toga, vidi se preporuka usklađivanja poruka iz ISDN (Integrated Service Digital Network) mreže, kao što je poziv na signalizaciju [51].

Ovaj protokol je bio standard za video konferencije, koje uključuju glas, video, i podatke. Zbog toga što se bavi takvim stvarima kao što je kontrola poziva i upravljanje njime od tačke do tačke (kao i više tačaka), administracija mrežnog prolaza medijskog saobraćaja, parametrima opsega, kao i učešćem korisnika, postao je defakto standard za Internet telefoniju, kao i za VoIP. Iako ovo više nije slučaj zbog pojačanja SIP-a, veliki broj korisnika ga i dalje upotrebljava H.323 pakete zbog jasnoće njegovih operacija. Takođe i zbog značajne količine instalirane opreme se ovaj protokol i dalje koristi, a interoperabilnost je vrlo bitna.

H.323 je u stvari deo kasnijih H.3XX serija preporuka iz ITU-T i predstavlja zbirku subprotokola. H.323 može biti raspoređen u bilo TCP ili UDP protokolu, i koristi portove 1719 i 1720, u zavisnosti od operacije. VoIP razmeštanje ove vrste će koristiti RTP protokolu za prenos glasovnih i video podataka između tačaka.

H.323 je izgrađen na bazi telefonije, i, u stilu ITU-T preporuka, predstavlja vrlo kompleksan skup protokola. Preporuke od strane ITU-T opisuju strukturu i operacije u kojima se diskutuje svaka mogućnost i zato imaju veliki broj polja. Polja, s druge strane, imaju mnoge kombinacije koje mogu načiniti stvari zbunjujućima, pa se moraju poznavati kako se takve stvari ne bi dešavale [52].

Dve topologije na slici 7 koriste se za opis ovog protokola. Iako koriste iste protokole, topologije su različite. Avaya topologija sadrži arhitekturu koja podržava date protokole, dok Polycom topologija ilustruje konfiguraciju od tačke do tačke.



Slika 7 Topologije

Na slici su prikazane topologije koje će se koristiti za opis H.323 standarda. (Preuzeto sa Hartpence B., (2013), Packet guide to Voice over IP, O'Reilly Media, Sebastopol, ISBN: 978-1-449-33967-8).

Postoji veliki broj terminala koji može podržati ovaj protokol, uključujući one koji se baziraju na ISDN, kao i tradicionalne telefonske uređaje.

Postoji nekoliko komponenti koje se nalaze u H.323 sistemu, uključujući terminale, mrežne prolaze, gatekeepers (čuvare prolaza), multipoint kontrolere koji komuniciraju preko videa, glasa, podataka, komunikacija i call-control poruka [53]. Ovo su informacioni protoci. Pojedinačno:

#### Terminali

Takođe poznati kao krajnje tačke sa H.225 i H.245 signalizacijom, kontrolnim jedinicama sistema, prenosom medija, audio i video kodekom, kao i paket-baziranim mrežnim interfejsom.

#### Mrežni prolaz

Upravlja komunikacijom i translacijom između H.323 terminala na IP mreži i switched-circuit mrežom kao što je Public Switched Telephone Network. Upravlja podešavanjima poziva, završetkom poziva i mogućom translacijom između kodeka preko H.225.

```

User Datagram Protocol, Src Port: 49301 (49301), Dst Port: h323gatestat (1719)
H.225.0 RAS
  ▫ RasMessage: gatekeeperRequest (0)
    ▫ gatekeeperRequest
      requestSeqNum: 2
      protocolIdentifier: 0.0.8.2250.0.5 (Version 5)
      ▫ nonStandardData
      ▫ rasAddress: ipAddress (0)
      ▫ endpointType
      ▫ endpointAlias: 1 item
      ▫ tokens: 1 item
      ▫ authenticationCapability: 2 items
      ▫ algorithmOIDs: 2 items
      ▫ featureSet

Transmission Control Protocol, Src Port: 4296 (4296), Dst Port: h323hostcall (1720)
TPKT, Version: 3, Length: 1367
Q.931
H.225.0 CS
  ▫ H323-UserInformation
    ▫ h323-uu-pdu
      ▫ h323-message-body: setup (0)
        ▫ setup
          protocolIdentifier: 0.0.8.2250.0.5 (Version 5)
          ▫ sourceInfo
            ...0 .... activeMC: False
            conferenceID: 00000000-0000-1000-0000-0000c0a81017
          ▫ conferenceGoal: callIndependentSupplementaryService (4)
          ▫ callType: pointToPoint (0)
          ▫ callIdentifier
          ▫ fastStart: 36 items
            1... .... mediaWaitForConnect: True
            1... .... canOverlapSend: True
            0... .... h245Tunnelling: False

```

Slika 8 Brojevi i portovi H.225 verzije

Na slici 8 je screenshot opcijom uhvaćeno polje iz Wireshark programa, gde se mogu videti brojevi i portovi H.225 verzije, iz topologija koje su navedene na str.39. (Preuzeto sa Hartpence B., (2013), Packet guide to Voice over IP, O'Reilly Media, Sebastopol, ISBN: 978-1-449-33967-8)

#### Gatekeeper

Ovo je opciona komponenta koja upravlja call-control servisima kao što su zahtev za lokacijom i potvrda, zatim rezolucija imena, prijemi kontrola opsega, takođe preko H.225.

#### Multipoint kontroler

Kontroler podržava konferencije između tri ili više H.323 krajnje tačke. Takođe upravlja razmeni sposobnosti za komunikaciju čvorova. To se događa kroz H.245.

Ostale komponente su multipoint procesor i multipoint kontrolna jedinica.

H.225 (“pakovanje” za signalne protokole i medijske tokove podataka za paket-bazirane sisteme za komunikaciju multimedijjskih sadržaja) ima za primarnu funkciju upravljanja audio, video sadržajem, kao i kontrolisanje informacija kako bi obezbedio “usluge konverzacije” koje trebaju H.323 opremi. Kao i H.323, postoji nekoliko verzija ovog standarda.

Q.931 je još jedan podprotokol koji se često povezuje sa signalizacijom poziva u H.323 okruženjima. Zapravo se zove “ISDN user-network interface layer 3 specification for basic call control” (ISDN specifikacija za 3 sloj korisničko-mrežnog interfejsa za osnovnu kontrolu poziva).

TPKT je podprotokol koji postoji u svrhu izveštavanja ne-TCP/IP aplikacije ka onome što danas zovemo IP-bazirano okruženje. ISO (International Standards Organization) usluge transporta ili protokoli transporta nisu IP-bazirani. H.225 kao i H.245 koriste TPKT.

H.323 takođe navodi četvrti sloj, ili transportne protokole koji će se koristiti u arhitekturi; naime, navodi da je pouzdan TCP (Transport Control Protocol) i da se treba koristiti za podršku i H.225 i H.245. nepouzdan UDP je obavezan za audio, video i RAS1 kanale. Odnosi se na H.225 kao kontrola pre obavljenog poziva u H.323 okruženju sa gatekeeper-om (čuvar mrežne kapije).

H.245 je kontrolni protokol za multimedijalne komunikacije i upravlja sposobnostima razmene podataka između H.323 krajnje tačke i konfiguracije logičkih kanala između njih.

H.235 opisuje sigurnosne konfiguracije.

Klasična veza bazirana na H.323 prikazana je na slici 9. Ova lista dolazi iz Polycom topologije (Slika 32.) koja ne koristi call manager-a, što znači da joj čvorovi nisu međusobno povezani. Na listi se mogu videti i H.225 i H.245 poruke.

10	10.210.200.112	10.210.200.111	H.225.0	CS: setup
12	10.210.200.111	10.210.200.112	H.225.0	CS: alerting
18	10.210.200.111	10.210.200.112	H.225.0	CS: connect
25	10.210.200.112	10.210.200.111	H.245	terminalCapabilitySet
27	10.210.200.112	10.210.200.111	H.245	masterSlaveDetermination
31	10.210.200.111	10.210.200.112	H.245	terminalCapabilitySet
33	10.210.200.111	10.210.200.112	H.245	masterSlaveDetermination
38	10.210.200.112	10.210.200.111	H.245	terminalCapabilitySetAck
39	10.210.200.112	10.210.200.111	H.245	masterSlaveDeterminationAck
41	10.210.200.111	10.210.200.112	H.245	terminalCapabilitySetAck
42	10.210.200.111	10.210.200.112	H.245	masterSlaveDeterminationAck
44	10.210.200.112	10.210.200.111	H.245	openLogicalChannel (generic)
46	10.210.200.112	10.210.200.111	H.245	openLogicalChannel (genericVideoCapability)

Slika 9 H.225 i H.245 lista paketa

Na slici je screenshot opcijom uhvaćeno polje iz Wireshark programa, gde se mogu videti lista paketa protokola H.225 i H.245. (Preuzeto sa Hartpence B., (2013), Packet guide to Voice over IP, O'Reilly Media, Sebastopol, ISBN: 978-1-449-33967-8)

Ponekad se H.245 poruke infiltriraju unutar H.225. To se može videti na dijagramu toka na slici 10.

Time	10.210.200.112	10.210.200.111	Comment
5.709857	(3232) →	setup (1720)	H225 From: To: TunnH245:off FS:off
5.882944	(3232) ←	alerting (1720)	H225 TunnH245:off FS:off
10.702663	(3232) →	connect (1720)	H225 TunnH245:off FS:off
10.991370	(3233) →	TCS (3231)	H245 terminalCapabilitySet
10.991521	(3233) →	MSD (3231)	H245 masterSlaveDetermination
11.310257	(3233) ←	TCS (3231)	H245 terminalCapabilitySet
11.310394	(3233) ←	MSD (3231)	H245 masterSlaveDetermination
12.140347	(3233) →	TCSAck (3231)	H245 terminalCapabilitySetAck
12.140352	(3233) →	MSDAck (3231)	H245 masterSlaveDeterminationAck
12.196862	(3233) ←	TCSAck (3231)	H245 terminalCapabilitySetAck
12.196866	(3233) ←	MSDAck (3231)	H245 masterSlaveDeterminationAck
12.824751	(3233) →	OLC (generic) (3231)	H245 openLogicalChannel
13.567298	(3233) →	OLC (genericVideoCapability) (3231)	H245 openLogicalChannel
13.762027	(3233) ←	OLC (generic) (3231)	H245 openLogicalChannel
14.070986	(3233) →	OLCAck (3231)	H245 openLogicalChannelAck
14.103243	(3233) →	OLC (genericVideoCapability) (3231)	H245 openLogicalChannel
14.247915	(3233) →	OLC (genericVideoCapability) (3231)	H245 openLogicalChannel
14.382810	(3233) →	OLC (genericDataCapability) (3231)	H245 openLogicalChannel
14.664064	(3233) ←	OLCAck (3231)	H245 openLogicalChannelAck
14.856877	(3233) →	OLCAck (3231)	H245 openLogicalChannelAck
14.932047	(3233) →	FCC (3231)	H245 flowControlCommand
14.991657	(3233) →	MC (3231)	H245 miscellaneousCommand
15.215888	(3233) →	OLC (genericVideoCapability) (3231)	H245 openLogicalChannel
15.355210	(3233) ←	OLCAck (3231)	H245 openLogicalChannelAck
15.393647	(3233) ←	FCC (3231)	H245 flowControlCommand
15.437293	(3233) ←	MC (3231)	H245 miscellaneousCommand
15.466774	(3233) →	OLCAck (3231)	H245 openLogicalChannelAck
15.546179	(3233) →	MC (3231)	H245 miscellaneousCommand
15.656911	(3230) →	RTP (RTPTType-127) (3230)	RTP Num packets:1187 Duration:47.382s SSRC:0xCF3B3B01
15.669859	(3233) ←	MI (3231)	H245 miscellaneousIndication
15.866618	(3233) ←	FCIndication (3231)	H245 flowControlIndication

Slika 10 Dijagram toka paketa

Na slici je prikazan dijagram protoka poziva, kao i protokole koji učestvuju u konverzaciji, do određenog trenutka. (Preuzeto sa Hartpence B., (2013), Packet guide to Voice over IP, O'Reilly Media, Sebastopol, ISBN: 978-1-449-33967-8)

### Skinny protokol

Skinny Client Control Protocol (SCCP) ili samo Skinny je Cisco protokol za signalizaciju VoIP uređaja. Kao signalni protokol, koristi se za registraciju krajnjih tačaka;

pozivnih poruka kao što su off-hook, on-hook, addressing (adresirajući); kao i za kontrolu displeja telefona. Kao što ime kaže, to je jednostavan protokol, i lak za razumevanje. Kada se kombinuje sa Cisco menadžerom za pozive – CME (Call Manager Express), mala VoIP raspoređivanja podržavaju brojne funkcije i telefoni mogu biti postavljeni vrlo brzo [54].

SCCP je u privatnom vlasništvu, za razliku od drugih protokola čiju je dokumentaciju moguće lako nabaviti. Masivni Cisco Web sajt i kompleksnost Cisco IOS mogu dovesti do nekih poteškoća u razumevanju čak osnovnih operacija. U svakom slučaju, veliki broj uređaja koristi SCCP, zato ga je potrebno obraditi.

Kao i većina VoIP protokola za signalizaciju, SCCP ima jasno definisani skup poruka i ponašanja koja se mogu posmatrati i ponavljati. Međutim, SCCP takođe implementira nekoliko metoda koje nisu standardne. To znači da, ako ste upoznati sa operacijama SIP-a, kao i H.323 protokola, ne znači da će SCCP imati iste, a čak ni slične operacije [55].

Krajem 1998. godine Cisco je kupio kompaniju Selsius, koja je vodeći dobavljač proizvoda IP telefonije, a SCCP priča počinje upravo tu. U stvari, nekoliko datoteka i naziva proizvoda koji se koriste u Cisco VoIP počinju sa „SEP“, što znači Selsius Ethernet Phone.

Skinny je protokol za signalizaciju, što znači da ima poruke koje se koriste od strane telefona za registraciju sa call serverom, uspostavljanjem poziva, prekidanjem poziva, karakteristikama poziva, konfiguraciji interfejsa, itd. Kao i drugi signalni protokoli, SCCP je obuhvaćen u TCP (Transmission Control Protocol) paketima. Podsetimo da TCP radi na 4. sloju TCP/IP i Open Systems Interconnecting (povezivanje otvorenih sistema) mrežnih modela.

Kao signalni protokol, SCCP se ne koristi za prenos podataka i glasa, tako da, kao i većina VoIP arhitektura, koristi Real Time Transport Protocol (RTTP) za ovu svrhu. Ovo znači da telefoni registruju, primaju informacije interfejsa, kao i zovu koristeći Skinny protokol i poruke, ali čim glasovni podaci treba da se prenesu na drugu stranu, koristi se RTP. Kada je RTP u igri, 4. sloj protokola prebacuje se na User Datagram Protocol (UDP).

SCCP poruke su vrlo čitljive, a imena su im opisna. Dizajniran je za korišćenje sa call serverom i komunicira na osnovu TCP protokola, port 2000. Primer Skinny protokola nalazi se na slici 11.

```

Ethernet II, Src: Cisco_61:5f:16 (00:13:c4:61:5f:16), Dst: Cisco_f4:c2:10 (00:1c:58:f4:c2:10)
Internet Protocol Version 4, Src: 192.168.1.2 (192.168.1.2), Dst: 192.168.1.254 (192.168.1.254)
Transmission Control Protocol, Src Port: 50201 (50201), Dst Port: cisco-sccp (2000), Seq: 209,
Skippy Client Control Protocol
Data length: 56
Header version: 0 (0x00000000)
Message ID: RegisterMessage (0x00000001)
Device name: SEP0013C4615F16
Station user ID: 0
Station instance: 1
IP address: 192.168.1.2 (192.168.1.2)
Device type: TelecasterMgr (7)
Max streams: 0

```

---

00 1c 58 f4 c2 10 00 13	c4 61 5f 16 08 00 45 68	..X..... .a....Eh
00 68 04 07 00 00 40 06	f1 d0 c0 a8 01 02 c0 a8	.h....@. ....
01 fe c4 19 07 d0 40 0e	6e 47 d2 32 5f 70 50 18	.....@. nG.2_pP.
05 78 88 2b 00 00 38 00	00 00 00 00 00 00 01 00	.x+..8. ....
00 00 53 45 50 30 30 31	33 43 34 36 31 35 46 31	..SEP001 3C4615F1
36 00 00 00 00 00 01 00	00 00 c0 a8 01 02 07 00	6.....
00 00 00 00 00 00 00 00	00 00 06 00 00 84 00 00	.....
00 00 00 00 00 00		.....

Slika 11 Skinny registration poruka

Na slici je screenshot opcijom uhvaćeno polje iz Wireshark programa, gde se može videti Skinny REGISTRATION poruka. Naslov je uokviren crvenim markerom. (Preuzeto sa Hartpence B., (2013), Packet guide to Voice over IP, O'Reilly Media, Sebastopol, ISBN: 978-1-449-33967-8)

Skinny poruke počinju sa istim setom polja koja variraju kako ID poruke (Message ID) omogućava indikaciju tipa poruke.

Data length (dužina podataka)

4 bita, ovo je količina podataka koji se nalaze u paketu, izvan zaglavlja.

Header version (verzija zaglavlja)

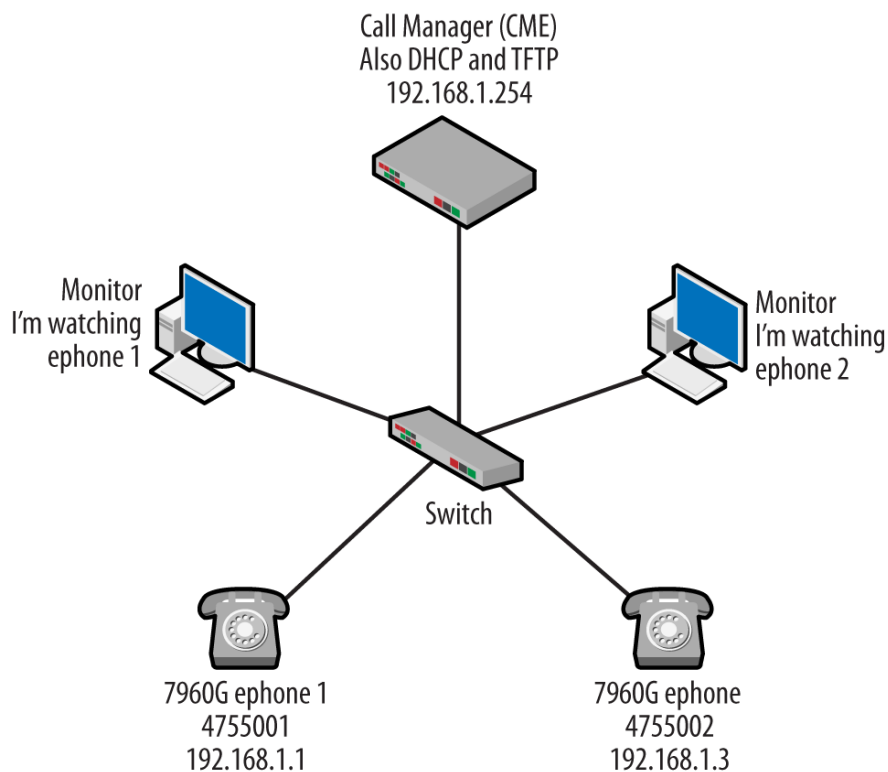
4 bita. Uobičajena verzija je 0x00000000, što znači da je u pitanju osnovno zaglavlje.

Message ID (ID poruke)

6 bita, omogućava heksadecimalnu vrednost za tip SCCP paketa.

Različiti paketi variraju u veličini i svrsi [56]. Dakle, nakon ovih polja, zaglavlje postaje specifično.

Na slici 12 nalazi se primer topologije Skinny protokola. U centru topologije nalazi se Cisco 3550 svič. Mašine sa monitorima su takođe uključene sa ciljem posmatranje saobraćaja koji se generiše. Dakle, svaka krajnja tačka ima svoju mašinu za monitoring kako bi se sve poruke uspešno propratile.



Slika 12 Skinny topologija

Na slici su prikazane topologije koje će se koristiti za opis Skinny protokola. (Preuzeto sa Hartpence B., (2013), Packet guide to Voice over IP, O'Reilly Media, Sebastopol, ISBN: 978-1-449-33967-8)

Ruter je Cisco 2811, verzije 12.4 IOS i verzije 3.3 Call Manager Express-a. Ruter će biti konfigurisan ne samo kao call server, već kao i DHCP server. DHCP konfiguracija rutera je vrlo jasna u datoj topologiji, zahteva samo mrežu, prolaz i TFTP server kroz opciju 150. Tehnički, može se i bez mrežnog prolaza, sa obzirom da se neće ići offline. Adresa 192.168.1.254 je isključena iz grupe kako bi se sprečilo dodeljivanje mrežnih čvorova ili telefona.

Lokacija poziva servera nije deo DHCP konfiguracije. Adresa call servera se u stvari čita iz fajlova koji se downloaduju sa TFTP servera. Primer komande koja se koristi za pokretanje TFTP servera na ruteru:

```
tftp-server P00307020400.bin
```

Dati fajl (P00307020400.bin) je sačuvan na fleš karti rutera.

Pošto su neki od fajlova automatski generisani, najbolje je koristiti CME ruter kao i TFTP server kojem se pristupa preko telefona. Kao i većina VoIP telefona, Cisco telefoni imaju sistem za pokretanje protokola koji omogućava poslednju konfiguraciju i firmware. Kad se telefon učita, ovaj mehanizam čita fajlove kako bi odredio koji signalni protokol i konfiguracija će se učitati. TFTP server obezbeđuje konfiguracijske fajlove za telefon. Zapravo postoji sedam fajlova u Cisco algoritmu korišćeni od strane UAL (Universal Application Loader) koji mogu biti uključeni u

proces, mada ne na jednom telefonu. Na primer, jedan telefon neće koristiti različite signalne protokole u isto vreme.

Generalno, VoIP telefoni imaju kolekciju prepoznatljivih faza i operacija. Za praćenje i snimanje svih paketa koje proističu iz ovih faza koriste se monitori iz Slike 12. Veze će se nastaviti na ovaj način:

VoIP telefoni dobijaju DHCP-bazirane adrese.

VoIP telefoni kontaktiraju TFTP server za izmenjene datoteke.

Telefoni se registruju sa call serverom.

Telefoni kontaktiraju call server sa brojem destinacija.

Call server kontaktira određeni telefon.

Call server informiše oba kraja o parametrima poziva.

Telefoni direktno komuniciraju.

Jedan od telefona završava poziv.

Call server obavlja „čišćenje“ operacija.

Kada je telefon priključen, dve konverzacije koje se odvijaju prvo su sa DHCP i TFTP-om. DHCP server takođe omogućava IP adresu TFTP servera preko opcije 150. Komande na ruteru vraćaju ga TFTP serveru i usmeravaju na određene datoteke koje su potrebne za podršku telefonima.

Nakon ovoga, telefon kontaktira call server sa standardnim TCP trosmernim rukovanjem<sup>2</sup> (Slika 13). Sada je telefon spreman za registraciju.

```
TCP 50752 > cisco-sccp [SYN] Seq=0 win=1400 Len=0 MSS=1400
TCP cisco-sccp > 50752 [SYN, ACK] Seq=0 Ack=1 win=4128 Len=0 MSS=1400
TCP 50752 > cisco-sccp [ACK] Seq=1 Ack=1 win=1400 Len=0
```

Slika 13 TCP rukovanje

Na slici je screenshot opcijom uhvaćeno polje iz Wireshark programa, gde se može videti TCP rukovanje. (Preuzeto sa Hartpence B., (2013), Packet guide to Voice over IP, O'Reilly Media, Sebastopol, ISBN: 978-1-449-33967-8)

Prve Skinny poruke se pojavljuju kako telefon počinje proces registracije. Prva poruka je u stvari alarmna poruka koja adresira zadnji poznati status telefona.

```

Ethernet II, Src: Cisco_61:5f:16 (00:13:c4:61:5f:16), Dst: Cisco_f4:c2:10 (00:1c:58:f4:c2:10)
Internet Protocol Version 4, Src: 192.168.1.1 (192.168.1.1), Dst: 192.168.1.254 (192.168.1.254)
Transmission Control Protocol, Src Port: 50752 (50752), Dst Port: cisco-sccp (2000), Seq: 209,
Skinny Client Control Protocol
  Data length: 56
  Header version: basic (0x00000000)
  Message ID: RegisterMessage (0x00000001)
  Device name: SEP0013c4615f16
  Station user ID: 0
  Station instance: 1
  IP address: 192.168.1.1 (192.168.1.1)
  Device type: TelecasterMgr (7)
  Max streams: 0

```

Slika 14 Skinny register poruka

Na slici je screenshot opcijom uhvaćeno polje iz Wireshark programa, gde se može videti Skinny REGISTER poruka. Naslov poruke uokviren je crvenim markerom. (Preuzeto sa Hartpence B., (2013), Packet guide to Voice over IP, O'Reilly Media, Sebastopol, ISBN: 978-1-449-33967-8)

Na slici 15 Prikazana je Register poruka. Sa ove slike, vidi se da je poruka unicast3 od telefona direktno ka call serveru (192.168.1.254) i da poruke koriste TCP port 2000. Register poruka ima ID poruke, i ime uređaja koje se koristi u poruci (SEP0013c4615F16), a takođe uključuje MAC adresu telefona. Call server odgovara sa RegisterACKMessage (Slika 40.)

```

Internet Protocol Version 4, Src: 192.168.1.254 (192.168.1.254), Dst: 192.168.1.1 (192.168.1.1)
Transmission Control Protocol, Src Port: cisco-sccp (2000), Dst Port: 50752 (50752), Seq: 1, Ack: 273
Skinny Client Control Protocol
  Data length: 24
  Header version: basic (0x00000000)
  Message ID: RegisterAckMessage (0x00000081)
  Keep-alive interval: 30
  Date template: M/D/YA
  Secondary keep-alive interval: 60

```

Slika 15 Skinny RegisterACK poruka

Na slici je screenshot opcijom uhvaćeno polje iz Wireshark programa, gde se može videti Skinny REGISTER ACK poruka. Naslov poruke uokviren je crvenim markerom. (Preuzeto sa Hartpence B., (2013), Packet guide to Voice over IP, O'Reilly Media, Sebastopol, ISBN: 978-1-449-33967-8)

Ova poruka takođe sadrži keep-alive vrednost. Keep-alive obezbeđuje „heartbeat“ (kucanje srca) registracije, kako bi se osiguralo da je telefon još uvek u funkciji. Nakon uspešne registracije, telefon i call server razmenjuju informacije o podešavanju interfejsa i podržanih karakteristika. Ovo se vrši preko Capability and Template poruka (poruke sposobnosti i šablona). Na primer, server traži od telefona njegove sposobnosti, a telefon odgovara, kao na slici 16.

No.	Source	Destination	Protocol	Info
222	192.168.1.254	192.168.1.1	SKINNY	CapabilitiesReqMessage
226	192.168.1.1	192.168.1.254	SKINNY	CapabilitiesResMessage

[ ] Frame 222: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)  
 [ ] Ethernet II, Src: Cisco\_f4:c2:10 (00:1c:58:f4:c2:10), Dst: Cisco\_61:5f:16 (00:13:c4:61:5f:16)  
 [ ] Internet Protocol Version 4, Src: 192.168.1.254 (192.168.1.254), Dst: 192.168.1.1 (192.168.1.1)  
 [ ] Transmission Control Protocol, Src Port: cisco-sccp (2000), Dst Port: 50752 (50752), Seq: 33, Ack: 273  
 [ ] Skinny Client Control Protocol  
     Data length: 4  
     Header version: Basic (0x00000000)  
     Message ID: CapabilitiesReqMessage (0x0000009b)

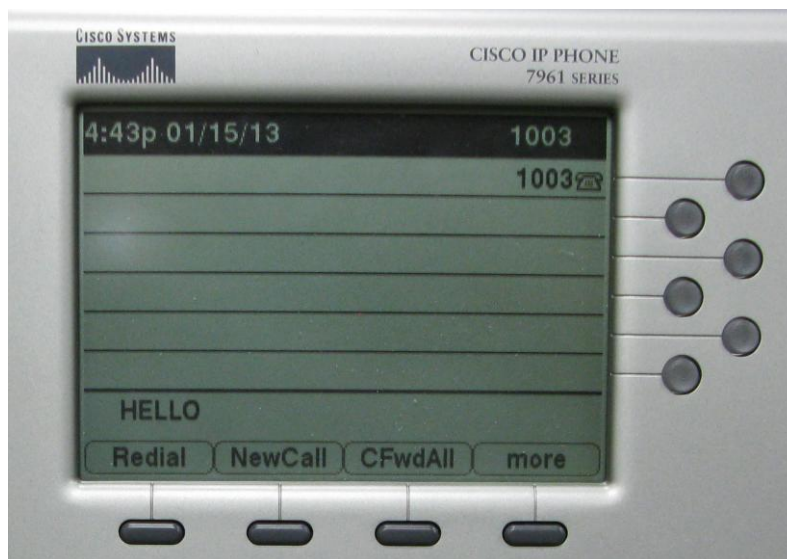
  

[ ] Frame 226: 182 bytes on wire (1456 bits), 182 bytes captured (1456 bits)  
 [ ] Ethernet II, Src: Cisco\_61:5f:16 (00:13:c4:61:5f:16), Dst: Cisco\_f4:c2:10 (00:1c:58:f4:c2:10)  
 [ ] Internet Protocol Version 4, Src: 192.168.1.1 (192.168.1.1), Dst: 192.168.1.254 (192.168.1.254)  
 [ ] Transmission Control Protocol, Src Port: 50752 (50752), Dst Port: cisco-sccp (2000), Seq: 289, Ack: 45  
 [ ] Skinny Client Control Protocol  
     Data length: 120  
     Header version: Basic (0x00000000)  
     Message ID: CapabilitiesResMessage (0x00000010)  
     Capabilities count: 7  
     Payload capability: Wideband 256k (25)  
     Max frames per packet: 120  
     Payload capability: G.711 u-law 64k (4)  
     Max frames per packet: 40  
     Payload capability: G.711 A-law 64k (2)  
     Max frames per packet: 40  
     Payload capability: G.729 Annex B (15)  
     Max frames per packet: 60  
     Payload capability: G.729 Annex A+Annex B (16)  
     Max frames per packet: 60  
     Payload capability: G.729 (11)  
     Max frames per packet: 60  
     Payload capability: G.729 Annex A (12)  
     Max frames per packet: 60

Slika 16 Skinny Capability razmena poruka

Na slici je screenshot opcijom uhvaćeno polje iz Wireshark programa, gde se može videti Skinny cabability razmena poruka. Uokvireni su zahtev za razmenom sposobnosti telefona, kao i odgovor u kojem se šalju tražene informacije. (Preuzeto sa Hartpence B., (2013), Packet guide to Voice over IP, O'Reilly Media, Sebastopol, ISBN: 978-1-449-33967-8)

Informacije sadržane u okviru 226 ukazuju da telefon može da podrži kolekciju kodeka iz ITU-T G serije. Jednom kada je telefon prošao kroz ceo proces registracije, interfejs na telefonu izgleda kao na slici 17.



Slika 17 Registrovani telefon

Na slici je prikazan izgled interfejsa na telefonu, nakon prolaska kroz process registracija. (Preuzeto sa Hartpence B., (2013), Packet guide to Voice over IP, O'Reilly Media, Sebastopol, ISBN: 978-1-449-33967-8)

Ali kako je ova informacija stigla do telefona? Nekoliko Skinny poruka se koriste za interfejs. Od rutera i call manager konfiguracionih linija je poznato da je telefonski broj za ephone 1 475-5001. Skinny LineStatMessage se koristi kako bi rekla telefonu da bi ovaj broj trebao biti na liniji broj 1. (Slika 18).

```

Skinny Client Control Protocol
Data length: 116
Header version: Basic (0x00000000)
Message ID: LineStatMessage (0x00000092)
Line number: 1
Line directory number: 4755001
Fully qualified display name: 4755001
Display name:
  
```

Slika 18 Skinny LineStat poruka

Na slici je screenshot opcijom uhvaćeno polje iz Wireshark programa, gde se može videti Skinny LineStat poruka, uokvirena crvenim markerom. (Preuzeto sa Hartpence B., (2013), Packet guide to Voice over IP, O'Reilly Media, Sebastopol, ISBN: 978-1-449-33967-8)

I Skinny DisplayPromptStatusMessage omogućuje poruku sistema "Packets!", što se može videt dešifrovano na slici 19.

```

Skinny Client Control Protocol
Data length: 48
Header version: Basic (0x00000000)
Message ID: DisplayPromptStatusMessage (0x00000112)
Message time-out: 0
Display message: Packets!
Line instance: 0
Call identifier: 0

```

Slika 19 Skinny DisplayPrompt poruka

Na slici je screenshot opcijom uhvaćeno polje iz Wireshark programa, gde se može videti Skinny DisplayPrompt poruka, uokvirena crvenim markerom. (Preuzeto sa Hartpence B., (2013), Packet guide to Voice over IP, O'Reilly Media, Sebastopol, ISBN: 978-1-449-33967-8)

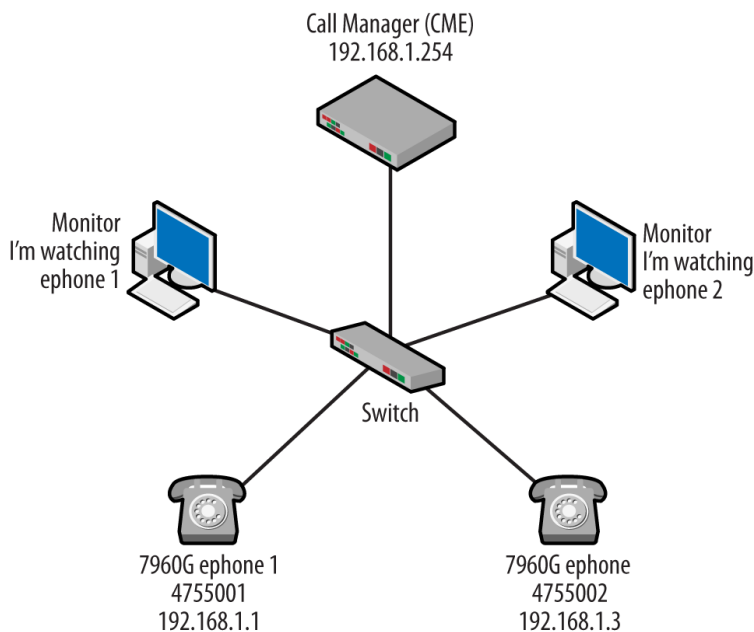
Skinny poruke su vrlo opisne i čitljive. Na slici 20 možemo videti KeepAliveMessage se pojavljuje u zadnjem redu. Na kraju, ova poruka je jedina koja nastavlja da se proizvodi, po stopi opisanoj u RegisterAckMessage.

No.	Source	Destination	Protocol	Info
217	192.168.1.1	192.168.1.254	SKINNY	AlarmMessage
218	192.168.1.1	192.168.1.254	SKINNY	AlarmMessage
219	192.168.1.1	192.168.1.254	SKINNY	RegisterMessage
224	192.168.1.1	192.168.1.254	SKINNY	HeadsetStatusMessage
226	192.168.1.1	192.168.1.254	SKINNY	CapabilitiesResMessage
227	192.168.1.1	192.168.1.254	SKINNY	HeadsetStatusMessage
228	192.168.1.1	192.168.1.254	SKINNY	ButtonTemplateReqMessage
231	192.168.1.1	192.168.1.254	SKINNY	SoftKeyTemplateReqMessage
234	192.168.1.1	192.168.1.254	SKINNY	SoftKeySetReqMessage
243	192.168.1.1	192.168.1.254	SKINNY	LineStatReqMessage
248	192.168.1.1	192.168.1.254	SKINNY	LineStatReqMessage
251	192.168.1.1	192.168.1.254	SKINNY	LineStatReqMessage
255	192.168.1.1	192.168.1.254	SKINNY	LineStatReqMessage
259	192.168.1.1	192.168.1.254	SKINNY	LineStatReqMessage
264	192.168.1.1	192.168.1.254	SKINNY	LineStatReqMessage
272	192.168.1.1	192.168.1.254	SKINNY	RegisterAvailableLinesMessage
274	192.168.1.1	192.168.1.254	SKINNY	TimeDateReqMessage
291	192.168.1.1	192.168.1.254	SKINNY	KeepAliveMessage

Slika 20 Skinny lista poruka

Na slici je screenshot opcijom uhvaćeno polje iz Wireshark programa, gde se može videti primer liste paketa koji se razmenjuju između korisnika, tačnije protokola koji se u tu svrhu koristi, a to je Skinny. (Preuzeto sa Hartpence B., (2013), Packet guide to Voice over IP, O'Reilly Media, Sebastopol, ISBN: 978-1-449-33967-8)

Nakon inicijalnog razgovora (DHCP, TFTP i Registracija), topologija se može ažurirati kao na slici 21.



Slika 21 Ažurirana topologija

Na slici je prikazana ažurirana topologija, nakon početnog razgovora. (Preuzeto sa Hartpence B., (2013), Packet guide to Voice over IP, O'Reilly Media, Sebastopol, ISBN: 978-1-449-33967-8)

Ispitivanjem poruka koje se prenose uspešno se nalazi problem u komunikaciji. Na primer, otvaranjem alarm poruka može se naći neusklađenost i na taj način detektovati mogući problem. Iako telefoni rade sasvim dobro, moguće da konfiguracija nije optimalna (Slika 22).

```

Skinny Client Control Protocol
Data length: 96
Header version: Basic (0x00000000)
Message ID: AlarmMessage (0x00000020)
Alarm severity: warning (1)
Display message: 3: Name=SEP0013C4615F16 Load=7.2(2.0)Version Error
Alarm param 1: 0x00000900
Alarm param 2: 192.168.1.1 (192.168.1.1)

```

Slika 22 Skinny Alarm poruka

Na slici je screenshot opcijom uhvaćeno polje iz Wireshark programa, gde se može videti Skinny Alarm poruka, uokvirena crvenim markerom. (Preuzeto sa Hartpence B., (2013), Packet guide to Voice over IP, O'Reilly Media, Sebastopol, ISBN: 978-1-449-33967-8)

### SIP PROTOKOL

Sve veća potražnja među komunikacionim pružaocima usluga, njihovim partnerima i pretplatnicima (potrošačima) za novom generacijom IP baziranih servisa se danas rešava pomoću SIP. SIP, koji je kao ideja rođen u računarskim laboratorijama pre manje od jedne decenije, predstavlja prvi protokol koji omogućava multikorisničke sesije nezavisno od medijskog sadržaja.

SIP protokol je razvijen od strane IETF4 (engl. Internet Engineering Task Force) i standardizovan je u RFC 3261. Ovaj signalizacioni protokol je sada podržan od gotovo svih proizvođača VoIP tehnologija. Iako postoji značajan broj instaliranih sistema koji koriste H.323 i Skinny protokole, novije mreže koje se budu implementirale će najverovatnije usvojiti SIP standard u odnosu na druge.

Poput ostalih signalnih protokola, SIP se oslanja na RTP protokol prilikom prenosa glasovnih paketa između izvora i odredišta. Pored toga, RFC 3261 ukazuje da drugi protokoli mogu biti deo SIP raspoređivanja [57].

Namena SIP protokola je uspostavljanje, održavanje, modifikovanje i raskid multimedijalnih sesija. Učesnici u sesijama mogu biti ljudi ili računarski mehanizmi (računarski serveri koji služe npr. slanju određenog podatka korisniku na koji se on prethodno pretplatio).

SIP poziv koristi se za kreiranje sesije i definisanje njenih parametara. Ovi parametri omogućavaju učesnicima prilagođavanje tipu medija koji se u toj sesiji koristi. Mobilnost korisnika podržana je kroz posredne (engl. proxy) i preusmerivačke (engl. redirect) servere, preusmeravanjem poziva na trenutnu lokaciju korisnika. Korisnici mogu na jednostavan način da registruju svoje nove lokacije, koje se beleže na SIP serverima.

Protokol je koncipiran nezavisno o transportnom mediju pa se može jednostavno implementirati na bilo kojoj vrsti mreže. Ipak isključivo se koristi na IP protokolu, a u transportnom sloju može koristiti ravnopravno TCP (engl. Transmission Control Protocol) i UDP (engl. User Datagram Protocol) protokole premda češće koristi UDP radi njegovih prednosti kod transporta. Nedostatak TSR protokola je to što zahteva potvrdu o prijemu paketa dok UDP to ne zahteva i time omogućava razgovor u realnom vremenu .

SIP radi na aplikacionom sloju u cilju iniciranja korisničkih sesija za multimedijalne prenose, kao što su glas, video čet, igranje, i virtuelne stvarnosti. Po RFC, sesija je razmena podataka između učesnika. SIP podržava određivanje mapa i usluge preusmeravanja, funkcije koje omogućavaju korisnicima da se realizuje prenos sa različitih lokacija. To znači da kada se VoIP čvorovi povezuju jedan sa drugim, mora da postoji mehanizam za uspostavljanje komunikacije i uspostavljanje nekih pravila. SIP i SDP (engl. Session Description Protocol) protokoli će se pobrinuti za to.

Kao što podržava sličan skup funkcija i kodeke, SIP se ponekad kombinuje zajedno sa N.323 ili Skinny protokolima da bi opslužili novije aplikacije ili funkcije. Često se kaže da je SIP mnogo lakše čitati i koristiti od drugih nezaštićenih signalizacionih protokola. Ovo može biti zbog toga što je SIP veoma sličan po strukturi sa Hypertext Transfer Protocol, ili HTTP [58].

SIP u svom radu koristi različite protokole:

- RSVP (engl. Resource reSerVation Protocol) koji se koristi za rezervaciju mrežnih resursa i pomaže za ostvarivanje određenog nivoa kvaliteta usluge,
- RTP (engl. Real Time Protocol)/RTCP (engl. Real Time Control Protocol)/RTSP (engl. Real Time Streaming Protocol) – protokoli aplikacionog nivoa koji služe za slanje podataka u realnom vremenu,
- SAP (engl. Session Announcement Protocol) – protokol za objavljivanje multimedijalnih sesija,
- SDP (engl. Session Description Protocol) – protokol za opis multimedijalnih sesija.

SIP poruke su vrlo slične HTTP-u (RFC 2068). Mnoge sintakse, zaglavlja poruka kao i HTTP kodovi su ponovo iskorišćeni na isti način. Na primer, kod greške u SIP-u za „Adresa nije pronađena – 404“ je identična onoj kod Web-a. SIP takođe koristi puno toga od adresnih šema iz SMTP-a.

SIP omogućava sledeće osnovne zahteve u komunikacijama:

- 1) Servis za lokaciju korisnika,
- 2) uspostavljanje sesije,
- 3) upravljanje učesnicima sesije,
- 4) delimično uspostavljanje osobina.

Važna osobina SIP-a je da on ne definiše tip sesije koja se uspostavlja, već samo na koji će način ona biti upravljana. Ova fleksibilnost praktično znači da SIP može biti upotrebljen za brojne aplikacije i servise kao što su: interaktivno igranje video igara, muzika i video na zahtev ili video i Web konferencije [59].

Osobine SIP-a koje ga razlikuju od ostalih novih protokola za signalizaciju su:

- SIP poruke su tekstualnog tipa i kao takve lake su za čitanje, analizu i otklanjanje grešaka. Programiranje novih servisa je lakše i intuitivnije za dizajnere;
- SIP koristi MIME tip deskripcije na isti način kao kod e-mail klijenata tako da aplikacije koje su vezane za sesije mogu da budu startovane automatski;
- SIP koristi nekoliko već postojećih internet servisa i protokola kao što su DNS, RTP i RSVP. Nije potrebno uvoditi nove servise da bi se podržala SIP infrastruktura jer većina toga već postoji i dostupno je na tržištu;
- SIP proširenja se lako definišu, što omogućava pružaocima usluga da dodaju nove aplikacije bez oštećenja njihovih mrežnih infrastruktura. Starija, već postojeća SIP-bazirana oprema neće ometati nove SIP servise. Na primer, starija SIP implementacija koja ne podržava određeni metod ili zaglavlje koju donosi nova SIP

aplikacija, će jednostavno ignorisati ono što ne prepoznaje bez nekih većih problema u radu;

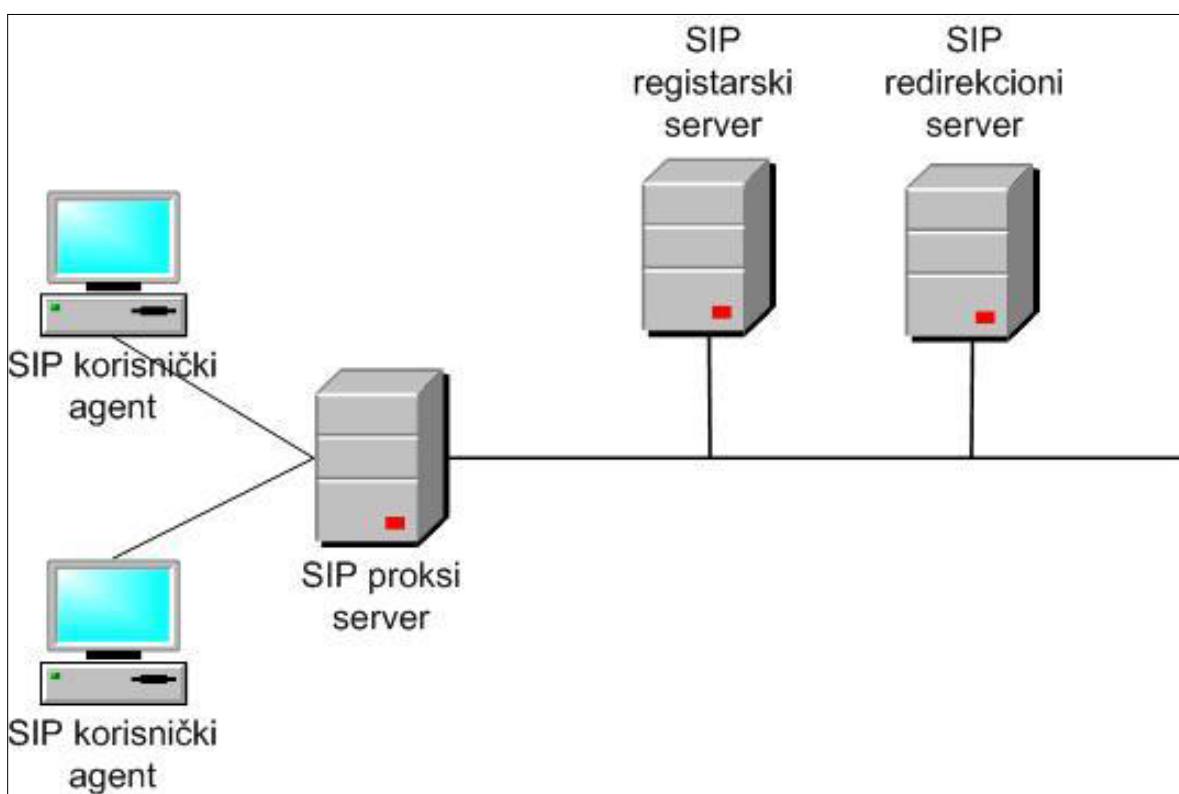
- SIP je nezavisan u pogledu transportnog sloja;
- SIP podržava ujednačavanje osobina različitih uređaja kao i posredovanje. Ako neki servis ili sesija inicira video i glas, glas se i dalje može prenositi uređajima koji nisu u stanju ujedno prenositi i video. U isto vreme, drugi uređaji se mogu koristiti za prenos jednosmernog video striminga.

SIP protokol je zasnovan na klijent-server principu koji je veoma popularan pristup u kreiranju aplikacija na Internetu. Klijent šalje zahtev serveru na koji, potom, server šalje odgovor klijentu. U slučaju SIP protokola, na klijentov SIP zahtev, server šalje jedan ili više SIP odgovora. Većina entiteta koji koriste SIP signalizaciju ima potrebu i da šalje i da prima zahteve, odnosno odgovore, stoga većina entiteta može da bude i klijent i server. Koja je trenutna uloga entiteta (klijent ili server) zavisi od trenutne komunikacije – da li trenutno šalje zahtev ili odgovor. Komponente koje koriste (razmenjuju) SIP signalizaciju su:

- Korisnički agent (engl. UA – User Agent) – Korisnički agent, ustvari, predstavlja terminal (krajnji uređaj) poput IP telefona (koji se često označava terminom SIP telefon da se naglasi da radi sa SIP signalizacijom), video telefona ili računara sa instaliranim odgovarajućim aplikacijama. Korisnički agent u sebi sadrži i klijent (engl. UAC - User Agent Client) i server (engl. UAS – User Agent Server) implementaciju, pošto može da se u komunikaciji ponaša i kao klijent (šalje zahteve) i kao server (šalje odgovore na primljene zahteve) [60].
- Proksi (engl. Proxy) – Srednji komponenta koja prosleđuje zahteve od klijenta (UAC) prema serveru (UAS) ili drugom proksiju. Ovo se radi pre svega za politiku usmeravanja, ali može da se koristi i npr. za proveru identiteta. Primer standardnog rasporeda je veb proksi. Klijenti pošalje zahteve na proksi, koji ih zatim prosleđuje veb serveru. Tako klijenti nikada zapravo ne komuniciraju sa veb serverom.
- Preusmeravački server (engl. Redirect Server) – Ovaj server vrši preusmeravanje SIP zahteva na sledeći način. On kao i proksi server vrši pretragu lokacijskog servera ili DNS servera, da bi našao adresu (lokaciju) traženog korisničkog agenta ili proksi servera i potom vraća odgovor izvoru SIP zahteva u kome se navodi adresa entiteta kome treba poslati dotični SIP zahtev (koji je primio i na njega odgovorio preusmeravački server). Razlika u odnosu na proksi server je u tome što preusmeravački server ne vrši prosleđivanje SIP zahteva/odgovora, već samo vraća adresu entiteta na koju treba poslati SIP zahtev. Odgovor koji vraća preusmeravački server je iz grupe 3xx odgovora.

- Registracioni server (engl. Registrar Server) – Ovaj server se još naziva i registrar. Korisnici se mogu registrovati na ovaj server sa ciljem da se pribeleži njihova lokacija tj. IP adresa koja se vezuje za njihov URI naziv (ili neki drugi alias). Korisnik može prilikom registracije prijaviti i više svojih lokacija. Registracioni server može da zahteva autentifikaciju korisnika pre prihvatanja njihove registracije. Registracioni server na osnovu primljenih registracija ažurira lokacijski server koji čuva adrese (lokacije) SIP korisnika.

Arhitektura SIP sistema je data na sledećoj slici (slika 23).



Slika 23 Arhitektura SIP sistema

Umesto IP adresom, određite u protokolu SIP može biti predstavljeno URI -em koji ima isti format kao i e-adresa i u skladu s tim ispravna SIP adresa može biti sip:naziv@server1.com. Korišćenje URI -a implicira upotrebu sistema imena domena (engl. DNS – Domain Name Sistem), kako bi se imena čvora (host) i domena mogla mapirati u IP adrese. Povezanost protokola SIP i DNS-a omogućava interoperabilnost telefonskih sistema i mehanizma adresiranja [61].

SIP razgovori mogu početi tako što ćete kontaktirati IP adresu ili korisničko ime u cilju da UAC (klijent) komunicira sa drugim korisnikom ili resursom na mreži. Standardni SIP

adresiranje je sličan e-majlu, uzimajući jedan od sledećih oblika (port (engl. port) je opciono, iako to nije navedeno, 5060 se koristi):

- sip:user@domain:port
- sip:user@host:port
- sip:phone\_number@domain

Još jedan primer SIP adresiranja je prikazan na slici 24. Format je URI (engl. *Uniform Resource Identifier*). Po RFC, vaša SIP adresa ili identitet je vaš SIP URI.

```
Internet Protocol Version 4, Src: 172.30.1.11 (172.30.1.11), Dst: 172.30.1.1 (172.30.1.1)
User Datagram Protocol, Src Port: sip (5060), Dst Port: sip (5060)
Session Initiation Protocol
Request-Line: INVITE sip:4752222@172.30.1.1 SIP/2.0
Message Header
Message Body
```

Slika 24 SIP adresiranje

Osim SIP URI-a podržan je i SIPS URI, koji podrazumeva primenu sigurnosnog mehanizma. Poziv prema SIPS URI-u garantuje da je siguran, kriptografski zaštićen transport (engl. TLS – Transport Layer Security) korišćen za prenos svih SIP poruka od onoga ko zove do domena pozvanog. SIP URI i SIPS URI, koji koristi port 5061, moraju sadržati ime čvora pa mogu sadržati i ime korisnika i broj portova.

Komunikacija pomoću SIP protokola, koja se često naziva i signalizacija, sastoji se od niza poruka (engl. messages). Uobičajeno je da se svaka poruka prenosi u posebnom UDP datagramu. Svaka se poruka sastoji od „početne linije“, zaglavlja poruke i tela poruke. Početna linija označava vrstu poruke. Postoje dve vrste poruka – zahtevi (engl. request) i odgovori (engl. response). Zahtevi se obično koriste za iniciranje neke akcije ili za obaveštavanje pozvane strane zahteva o nečemu. Odgovori se koriste za potvrđivanje da je zahtev primljen i obrađen pa sadrže i status obrade.

SIP je opširno opisan u RFC, delom zbog broja poruka koje definiše. Poput ostalih signalnih protokola, SIP ima poruke koje su specifične po svojim operacijama, iako mnoge od njih dele slične polja. SIP RFC takođe koristi termin METHOD. METHOD je funkcija, a ove funkcije su smeštene u tip poruke. SIP čvorovi imaju pravila za svaki metod, mada neka pravila su zajednički za sve njih. Na primer, svi SIP zahtevi moraju sadržati polja za To, From, CSeq, Call-ID, Max-Forwards, i Via [62].

Jedan od razloga zašto SIP ima poznato zaglavlje je to što, po RFC 3261, SIP poruke prate opšti format za određenu Internet poruku kao što je opisano u RFC 5322. Ovo važi i za zahteve i za odgovore. RFC 5322 je fokusiran na tekstualno-baziranu komunikaciju, i tako je SIP proširen i stvarima koje nisu navedene u RFC 5322.

Postoji više vrsta SIP zahteva i svaki od njih ima određene osobine. Najvažniji zahtevi su:

- INVITE – ova poruka se koristi za uspostavljanje sesije. Takođe se sa njom mogu menjati parametri poziva (re-INVITE).
- ACK – poruka kojom se potvrđuje prijem završnog odgovora na INVITE.
- BYE – koristi se da bi se prekinula sesija. Strana koja želi da prekine sesiju šalje BYE drugoj strani.
- CANCEL – koristi se za prekid još ne uspostavljenih sesija. Ukoliko pozvani korisnik nije odgovorio, a korisnik koji je inicirao poziv želi da prekine poziv (tipično kada pozvani korisnik ne odgovara neko vreme).
- REGISTER – svrha REGISTER zahteva je da se registraru dostavi trenutna lokacija korisnika. Preko REGISTER poruka se prenose informacije o trenutnoj IP adresi i portu na kome se može doći do korisnika.
- OPTIONS – koristi se da bi se saznale mogućnosti druge strane u sesiji.
- INFO – služi za slanje informacija u toku sesije koje ne menjaju stanje sesije.

Kod SIP poruke tipa zahtev, početna linija se zapravo zove linija zahteva. Linija zahteva sadrži SIP URI i verziju i završava sa CR LF karakterima, kao što je prikazano na slici 25. Na slici, linija zahteva je istaknuta.

```

Internet Protocol Version 4, Src: 172.30.1.11 (172.30.1.11), Dst: 172.30.1.1
User Datagram Protocol, Src Port: sip (5060), Dst Port: sip (5060)
Session Initiation Protocol
Request-Line: REGISTER sip:172.30.1.1 SIP/2.0
Message Header
000 00 11 d8 d6 05 9b 00 09 6e 05 cb 11 08 00 45 00 ..... n.....E.
010 02 3c 00 5f 00 00 40 11 1e 0a ac 1e 01 0b ac 1e .<._..@. ....
020 01 01 13 c4 13 c4 02 28 5f 0f 52 45 47 49 53 54 .....( _REGIST
030 45 52 20 73 69 70 3a 31 37 32 2e 33 30 2e 31 2e ER sip:172.30.1.
040 31 20 53 49 50 2f 32 2e 30 0d 0a 56 69 61 3a 20 1 SIP/2.0..v a:
050 53 49 50 2f 32 2e 30 2f 55 44 50 20 31 37 32 2e SIP/2.0/ 66 172.
060 33 30 2e 31 2e 31 31 3a 35 30 36 30 3b 62 72 61 30.1.11: 5060;bra
070 6e 63 68 3d 7a 39 68 47 34 62 4b 33 31 65 37 65 nch=z9hg 4bk31e7e
080 31 63 66 61 01 00 41 61 78 21 46 66 73 77 61 73 1 f M...

```

Slika 25 SIP linija zahteva prikazana u heksadecimalnom zapisu

SIP odgovori poput ovog prikazanog na slici 26 imaju statusnu liniju na početku i statusni kod za transakcije. U ovom slučaju, server (172.30.1.1) vraća statusni kod 200 prema VoIP telefonu, koji ukazuje na uspeh.

```

Internet Protocol Version 4, Src: 172.30.1.1 (172.30.1.1), Dst: 172.30.1.11 (172.30.1.11)
User Datagram Protocol, Src Port: sip (5060), Dst Port: sip (5060)
Session Initiation Protocol
  Status-Line: SIP/2.0 200 OK
  Message Header
    Via: SIP/2.0/UDP 172.30.1.11:5060;branch=z9hG4bK31e7e1cfa;received=172.30.1.11
    From: 4751111 <sip:4751111@172.30.1.1>;tag=afec0e2ffae8169
    To: 4751111 <sip:4751111@172.30.1.1>;tag=as10e6b586
    Call-ID: dd5d67821535446081b1eece5c33e606@172.30.1.11
    CSeq: 87753542 REGISTER
    Server: FPBX-2.10.0rc1(1.8.11)
    Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, SUBSCRIBE, NOTIFY, INFO, PUBLISH
    Supported: replaces, timer
    Expires: 3600
    Contact: <sip:4751111@172.30.1.11:5060>;expires=3600
    Date: Wed, 15 Aug 2012 16:10:31 GMT
    Content-Length: 0

```

Slika 26 SIP odgovor na registracioni zahtev

Kad korisnički agenti ili proksi server prime zahtev, šalju odgovor. Na svaki zahtev mora se odgovoriti. Izuzetak su ACK zahtevi na koje ne treba odgovarati. Kôd odgovora je celi broj od 100 do 699 i ukazuje na vrstu odgovora. Većina odgovora je definisana u okviru osnovne preporuke za SIP protokol (RFC 3261), a ostali su definisani u naknadnim RFC preporukama koje proširuju osnovnu SIP preporuku (RFC 3261). Postoji 6 grupa odgovora:

- **1xx** – Privremen (engl. Provisional), su odgovori kojima se signalizira da je zahtev prihvaćen i da je u toku njegovo procesiranje. Odgovori iz ove grupe su:
  - o 100 (Trying) – Ovim odgovorom se signalizira primanje i procesiranje zahteva. Koristi se da bi se izbegle nepotrebne retransmisije zahteva (ako se dovoljno dugo ne dobije odgovor dolazi do retransmisije zahteva)
  - o 180 (Ringing) – Ovim odgovorom se signalizira da se traženi korisnik (koji je primio INVITE zahtev) obaveštava o pozivu („zvoni“ mu telefon).
  - o 181 (Call Is Being Forwarded) – Ovim odgovorom se signalizira da je poziv prosleđen na jednu ili više drugih destinacija.
  - o 182 (Queued) – Ovim odgovorom se signalizira da je zahtev primljen, ali trenutno ne može da se opsluži. Međutim, zahtev nije odbijen, već stavljen u red za čekanje. Fraza odgovora može da sadrži informaciju o broju zahteva koji se nalaze u redu za čekanje i procenu vremena čekanja. Na primer, 3 calls queued; expected waiting time is 5 minutes (3 poziva na čekanju; očekivano vreme čekanja je 5 minuta).
  - o 183 (Session Progress) – Ovim odgovorom se signalizira da je procesiranje u toku, pri čemu nijedan od prethodno opisanih 1xx odgovora nije adekvatan, pa se može reći da odgovor 183 predstavlja sve slučajeve koji nisu pokriveni preostalim

1xx odgovorima. Informacije o progresu procesiranja mogu da se navedu u frazi odgovora, telu poruke ili u poljima zaglavlja poruke.

- **2xx** – Uspešan (engl. Success), su pozitivni odgovori kojima se signalizira uspešna obrada zahteva (zahtev je primljen, razumljiv i prihvaćen). Odgovori iz ove grupe su:
  - o 200 (OK) – Ovim odgovorom se signalizira uspešna obrada zahteva. Informacija koja se nalazi u telu poruke zavisi od tipa zahteva na koji se odgovara. Na primer, u slučaju odgovora na INVITE zahtev, telo poruke će sadržati parametre prenosa prijemnika (mogućnosti prijemnika). Telo poruke neće postojati ako se ovaj odgovor šalje na zahteve CANCEL, INFO, MESSAGE, SUBSCRIBE, NOTIFY, PRACK.
  - o 202 (Accepted) – Ovaj odgovor je originalno zamišljen kao odgovor koji bi se koristio u sprezi sa SUBSCRIBE i NOTIFY zahtevima, a kasnije se definisala sprega i sa drugim metodama poput REFER. Međutim, u praksi se pokazalo da se njegova upotrebna vrednost ne razlikuje mnogo od one koja bi se postigla odgovorom 200, a čak su i primećene ne kompatibilnosti između različitih implementacija. Otuda je preporuka da se ovaj odgovor više ne generiše, a ako se primi (od starijih implementacija), treba da se tretira kao odgovor 200. Ovaj odgovor je originalno definisan u RFC 3265 koji je potom zamenjen sa RFC 6665 u kome je preporučeno da se ne treba više generisati odgovor 202.
- **3xx** – Preusmeren (engl. Redirection), su odgovori kojima se signalizira nova lokacija traženog korisnika ili alternativni servisi kojima bi se mogao alternativno ispuniti zahtev korisnika (na primer, mejl servis ili govorna pošta). Odgovori iz ove grupe su:
  - o 300 (Multiple Choices) – Za korisnika je vezano više lokacija, pa se ovim odgovorom vraća lista lokacija da pozivajući korisnik sam izabere adresu (lokaciju) koju će pozvati da bi došao do traženog korisnika. Naravno, ovaj odgovor se vraća ako proksi server ne želi sam da izabere lokaciju traženog korisnika koju će nazvati (jer proksi server nije tako programiran).
  - o 301 (Moved Permanently) – Traženi korisnik se više ne nalazi na adresi navedenoj u URI odredišta koja se nalazi u startnoj liniji zahteva, i u odgovoru se vraća nova adresa korisnika koju treba koristiti u budućim komunikacijama sa njim.

- o 302 (Moved Temporarily) – Traženi korisnik se ne nalazi na adresi navedenoj u URI odredišta koja se nalazi u početnoj liniji zahteva, i u odgovoru se vraća nova (privremena) adresa korisnika koju treba koristiti u komunikaciji sa njim. U ovom odgovoru se može nalaziti vrednost trajanja privremene adrese i dok ne istekne to vreme ova privremena adresa se može koristiti u budućim komunikacijama sa tim traženim korisnikom. Ako vrednost trajanja nije navedena, onda se privremena adresa može samo trenutno (odmah) iskoristiti.
- o 305 (Use Proxy) – Poziv mora da ide kroz proksi server naveden u odgovoru, pa je stoga pozivajući korisnik dužan da ponovo pošalje zahtev, ali sada preko navedenog proksi servera.
- o 380 (Alternative Service) – Poziv nije mogao da se uspešno uspostavi, ali ovim odgovorom se nude alternativni servisi (koje nudi traženi korisnik) koji mogu da se iskoriste za (svojevrni) kontakt sa traženim korisnikom, poput govorne pošte. Alternativni servisi se navode u telu poruke.
- **4xx** – Greška klijenta (engl. Client Error), su odgovori koji signaliziraju greške na klijent strani (koja je i slala zahtev). Na primer, klijentov zahtev sadrži sintaksne greške ili klijentov zahtev traži od servera nešto što server ne može da uradi. Odgovor iz ove grupe definitivno odbija primljeni zahtev. Zahtev se ne sme ponovo slati istom serveru, a da prethodno nije modifikovan tako da se koriguje greška koja je signalizirana odgovorom iz ove grupe. Odgovori ove grupe su:
  - o 400 (Bad Request) – Ovim odgovorom se signalizira greška u sintaksi zahteva. Na primer, ne postoji prazna linija. Fraza odgovora bi trebala da sadrži detalje o detektovanoj sintaksoj grešci.
  - o 401 (Unauthorized) – Ovaj odgovor signalizira da korisnik (klijent) mora da se autentifikuje da bi se zahtev mogao izvršiti. Ovaj odgovor šalje server ili registrar, dok u identičnoj situaciji proksi server šalje odgovor 407.
  - o 402 (Payment Required) – Ovaj odgovor je rezervisan za buduću upotrebu i očigledno je predviđen za situacije u kojoj servis mora da se plati.
  - o 403 (Forbidden) – Server je primio i razumeo zahtev, ali odbija da ga izvrši.
  - o 404 (Not Found) – Ovaj odgovor signalizira da traženi korisnik sigurno ne postoji u domenu koji je naveden u URI odredišta startne linije zahteva ili da sam domen ne postoji.
  - o 405 (Method Not Allowed) – Server je primio i razumeo zahtev, ali dotični zahtev nije podržan. Na primer, korisnički agent je primio REGISTER zahtev. U odgovoru se nalazi i lista metoda koje server podržava.

- o 406 (Not Acceptable) – Ovim odgovorom server signalizira klijentu da procesiranje njegovog zahteva ne može da se izvrši jer ne podržava opcije navedene u Accept polju zaglavlja.
- o 407 (Proxy Authentication Required) – Odgovor ekvivalentan odgovoru 401, samo ga u ovom slučaju šalje proksi server (klijent mora prvo da se autentifikuje kod proksi servera).
- o 408 (Request Timeout) – Server nije mogao da izvrši zahtev u određenom vremenskom okviru. Klijent može kasnije da ponovi ovaj zahtev u neizmenjenom obliku.
- o 410 (Gone) – Traženi korisnik se više ne nalazi u navedenom domenu (koji je naveden u okviru URI odredišta), pri čemu se ne zna njegova nova lokacija. Server je siguran da je ovo stanje permanentno (korisnik sigurno nije više u tom domenu), u suprotnom bi server poslao 404 odgovor.
- o 413 (Request Entity Too Large) – Ovim odgovorom se signalizira da je telo poruke unutar primljenog zahteva predugačko i da server stoga neće da ga procesira.
- o 414 (Request-URI Too Long) – Ovim odgovorom se signalizira da je URI odredišta predugačak i da ga stoga server ne može procesirati.
- o 415 (Unsupported Media Type) – Ovim odgovorom se signalizira da je telo poruke zahteva u formatu koji server ne može da procesira. U okviru ovog odgovora se vraća lista podržanih formata upotrebom polja zaglavlja Accept, Accept-Encoding, ili Accept-Language u zavisnosti koji problem u formatu je detektovao server.
- o 416 (Unsupported URI Scheme) – Ovim odgovorom se signalizira da format korišćen za URI identifikator nije podržan od strane servera. Na primer, u zahtevu se koristio format telefona (Telephone URI) koji server ne podržava.
- o 420 (Bad Extension) – Ovim odgovorom se signalizira da server nije prepoznao ekstenziju protokola navedenu u polju zaglavlja Proxy-Require ili Require primljenog zahteva. U ovom odgovoru server navodi listu ne podržanih ekstenzija (one se stavljaju u polje zaglavlja Unsupported).
- o 421 (Extension Required) – Ovim odgovorom se signalizira da u zahtevu nije navedena određena ekstenzija protokola (u polju zaglavlja zahteva Supported), a bez koje server ne može da procesira zahtev. Odgovor vraća listu zahtevanih ekstenzija u okviru polja zaglavlja Required. Pre nego što se ovaj odgovor vrati, server mora da pokuša da procesira zahtev koristeći difolt podešavanja, kao i sve ekstenzije navedene u Supported polju zaglavlja zahteva, a koje server podržava.

- o 480 (Temporarily Unavailable) – Ovim odgovorom se signalizira da je zahtev uspešno primljen, ali da je traženi korisnik trenutno nedostupan (na primer, uključio je uslugu „ne smetaj“ kojom se onemogućava prijem poziva od drugih, pri čemu sam taj korisnik i dalje može da poziva druge).
- o 486 (Busy Here) – Ovim odgovorom traženi korisnik signalizira da ne želi ili ne može da prihvati zahtev za uspostavom veze.
- o 487 (Request Terminated) – Ovim odgovorom se signalizira nasilni prekid procesiranja zahteva usled prijema CANCEL ili BYE zahteva od samog klijenta.
- **5xx** – Greška servera (engl. Server Error), su odgovori koji signaliziraju greške na serverskoj strani (koja je primila zahtev). Serverska strana ovim signalizira da iz određenog razloga nije mogla da ispuni validan klijentov zahtev. Odgovor iz ove grupe definitivno odbija primljeni zahtev. Odgovori iz ove grupe su:
  - o 500 (Server Internal Error) – Ovim odgovorom se signalizira da je došlo do neočekivane greške u obradi zahteva i da zato nije došlo do ispunjenja zahteva.
  - o 501 (Not Implemented) – Ovim odgovorom se signalizira da tip zahteva (metoda) nije prepoznat i stoga ne može da se procesira jer nije implementirana funkcionalnost obrade dotičnog tipa zahteva.
  - o 502 (Bad Gateway) – Ovim odgovorom se signalizira da gejtvej nije uspeo uspešno da procesira zahtev jer je primio negativan odgovor od druge mreže sa kojom gejtvej povezuje SIP mrežu (došlo je do greške u drugoj mreži koja sprečava uspešno procesiranje zahteva).
  - o 503 (Service Unavailable) – Ovim odgovorom se signalizira privremena nedostupnost servera usled preopterećenja ili održavanja. Korisnik može da ponovi zahtev nakon vremena specificiranog u zaglavlju poruke (Retry-After polje). Ako ovo vreme nije specificirano, klijent tretira ovaj odgovor kao odgovor 500.
  - o 504 (Server Time-Out) – Ovim odgovorom se signalizira da je server koji je primio zahtev tražio uslugu od nekog drugog servera prilikom procesiranja zahteva, ali da je vreme čekanja na odgovor od tog drugog servera isteklo pa stoga server koji je originalno primio zahtev ne može da ga uspešno obradi. Na primer, ako se tražila usluga od DNS servera radi određivanja proksi servera u domenu u kome se nalazi traženi korisnik.
  - o 505 (Version Not Supported) – Server ne podržava verziju SIP protokola navedenu u početnoj liniji zahteva i samim tim ne može da obradi primljeni zahtev.

- o 513 (Message Too Large) – Server ne može da obradi primljenu poruku jer je predugačka.
- **6xx** – Globalni neuspeh (engl. Global Failure), su odgovori koji signaliziraju generalne greške. Ovi odgovori signaliziraju da nijedna serverska strana ne može da ispuni klijentov zahtev. Odgovor iz ove grupe definitivno odbija primljeni zahtev. Odgovori iz ove grupe su:
  - o 600 (Busy Everywhere) – Ovim odgovorom se signalizira da je zahtev uspešno stigao na odredište, ali da je korisnik zauzet na svim lokacijama koje su vezane za njega.
  - o 603 (Decline) – Ovaj odgovor ima isti efekat kao odgovor 600. Razlika je u tome što se ovde ne navodi razlog odbijanja veze (da li je u pitanju zauzetost korisnika ili korisnik jednostavno ne želi da prihvati poziv).
  - o 604 (Does Not Exist Anywhere) – Ovim odgovorom se signalizira da korisnik naveden u URI odredišta sigurno ne postoji.
  - o 606 (Not Acceptable) – Traženi korisnik je uspešno primio zahtev, ali ovim odgovorom signalizira da ne može da prihvati vezu jer parametri veze koji su mu ponuđeni nisu za njega prihvatljivi. Korisnik ovim odgovorom signalizira da je želeo da prihvati vezu, ali da nije mogao iz tehničkih razloga.

Polja zaglavlja SIP poruka su definisana kao polja zaglavlja HTTP poruka. Polje zaglavlja se definiše u formatu:

naziv polja: vrednost polja

pri čemu se za vrednost polja može navesti više vrednosti koje su međusobno razdvojene zarezima. Svako polje zaglavlja mora da se završi CRLF karakterom. Takođe, iza vrednosti polja mogu da se stave parametri i njihove vrednosti u formatu:

naziv parametra = vrednost parametra

U slučaju upotrebe parametara u polju zaglavlja, iza vrednost polja mora da stoji tačka-zarez (;). Parametri su međusobno razdvojeni tačka-zarezom (;). Isti parametar ne sme da se javi više od jednom u okviru iste vrednosti polja.

Pojedina polja mogu da se pojave samo u zahtevima, a pojedina samo u odgovorima. Takođe, pojedina polja mogu da jave samo u određenim tipovima zahteva ili odgovora. Svaki tip zahteva/odgovora ima listu obaveznih polja zaglavlja koja obavezno moraju da budu prisutna u zaglavlju dotične poruke.

Većina polja zaglavlja je definisana u okviru osnovne preporuke za SIP protokol (RFC 3261), a ostala su definisana u naknadnim RFC preporukama koje proširuju osnovnu SIP preporuku (RFC 3261) 4.

Poslednji korak neophodan pre nego telefon može napraviti poziv je da se registruje sa serverom poziva. Ovo se postiže preko SIP poruke: registrar. Ova poruka je deo razmene sa PBX-om, kao što je prikazano na slici 27.

110	GiantEle_05:cb:11	Broadcast	ARP	who has 172.30.1.1? Tell 172.30.1.11
111	AsustekC_d6:05:9b	GiantEle_05:cb:11	ARP	172.30.1.1 is at 00:11:d8:d6:05:9b
112	172.30.1.11	172.30.1.1	SIP	Request: REGISTER sip:172.30.1.1
113	172.30.1.1	172.30.1.11	SIP	Request: OPTIONS sip:4751111@172.30.1.11:5060
114	172.30.1.1	172.30.1.11	SIP	Status: 200 OK (1 bindings)
115	172.30.1.11	172.30.1.1	SIP	Status: 200 OK

Slika 27 Razmena registracije

Na slici 28 prikazano je poređenje paketa 112 i 114 sa slike 7.

Internet Protocol Version 4, Src: 172.30.1.11 (172.30.1.11), Dst: 172.30.1.1 (172.30.1.1) User Datagram Protocol, Src Port: sip (5060), Dst Port: sip (5060) Session Initiation Protocol Request-Line: REGISTER sip:172.30.1.1 SIP/2.0 Message Header Via: SIP/2.0/UDP 172.30.1.11:5060;branch=z9hG4bK31e7e1cfa Max-Forwards: 70 Content-Length: 0 To: 4751111 <sip:4751111@172.30.1.1> From: 4751111 <sip:4751111@172.30.1.1>;tag=afec0e2ffae8169 Call-ID: dd5d67821535446081b1eece5c33e606@172.30.1.11 CSeq: 87753542 REGISTER
Internet Protocol Version 4, Src: 172.30.1.1 (172.30.1.1), Dst: 172.30.1.11 (172.30.1.11) User Datagram Protocol, Src Port: sip (5060), Dst Port: sip (5060) Session Initiation Protocol Status-Line: SIP/2.0 200 OK Message Header Via: SIP/2.0/UDP 172.30.1.11:5060;branch=z9hG4bK31e7e1cfa;received=172.30.1.11 From: 4751111 <sip:4751111@172.30.1.1>;tag=afec0e2ffae8169 To: 4751111 <sip:4751111@172.30.1.1>;tag=as10e6b586 Call-ID: dd5d67821535446081b1eece5c33e606@172.30.1.11 CSeq: 87753542 REGISTER

Slika 28 Zaglavlje – poređenje polja

Polja zaglavlja koja se pojavljuju na slici 8 imaju sledeća značenja:

- Via – Ovo polje snima putanju zahteva tako što svaki proksi server dodaje ovo polje i u njega upisuje svoju URI lokaciju. Polje se dodaje ispred ostalih Via polja ako ona postoje. Na osnovu ovog polja, koje se kopira u odgovor, se postiže da odgovor ide istom putanjom kao zahtev. Tada svaki proksi server skida svoje Via polje kako odgovor prođe kroz njega. Ovo polje koristi tzv. branch parametar u koji se stavlja heširana vrednost polja To, From, Call-ID, kao i vrednost URI odredišta iz startne linije zahteva. Naravno, pod heširanjem ovih vrednosti se misli na heširanje ovih vrednosti posmatranih zajedno. Parametar branch predstavlja identifikaciju transakcije i pomoću ovog parametra proksi serveri mogu detektovati eventualne petlje. Vrednost branch parametra mora početi vrednošću z9hG4bK koja predstavlja tzv. magični kolačić (engl. magic cookie). Na početku Via polja se nalazi naziv

protokola (SIP), verzija protokola (2.0) i identifikacija transportnog protokola (UDP, TCP, SCTP ili TLS, gde TLS predstavlja TLS preko TCP).

- From – Ovo polje sadrži URI identifikator izvorišta zahteva. U slučaju odgovora ovo polje predstavlja izvorište zahteva na koji se odnosi odgovor. U ovom polju se može koristiti tag parametar koji predstavlja jedinstveni identifikator dijaloga (u gotovo svim slučajevima se koristi tag parametar). Kao i u slučaju Contact polja, uz URI se može navesti i display name, i tada URI identifikator ide u < > zagrade. Jedinstvena identifikacija je zapravo kombinacija Call-ID i dva tag-a u „To“ i „From“ polja. Na primer, „To“ oznaka za oba registraciona zahteva (paket 112) i odgovarajući odgovor (paket 114) imaju vrednost afec0e2ffae8169 i isti odgovarajući Call-ID. Odgovor u paketu 114 pruža drugu polovinu označavanja (tag-ovanja). Oznake treba da budu jedinstvene, sa 32 slučajno odabrana bita, i ne sme se ponovo koristiti već korišćena kombinacija. RFC 3261 navodi da svi zahtevi i odgovora u dijalogu moraju imati istu vrednost. To je slučaj i u paketima prikazanim na slici 8.
- To – Ovo polje sadrži URI identifikator odredišta zahteva, tj. određuje primaoca zahteva. U slučaju odgovora ovo polje mora sadržati tag parametar. Ovaj parametar iz 200 odgovora se mora koristiti u nastavku sesije (dijaloga) kao identifikator dijaloga. Važno je napomenuti da se ovo polje ne koristi za rutiranje SIP poruke iako ime polja to sugeriše. Za rutiranje zahteva se koristi URI odredište iz startne linije zahteva, a rutiranje odgovora se vrši na osnovu Via polja.
- Cseq – Ovo polje predstavlja jedinstvenu identifikaciju transakcije unutar dijaloga. Time je omogućeno i da se razlikuju novi zahtevi od retransmitovanih zahteva. Ovo polje sadrži decimalnu vrednost i naziv metoda zahteva. Za svaki novi zahtev unutar istog dijaloga se decimalna vrednost tipično inkrementira (vrednost mora da se promeni, a promena se tipično sastoji u inkrementiranju za 1 u odnosu na prethodnu decimalnu vrednost). Jedino se kod slanja ACK i CANCEL ne inkrementira decimalna vrednost, već se stavlja vrednost iz INVITE zahteva na koji se ACK ili CANCEL odnose.

Možemo videti iz slika 8 da Cseq brojevi su isti. Međutim, sledeći paket (113) je različita transakcija, pa su vrednosti koje se koriste za ovaj paket su promenjeni.

Slika 29 pokazuje SIP OPTIONS poruku iz call servera prema VoIP telefonu.

```
Internet Protocol Version 4, Src: 172.30.1.1 (172.30.1.1), Dst: 172.30.1.11 (172.30.1.11)
User Datagram Protocol, Src Port: sip (5060), Dst Port: sip (5060)
Session Initiation Protocol
  Request-Line: OPTIONS sip:4751111@172.30.1.11:5060 SIP/2.0
  Message Header
    Via: SIP/2.0/UDP 172.30.1.1:5060;branch=z9hg4bk74fc46f8
    Max-Forwards: 70
    From: "Unknown" <sip:Unknown@172.30.1.1>;tag=as0c7400a4
    To: <sip:4751111@172.30.1.11:5060>
    Contact: <sip:Unknown@172.30.1.1:5060>
    Call-ID: 6769ae784bfb461948c8c8a0733eca09@172.30.1.1:5060
    CSeq: 102 OPTIONS
      Sequence Number: 102
      Method: OPTIONS
      User-Agent: FPBX-2.10.0rc1(1.8.11)
      Date: Wed, 15 Aug 2012 16:10:31 GMT
      Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, SUBSCRIBE, NOTIFY, INFO, PUBLISH
      Supported: replaces, timer
      Content-Length: 0
```

Slika 29 SIP OPTIONS poruka

Cseq ne sadrži samo redni broj (32-bitni neoznačeni celobrojni (engl. integer) broj manji od 231), već takođe i metod (OPTIONS).

Max-Forwards – Ovo polje definiše maksimalan broj hopova zahteva. Pod hopom se podrazumeva jedan prolazak kroz proksi server. Proksi server dekrementuje vrednost u ovom polju. Kada se detektuje da je vrednost ovoga polja 0 (u proksi serveru ili korisničkom agentu), vraća se odgovor 483.

Contact – Ovo polje sadrži URI identifikator čije tumačenje zavisi od tipa zahteva/odgovora. Na primer, u slučaju INVITE zahteva označava lokaciju na koju suprotna strana treba da šalje svoje (eventualne) buduće zahteve u dijalogu. U slučaju odgovora na INVITE označava lokaciju na koju suprotna strana treba da šalje svoje buduće zahteve u dijalogu čime se omogućava direktna razmena SIP signalizacije (bez učešća proksi servera) između korisnika u nastavku dijaloga. Može biti navedeno i više identifikatora. Uz identifikator može da stoji i tzv. display name. U slučaju da se koristi display name tada se URI identifikator stavlja u < > zagradu.

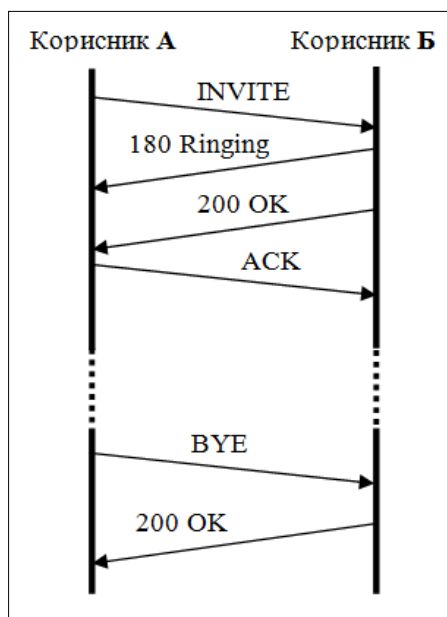
Ispitivanjem zaglavlja na slici 29, može se videti da postoji konzistentnost između polja u pogledu strukture URI. „Unknown“ je naveden kao korisničko ime, jer korisnik nije vezan za ovaj broj telefona; samo IP adresa je.

Allow – Ovo je spisak polja koje su podržane od strane poruka koje generiše korisnički agent. Kao što je ranije pomenuto, metode su funkcije koje obavlja SIP UAC ili UAS. Normalno, OPTIONS poruka se koristi za ovu svrhu. Operativno, SIP poruke mogu biti Allow polja kako bi

se smanjio ukupan broj potrebnih poruka. Slika 29 pokazuje da postoji širok spektar podržanih metoda, a samim tim i poruka.

Postoji još dosta polja koja mogu biti deo zaglavlja, neka mogu biti samo deo zahteva a neka samo deo odgovora, ali gore pomenuta su ona osnovna i mogu se najčešće sresti kada je u pitanju zaglavlje SIP poruke. Za pojedina polja se mogu koristiti i skraćeni nazivi 7 radi smanjenja veličine SIP poruke. Na primer, naziv polja Contact se može skraćeno napisati m. Ovo donekle narušava ideju da zaglavlje bude čitljivo, ali je korisna opcija ako je potrebno skratiti poruku tako da može da se prenese UDP protokolom.

U ovom delu će biti prikazano nekoliko primera koji opisuju uspostavu i prekid veze pomoću SIP signalizacije. Na slici 30 je prikazana direktna razmena SIP signalizacije između dva korisnika bez učešća proksi servera. Ovakav tip signalizacije se koristi u manjim mrežama sa malim brojem korisnika.



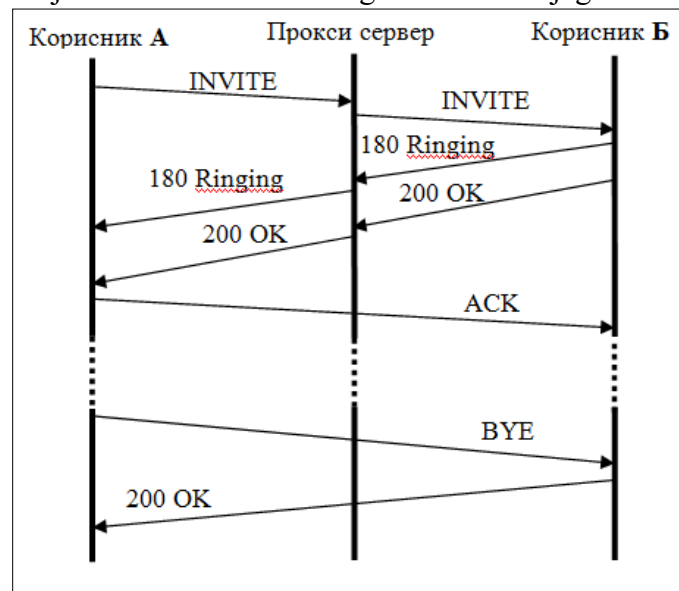
Slika 30 Direktna razmena SIP signalizacije

Korisnik A šalje INVITE zahtev korisniku B kojim signalizira da želi da uspostavi vezu sa njim. Korisnički agent korisnika B prvo vraća privremeni odgovor 180 (Ringing) kojim signalizira da se korisniku pušta „zvono“ tj. da se obaveštava o pozivu. Kada se korisnik B odazove, vraća se konačni odgovor 200 (OK) kojim se signalizira prihvatanje veze. Korisnik A potvrđuje uspostavu veze slanjem ACK zahteva čime je sesija uspostavljena i počinje razmena paketa koji nose govorne signale. Bilo koja od strana može da raskine vezu, a u primeru sa slike

30 vezu raskida korisnik A slanjem BYE zahteva. Korisnik B potvrđuje prihvatanje BYE zahteva slanjem odgovora 200 (OK). Sve SIP poruke sa slike 30 čine jedan dijalog.

Dijalog iz primera se sastoji od tri transakcije:

- 1) Prvu transakciju čine INVITE zahtev i odgovori na njega (180 i 200).
- 2) Drugu transakciju čini ACK zahtev. Na ACK zahtev se ne šalju odgovori.
- 3) Treću transakciju čine BYE zahtev i odgovor 200 na njega.



Slika 31 Razmena SIP signalizacije preko proksi servera

Na slici 31 je prikazan nešto složeniji primer u odnosu na sliku 30. U ovom slučaju korisnik A ne zna lokaciju korisnika B, već samo njegovu SIP URI identifikaciju. Korisnički agent korisnika A se obraća proksi serveru tako što mu šalje INVITE zahtev gde kao URI odredišta, navodi SIP URI korisnika B. Korisnički agent zna lokaciju tj. IP adresu proksi servera (na primer, ručno je konfigurisan sa tom informacijom, ili preko DHCP protokola mu je dostavljena ta informacija, itd). Proksi server prima INVITE zahtev, ispituje lokacijski server da odredi IP adresu korisnika A i potom mu šalje INVITE zahtev. U INVITE zahtev dodaje sebe u novom Via polju (koje ide iznad svih ostalih Via polja, u našem primeru samo Via polja kojeg je kreirao korisnik A) jer odgovor od korisnika A na INVITE zahtev korisnika B mora proći isti put kao INVITE zahtev samo u suprotnom smeru. Korisnik B prvo šalje odgovor 180, pa odgovor 200 da prihvata vezu i oba ova odgovora idu preko proksi servera (prilikom prolaska odgovora, proksi server skida iz njih Via polje koje je dodao – setimo se da se Via polja kopiraju iz zahteva u odgovor). Pošto je primio odgovore, korisnik B preko 200 odgovora tj. polja Contact (koje je obavezno u odgovoru 200) saznaje IP adresu korisnika A (ili direktno ako je stavljena IP adresa ispred naziva domena ili indirektno ako je stavljen naziv hosta jer tada treba da se obavi DNS upit). Pošto sada oba

korisnika iz polja Contact u INVITE zahtevu, odnosno u 200 odgovoru znaju IP adrese suprotne strane, preostale SIP poruke mogu da se razmenjuju direktno između korisnika A i korisnika B (ACK, BYE i odgovor 200 na BYE zahtev).

## **RTP I RTSP PROTOKOLI**

RTP (engl. Real-time Transport Protocol) osigurava podršku za prenos podataka u stvarnom vremenu (real-time). Usluge koje pruža RTP su:

- vremenska rekonstrukcija,
- otkrivanje izgubljenih paketa,
- sigurnost i identifikacija sadržaja.

RTP je primarno stvoren za multicast prenos real-time podataka, ali može se koristiti i za pojedinačni (unicast) prenos. RTP se nadopunjuje s RTCP kontrolnim protokolom kako bi dobio podatke o kvalitetu prenosa i o učesnicima u prenosu. Paketi poslani preko interneta imaju nepredvidivo kašnjenje zbog nesinhronizovanosti odašiljačke i prijemne strane. Real-time aplikacije zahtevaju vremenski sinhronizovano slanje i reprodukciju podataka. RTP omogućava vremensko označavanje, numeraciju paketa unutar niza. Vremensko označavanje (engl. timestamping) je najvažniji podatak aplikacije koji zahtevaju istovremenu komunikaciju. Pošiljaoc u to polje upisuje trenutak slanja prvog uzorka (npr. prvog audio uzorka ili slike). Vremenske oznake rastu s količinom vremena koju pokriva paket. Nakon prijema paketa, prijemnik koristi vremenske oznake kako bi pravilno rekonstruisao podatke. Vremenske oznake služe i za međusobnu sinhronizaciju različitih medija radi sinhronizacije video i audio signala, ali sam RTP nije zadužen za ovo, korekcija treba da se vrši na aplikacijskom sloju. UDP ne isporučuje pakete redosledom kojim su poslani pa se koristi numeracija paketa (engl. sequence numbers) kako bi se pristigli paketi pravilno rasporedili [63].

Pomoću numeracije paketa takođe se može otkriti i gubitak paketa. Identifikacija vrste tereta (engl. payload type identifier) određuje format tereta i koji su postupci kompresije i kodiranja korišćeni. Iz tog polja aplikacija na prijemnoj strani zna kako interpretirati i pravilno reprodukovati podatke. Osnovni tipovi tereta su definisani u RFC 1890 (npr. PCM, MPEG1/MPEG2 audio i video, JPEG video, itd.). U jednom trenutku prenosa pošiljaoc RTP paketa može slati samo jednu vrstu tereta iako se tokom prenosa ta vrsta može promeniti (npr. zbog zagušenja mreže). RTP sadrži i informaciju o izvoru signala, omogućava prijemnoj strani da zna odakle dolaze paketi. RTP radi preko UDP-a kako bi iskoristio njegovo multipleksiranje i funkciju zaštitne sume (engl. checksum). TCP i UDP su dva najčešće korišćena prenosna protokola na Internetu. TCP je konekcijski orijentisan protokol koji osigurava direktnu

vezu i pouzdan tok podataka između dve tačke, dok je UDP beskonekcijski orijentisan i nepouzdan datagramski protokol za prenos. UDP je izabran kao odredišni protokol za RTP iz dva razloga. Prvo, RTP je dizajniran primarno za multicast slanje pa mu samim tim direktna TCP veza ne odgovara. Drugo, za real-time aplikacije pouzdanost isporuke nije jednako važna kao pravovremenost dolaska podataka. Pouzdana veza kao što je TCP čak nije poželjna. Npr. prilikom zagušenja mreže neki paketi će biti izgubljeni i aplikacija će moći reprodukovati sadržaj, ali s puno nižim kvalitetom. Ako protokol insistira na pouzdanom prenosu i traži da se izgubljeni paketi ponovo pošalju, to će povećati kašnjenje, zagušiti mrežu i na kraju aplikacija verovatno više neće imati dovoljno podataka za obradu [64].

Na osnovu pregleda tim za kontrolu kvaliteta je proverio još jednom da li predloženi protokoli u skladu sa standardima. Odradili su proveru da li predloženi softverski paketi omaju odgovarajuću dokumentaciju, tehničke papire, dokaze o kompatibilnosti, kao i da li je dostupan sors kod za programe u slobodnom vlasništvu. Nakon utvrđivanja svega ovoga poslato je obaveštenje svim timovima de se može preći u iduću fazu implementiranja.

### **7.1.2 Faza implementacije**

Da bi se istraživanje odradilo na adekvatan način omogućen je pristup laboratoriji za računarske mreže, koja nije imala instaliranu laboratorijsku podršku za internet telefoniju. U svrhu instalacije simulirane virtuelne laboratorije i stvarne hardverske implementacije, laboratorija za računarske mreže obezbedila je 2 serverske mašine, ruter i 25 klijentskih mašina. Detaljna specifikacija mašina data je u Tabeli 1.

Tabela 1: Specifikacija hardvera za implementaciju simuliranog Sistema

Komponenta	Specifikacija
Server	2 x Intel E5-2609 2.4 GHz (4 jezgra)
	16GB ECC DDR3 RAM
	4 x 1TB SATA 7,2K RPM
	Hot swap: napajanje, HDD
	Operativni sistem: CentOS
Klijent	Intel i5-3450 3.5 GHz (4 jezgra)
	4GB DDR3 RAM
	1TB SATA HDD
	10/100/1000 Mbps LAN
	Operativni sistem: Windows 7
Svič (Switch)	Layer 3
	24 x 10/100/1000 Ethernet portova
	2 x SFP Gigabit Ethernet portova
Ruter (Router)	DLP: Ethernet, Fast Ethernet
	RMP: SNMP, HTTP, SSH-2
	Flash memorija: 64MB
	DRAM: 256MB
	Podrška za enkripciju: DES, 3DES, AES 128, AES 192, AES 256

Sva instalirana rešenja postala su sastavni deo istraživanja. Tim zadužen za razvoj virtuelnog VoIP sistema imao je pristup jednoj serverskoj mašini, koja je pružala dovoljno hardverske snage za ispunjenje zahteva simulacije. Ostatak opreme bio je namenjen timu koji je trebalo da implementira stvarni VoIP sistem, zasnovan na rezultatima simulacije.

Na osnovu preporuke prethodnog tima Team\_A, koji je pripremio pregled funkcionalnosti, kao i softverskih i hardverskih zahteva za instalaciju virtuelnog i stvarnog VoIP sistema, Team\_B i Team\_C su angažovani u fazi implementacije. Ova faza uključivala je dva važna koraka za GLP/GMP kvalifikaciju instalacije (IQ) i operativnu kvalifikaciju (OQ). Kvalifikacija instalacije (IQ) je osigurala da je sistem pravilno instaliran u planiranom okruženju. Ovo okruženje je bilo adekvatno za pravilno korišćenje sve opreme uključene u ovu fazu.

Team\_B, koji je bio zadužen za virtuelni VoIP sistem, instalirao je mrežni simulator na serverskoj mašini, softver za virtuelne mašine sa osam virtuelnih mašina. Jedna virtuelna mašina je bila namenjena kao serverska, dok su ostale služile kao klijentske mašine. Virtuelne mašine su povezane preko virtuelne mreže, a na serveru je instaliran PBX server. Na klijentskim mašinama instaliran je softverski telefon 3CXPhone.

Team\_C, koji je bio zadužen za stvarni VoIP sistem, instalirao je računarsku mrežu od osam mašina. Na serverskoj mašini je instaliran PBX softver, dok su na klijentskim mašinama takođe instalirani softverski telefoni 3CXPhone.

Oba tima su koristila istu topologiju i iste postavke za rad VoIP sistema. Pokretanjem ovih sistema i pripremom detaljne dokumentacije o koracima instalacije, ispunjeni su IQ zahtevi GLP/GMP standarda. Na osnovu ovih sistema, Team\_D je završio konfiguraciju PBX-a, podešavanje korisničkih uređaja, povezivanje sa spoljnim mrežama, konferencijske funkcije, nadgledanje sistema i bezbednosne mere.

### **Instaliranje VoIP sistema**

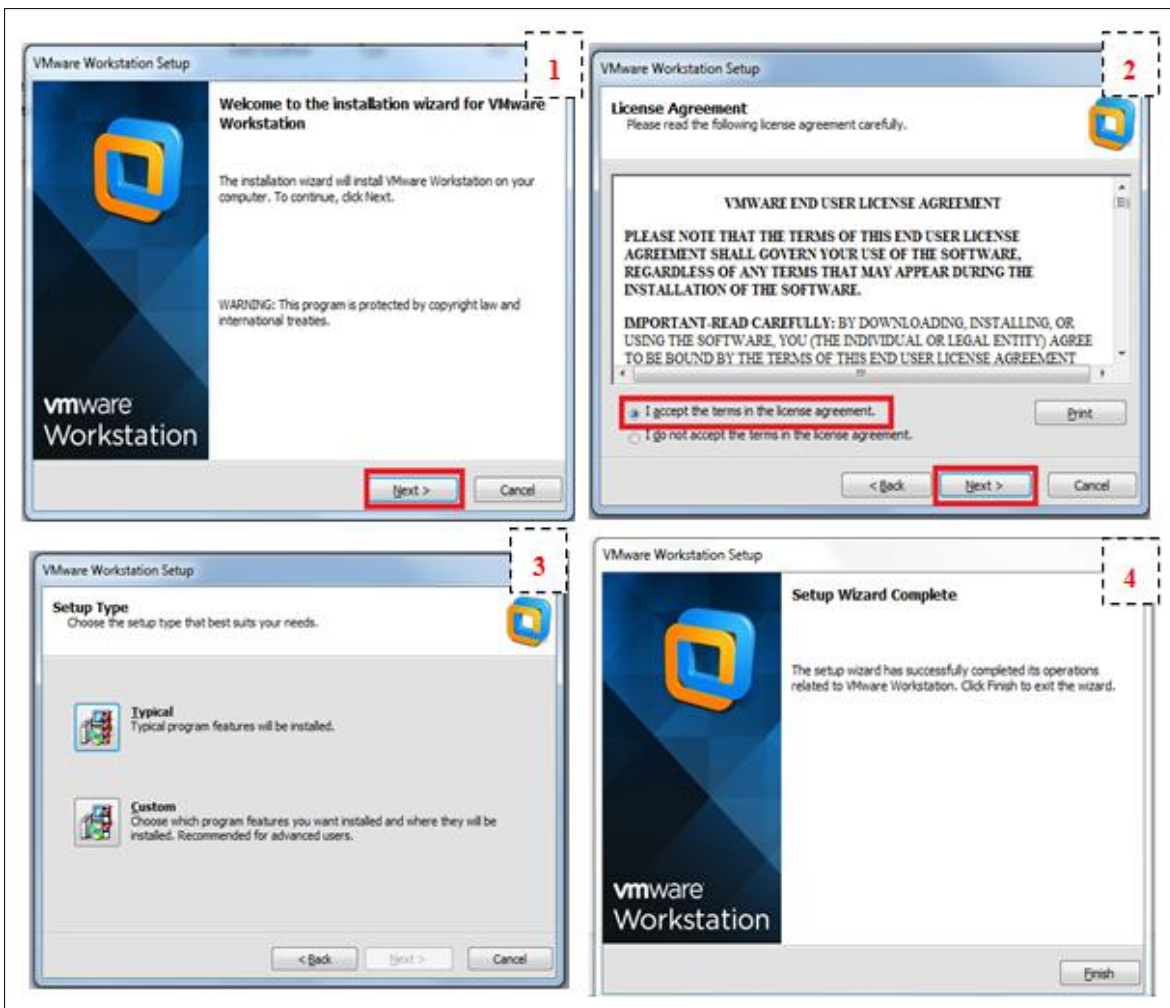
Kao što smo naveli tim koji je ima za zadataka da razvije virtuelnu simulacionu mrežu kao prvi korak morao je da instalira programe neophodne za besprekorno funkcionisanje mreže. Da bi se obezbedila istovetnost postavki instalacija i podešavanje Cisco rutera su radjena istovremeno.

### **VMware Workstation**

VMware Workstation je softver koji pruža mogućnost korisniku/korisnicima da kreiraju neograničen broj virtuelnih mašina na računarima sa x86 procesorima. Svaka virtuelna mašina može imati svoj „gostujući“ operativni sistem kao što su windows, linux i druge slične varijante. Jednostavno govoreći, VMware Workstation dopušta jednoj fizičkoj mašini da pokrene neograničeno mnogo operativnih sistema istovremeno. Ostali VMware proizvodi pomažu rukovođenjem (VMware Player) i prenošenjem (VMware Server) sa određenim prednostima ali i manama u odnosu na VMware Workstation [65].

Osim premošćavanja mrežnih adaptera, CD-ROM čitača, HDD diskova i USB uređaja, VMware Workstation dopušta i simulacije sa hardverom. Na primer, moguće je kreirati ISO fajl za CD-ROM i .vmdk fajlove za HDD; mrežni adapter drajver se može konfigurisati da koristi mrežnu adresu sa prolazom kroz host računar dosta efikasnije nego kada bi koristio premošćenu varijantu (koja bi zahtevalo IP adresu za svaki gostujući operativni sistem na host mreži) [66].

Osnovni koraci instalacije VMware -a na operativnom sistemu Windows 8.1 su prikazani na sledećoj slici (slika 32) :



Slika 32 Proces instalacije VMware Workstation na Windows 8.1 operativnom sistemu

Na slici su brojevima označeni koraci prilikom instalacije datog programa. Instaliranje ovog programa je slično instaliranju drugih programa.

Pošto je proces instalacije *VMware* -a uspešno okončan i pokrenut je program, ono što sada sledi jeste kreiranje i podešavanje novih virtuelnih mašina i instalacija potrebnih operativnih sistema na svakoj virtuelnoj mašini.

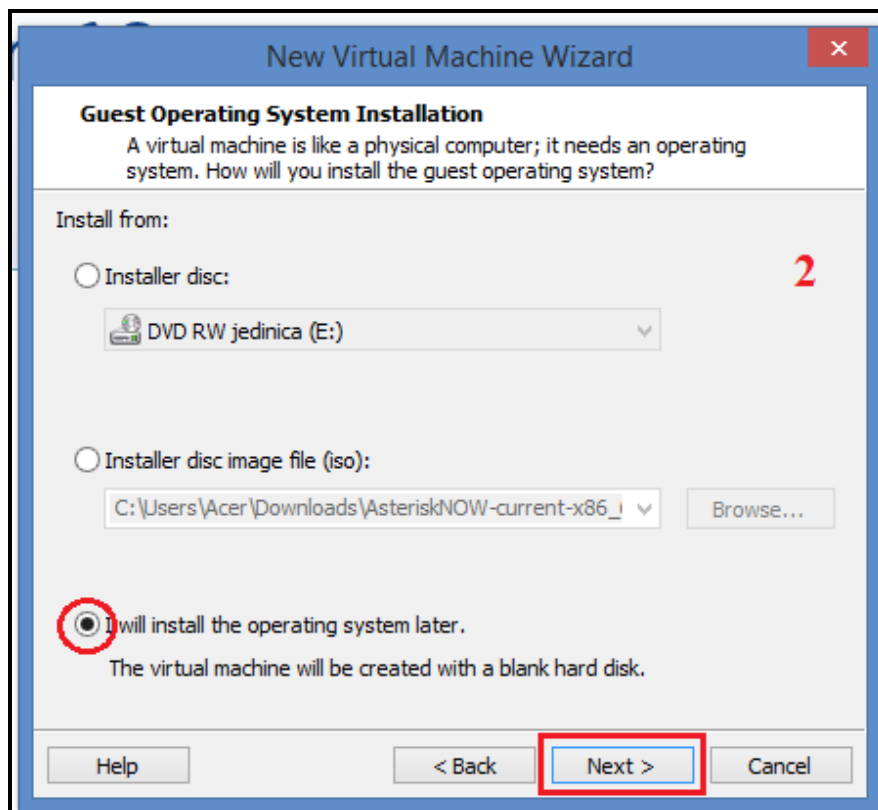
Na slikama koje slede je slikovito (korak po korak) prikazan proces kreiranja i podešavanja nove virtuelne mašine i instalacija Asterisk servera. U prvom delu je prikazan proces podešavanja nove virtuelne mašine, dok je u drugom delu prikazan postupak instalacije Asterisk servera sa sopstvenim operativnim sistemom [67].

U prvom koraku je potrebno odabrati opciju za kreiranje nove virtuelne mašine – New Virtual Machine, čime se pokreće proces kreiranja (slika 33).



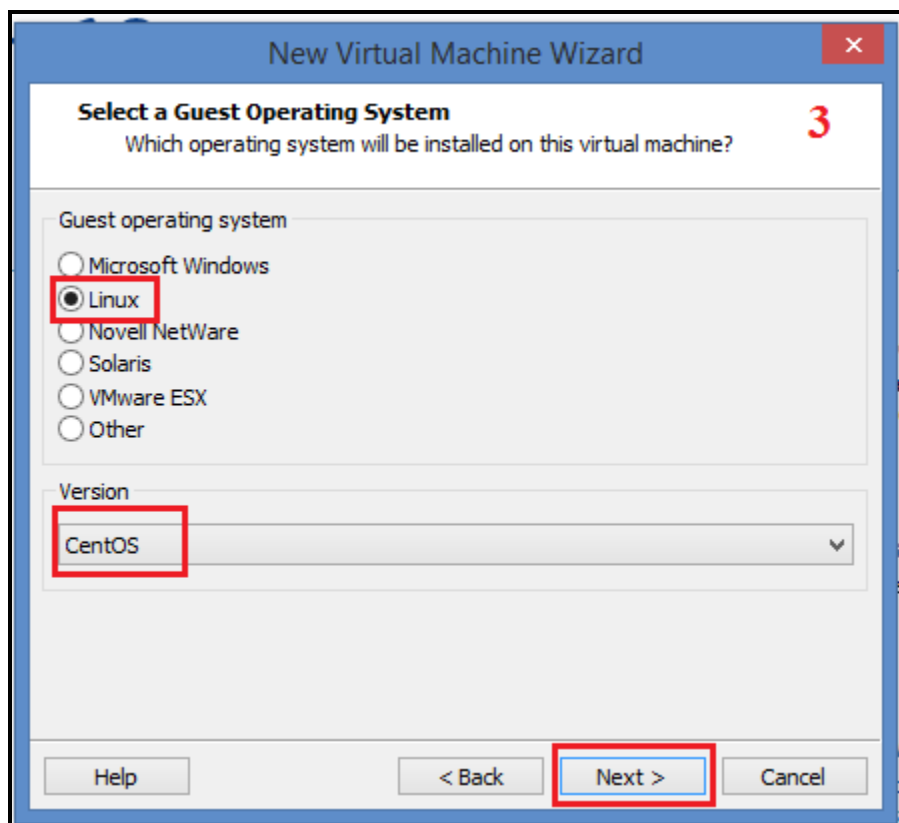
Slika 33 Početni korak pri instalaciji nove virtuelne mašine (korak I)

U narednom koraku biramo odakle i na koji način želimo da instaliramo sistem. U našem slučaju biramo treću opciju, tj. da želimo kasnije instalirati sistem (slika 34).



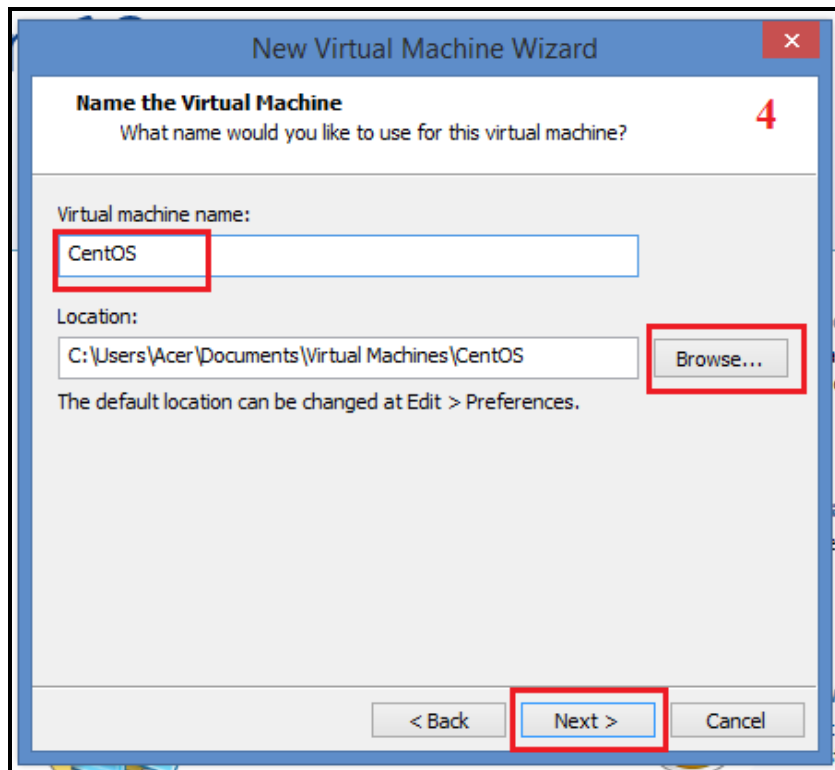
Slika 34 Odabir lokacije instalacionog fajla operativnog sistema i načina instalacije (korak II)

Naredni korak se sastoji u odabiru operativnog sistema koji želimo da instaliramo. U našem primeru je u pitanju *Linux* operativni sistem, a verzija je CentOS, u sklopu koga i dolazi Asterisk server (slika35).



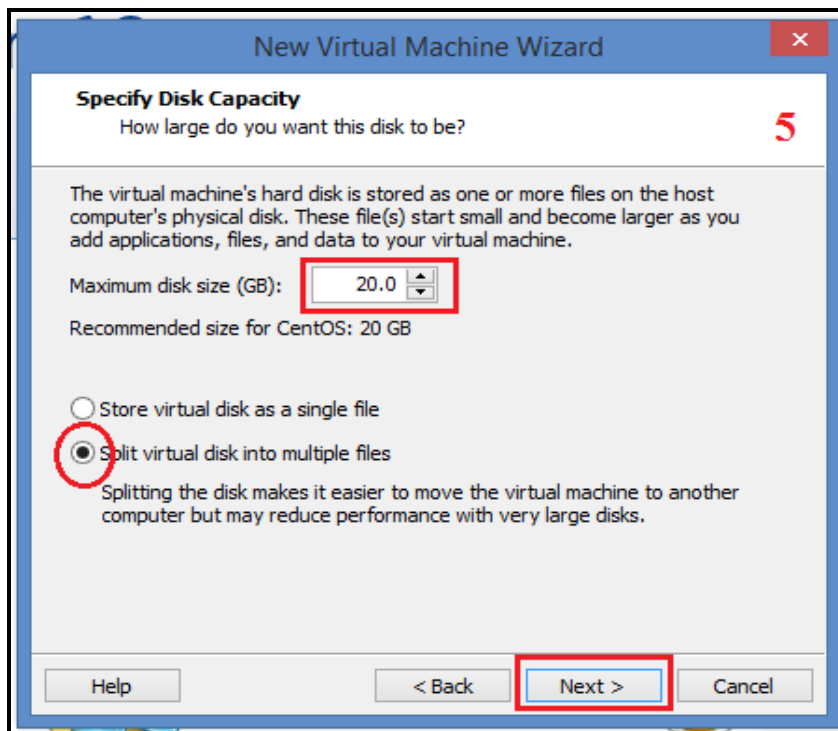
Slika 35 Selektovanje odgovarajućeg operativnog sistema (korak III)

Ono što sistem u sledećem koraku traži od nas jeste da unesemo ime virtuelne mašine i da odaberemo lokaciju gde će biti instalirana (slika 36).



Slika 36 Dodeljivanje imena virtuelnoj mašini i odabir lokacije (korak IV)

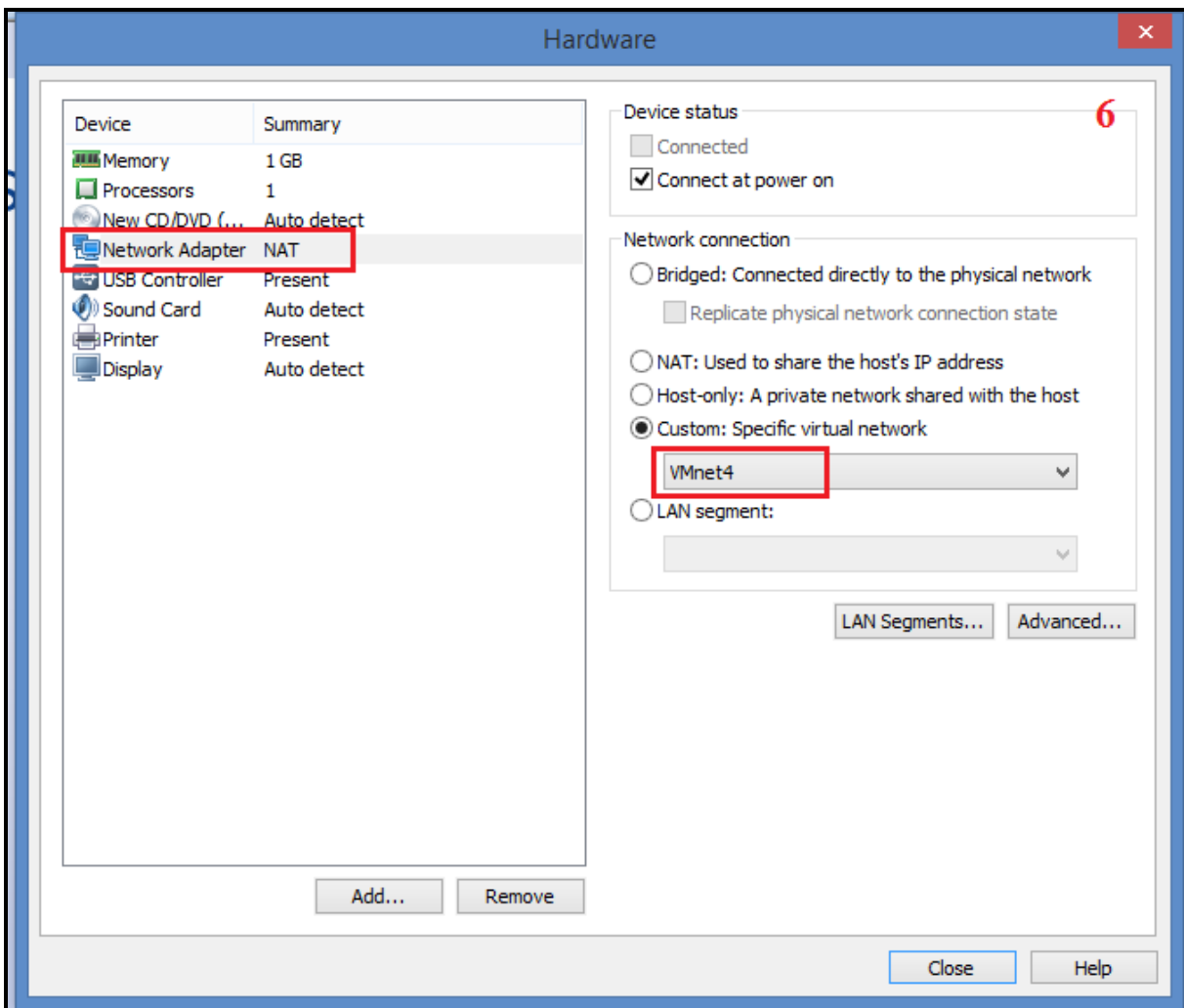
Sledeće što je potrebno odraditi je dodeliti kapacitet diska virtuelnoj memoriji, u našem primeru je 20GB memorije (slika 37).



Slika 37 Dodela memorijskog prostora virtuelnoj mašini na disku (korak V)

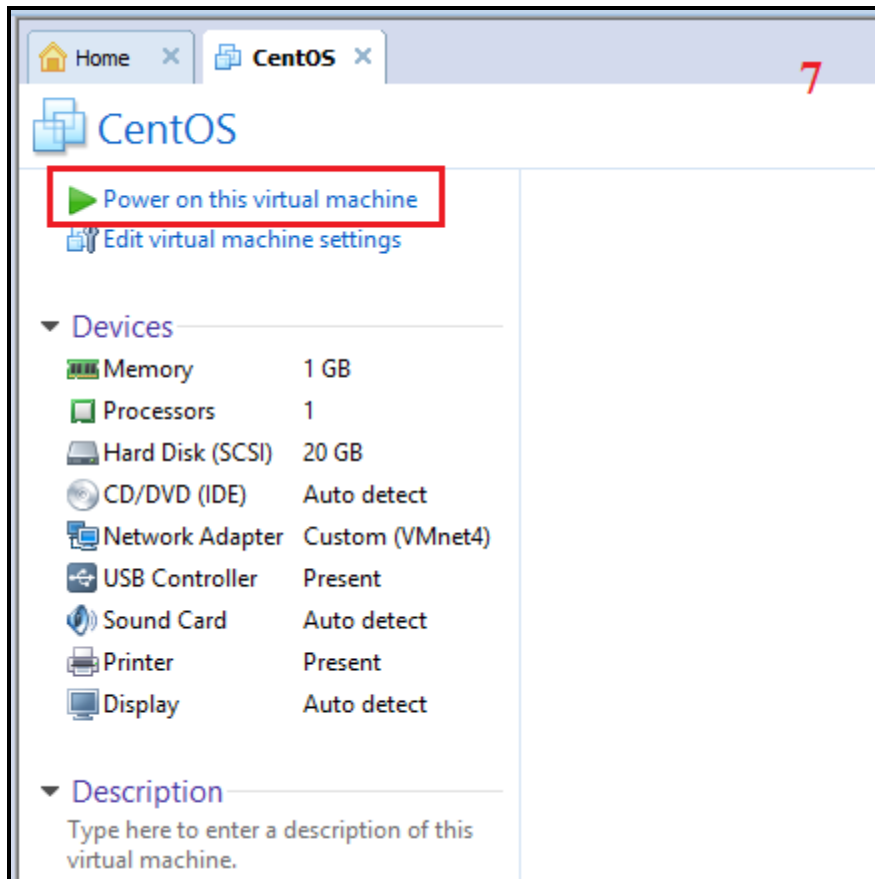
Na slici je prikazan prozor gde nam sistem nudi opcije za podešavanje nekoliko parametara. Parametri se tiču hardvera, gde je, između ostalog, moguće podesiti kapacitet RAM memorije, procesorske jedinice, mrežni adapter, USB kontroler i drugo.

Sve parametre je potrebno podesiti kao što je to prikazano na slici 38, a na istoj slici je pokazan primer podešavanja mrežnog adaptera (engl. *Network Adapter – NAT*). Potrebno je čekirati pretposlednju opciju (*Custom: Specific virtual network*) i u padajućoj listi izabrati željenu virtuelnu mrežu.



Slici 38 Odabir virtuelne mreže (korak VI)

Kada se svi potrebni parametri uspešno podese, virtuelna mašina je uspešno kreirana i time je proces kreiranja završen. U narednom koraku (slika 39) se vrši pokretanje kreirane virtuelne mašine i sledi proces instalacije operativnog sistema [68].



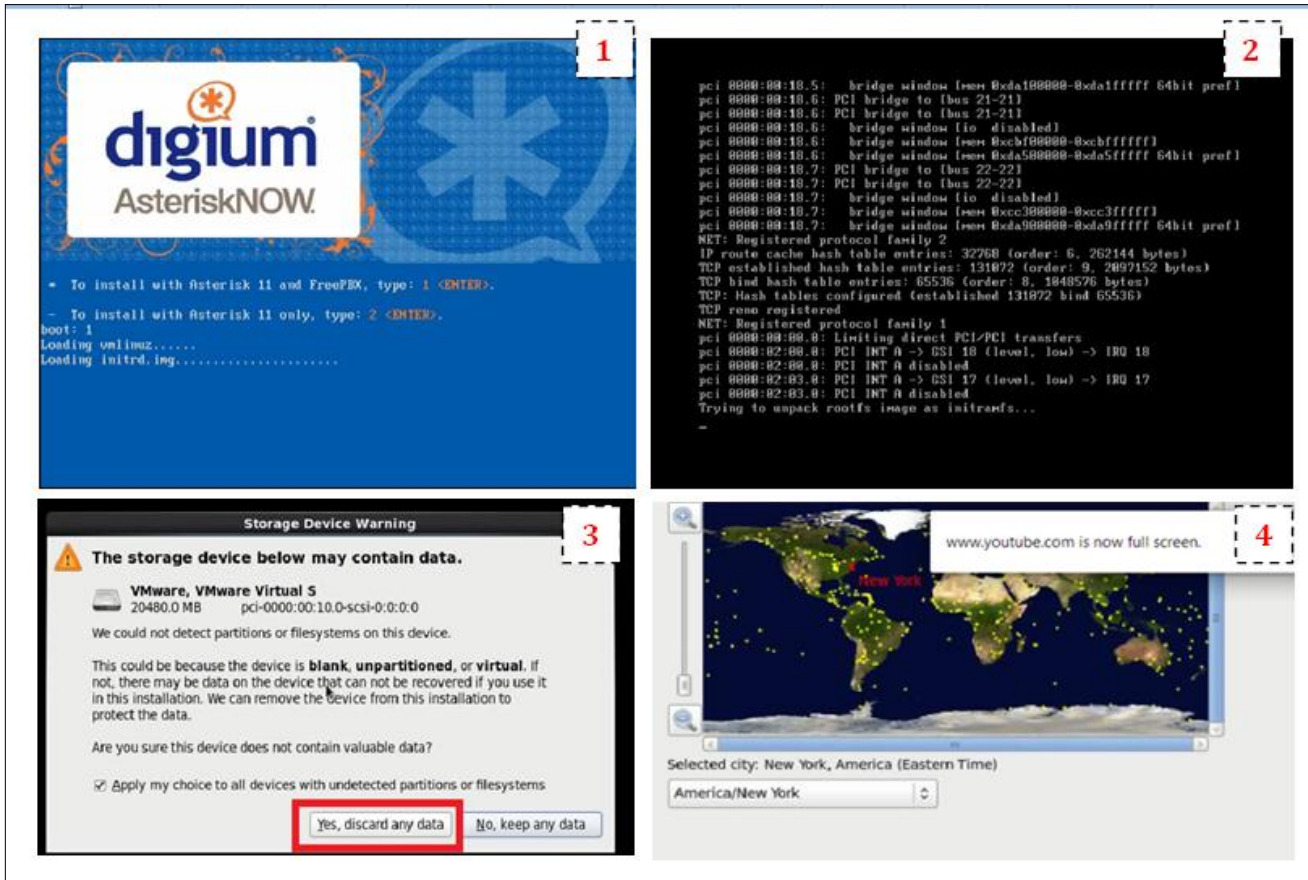
Slika 39 Pokretanje kreirane virtuelne mašine (korak VII)

### **Instalacija Asterisk PBX**

Pokretanjem kreirane virtuelne mašine započinje proces instalacije CentOS operativnog sistema u sklopu koga dolazi Asterisk server.

Na slici 40 su prikazani neki koraci prilikom instalacije CentOS operativnog sistema. U prvom koraku je potrebno izabrati između dve opcije. U našem primeru smo odabrali broj 1 gde se Asterisk server instalira sa FreePBX koji nam omogućava lakši rad sa serverom [69].

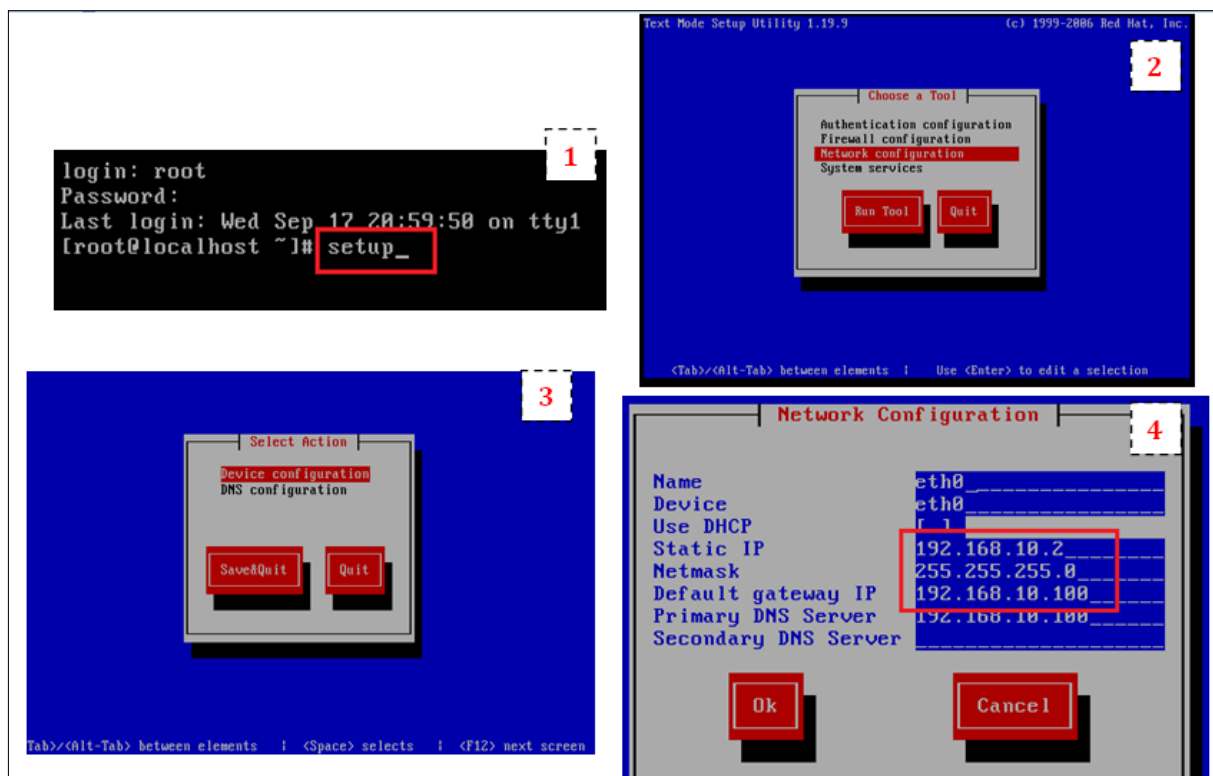
U narednim koracima je, izneđu ostalog, potrebno podesiti kapacitet smeštajne memorije na disku, podesiti vreme i sl. Instalacija je slična instalaciji ostalih verzija Linux sistema [70].



Slika 40 Postupak instalacije CentOS operativnog sistema

Nakon instalacije Asterisk servera potrebno je konfigurisati statičku IP adresu kako bi server moga da komunicira sa ostalim učesnicima u mreži.

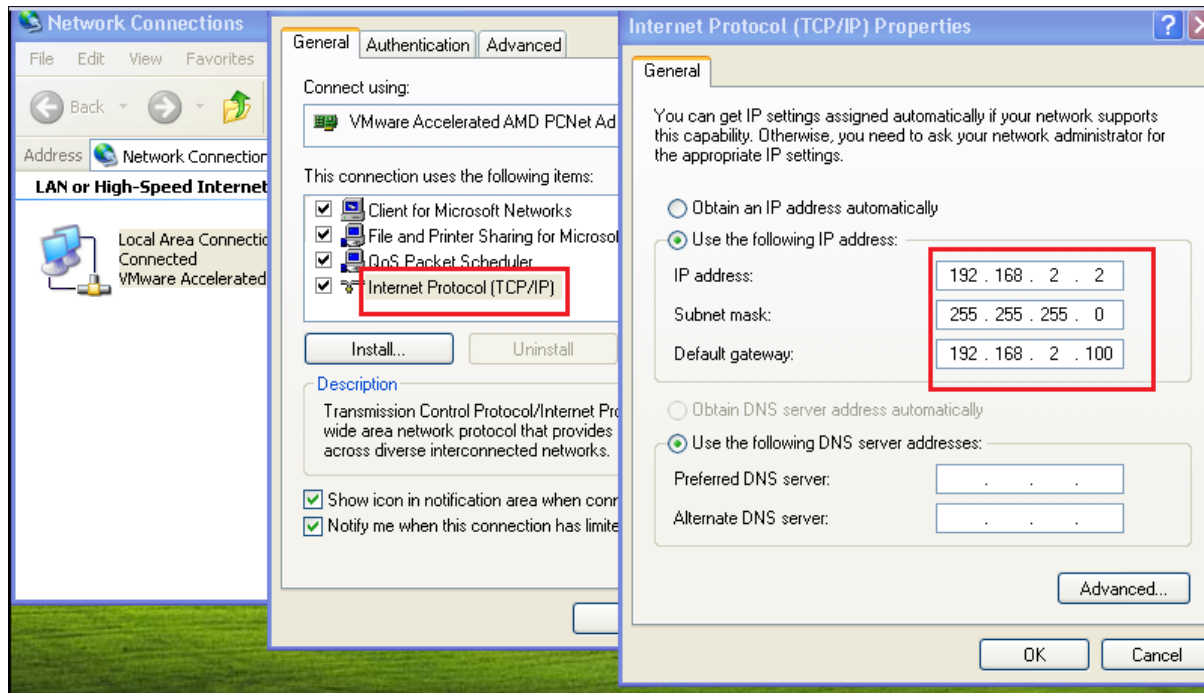
Prvi korak je u komandnom modu ukucati komandu setup koja nam omogućava da pristupimo opcijama za podešavanje IP adrese (slika 41 koraci 2 i 3) [71]. Kada pristupimo opcijama za podešavanje mreže, potrebno je popuniti sledeća polja: Static IP, Netmask, Default gateway IP kao na slici 41 korak 4.



Slika 41 Podešavanje statičke IP adrese na Asterisk serveru

Osim prethodne virtualne mašine potrebno je instalirati još dodatne dve. Na obe virtualne mašine je potrebno instalirati Windows XP operativni sistem na kojima će kasnije biti instalirani softverski telefoni i povezani sa Asterisk serverom. Postupak kreiranja ostale dve virtualne mašine je sličan prethodno prikazanom, jedina razlika je u odabiru operativnog sistema kojeg želimo da instaliramo na kreiranoj virtualnoj mašini [72].

Kako i kod Asterisk servera i kod Windows XP je potrebno podesiti statičku IP adresu kako bi bilo moguće povezivanje tih virtualnih mašina na mrežu. Primer podešavanja statičke IP adrese na jednoj virtualnoj mašini na kojoj je instaliran Windows XP je prikazan na slici 42. Identičan postupak je potrebno ponoviti za drugu virtualnu mašinu na kojoj je, takođe, instaliran operativni sistem Windows XP [73].

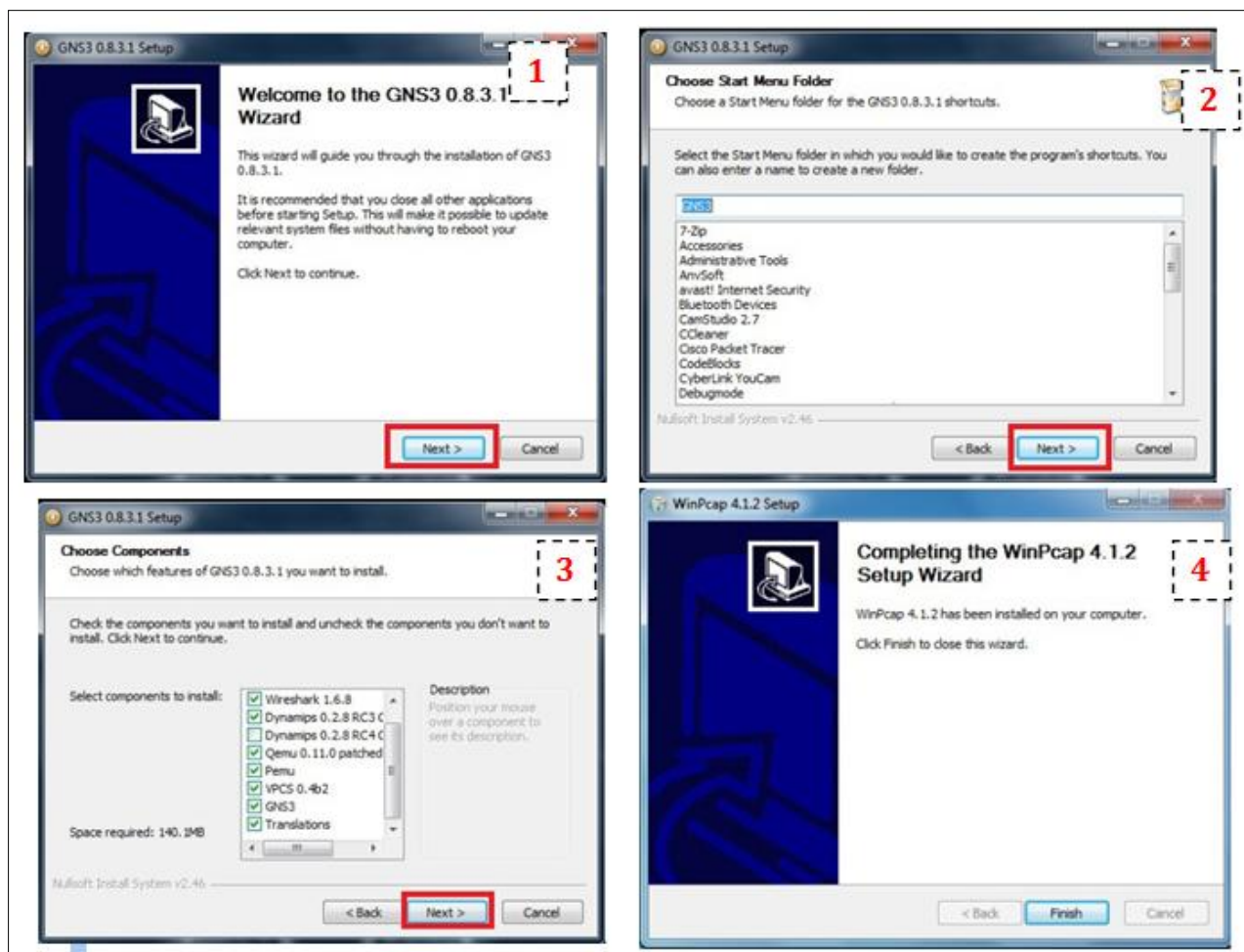


Slika 42 Podešavanje statičke IP kod Windows XP operativnog sistema

### Instaliranje GNS3 (Graphical Network Simulator 3)

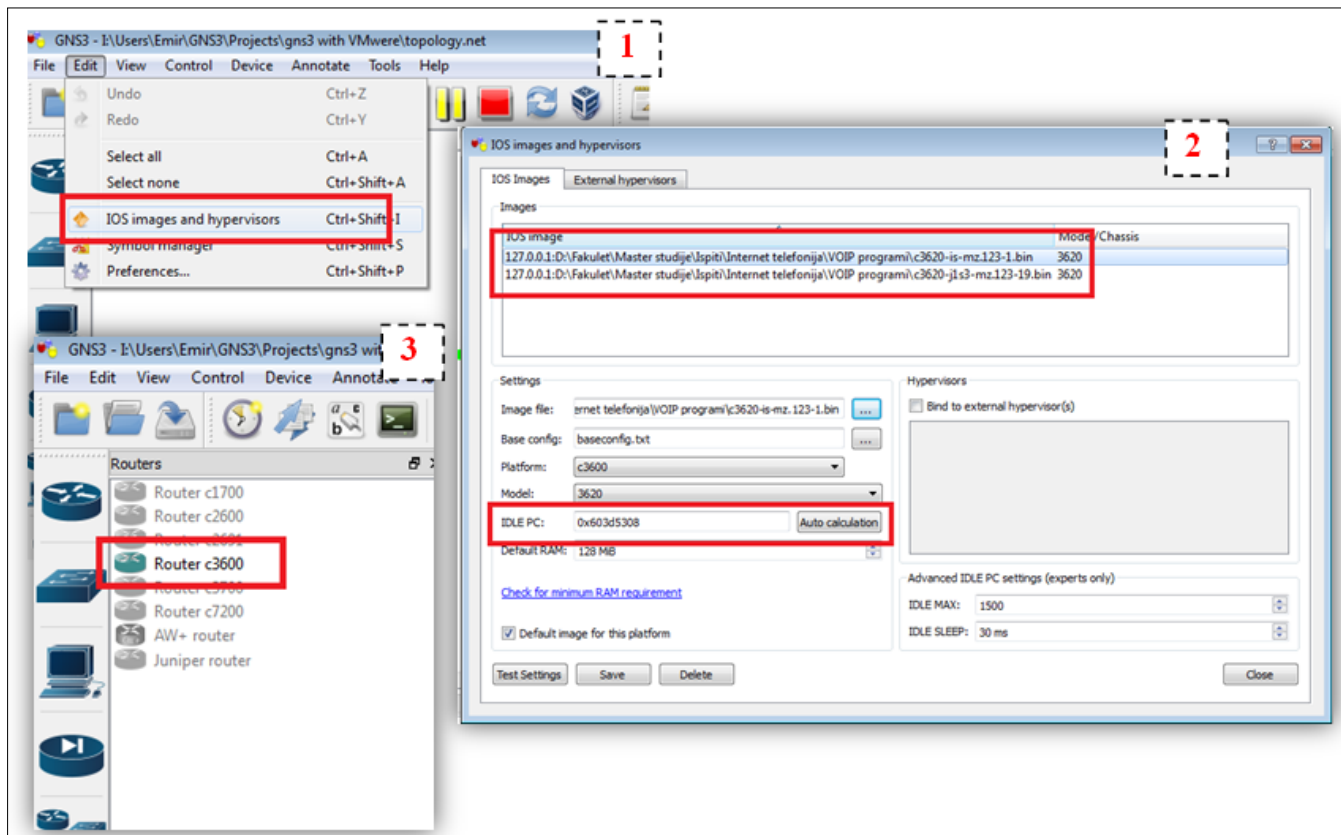
GNS3 je softverski alat koji služi da povezivanje više pod mreža kao i virtuelnih mreža. Ovaj softver se koristi kako bi povezali virtuelne mašine sa VoIP serverom. Kao i kod CISCO softvera za projektovanje mreže i ovaj softver koristi sve alate za kreiranje mreže. U ovom delu biće opisano i predstavljeno način na koji će biti povezane virtuelne mašine kao i način na koji komuniciraju. Prema zahtevima projekta neophodno je povezati virtuelne mašine u jednu mrežu gde će one komunicirati kao jedna mreža. U prvom delu biće predstavljena instalacija GNS3 softvera. U drugom delu biće opisan načina na koji se unosi poseban softver za virtuelni ruter a to je IOS Image. Treći deo ove oblasti prikazaće način na koji se vežu ove virtuelne mašine i poslednji deo konfiguracija rutera [74].

Na slici 43 prikazani su neki koraci prilikom instaliranja ovog softverskog paketa. Proces instalacije se ne razlikuje puno od instalacije ostalih programa, tako da su na slici prikazani samo neki koraci kako bi vam približili i olakšali instalaciju ovog programa. Ono što treba napomenuti, i što se vidi na slici 43 (korak 3), je da u sklopu ovog softvera, pored ostalih programa, možemo instalirati i wireshark program što je u ovom slučaju i učinjeno [75].



Slika 43 Instalacija GNS3 softvera

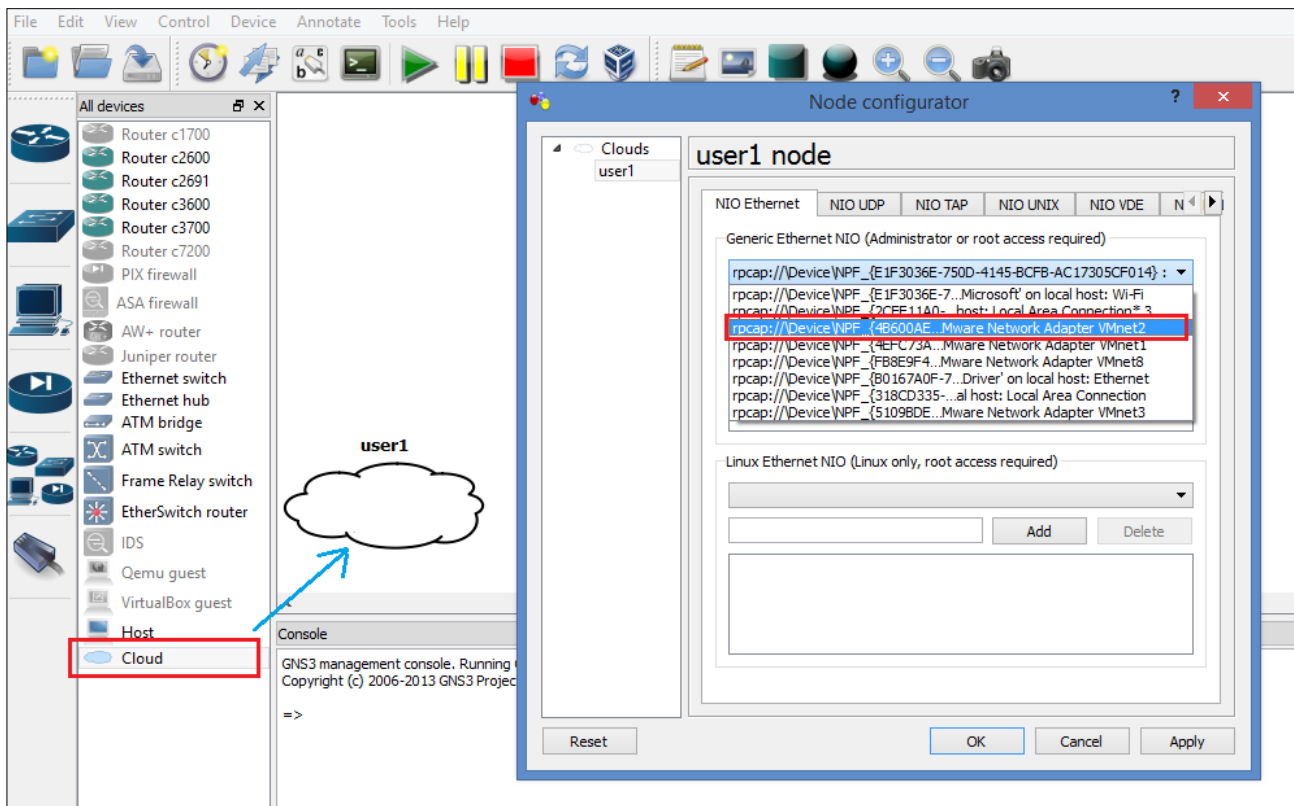
Kada je reč o GNS3 softveru neophodno je omogućiti pristup već ponuđenom alatu odnosno virtuelnom ruteru. Da bi koristili ovaj ruter neophodno je pronaći sistem za taj ruter a taj sistem predstavlja IOS Images fajl. Konkretno u ovom slučaju je IOS Images fajl za ruter tipa c3600. U sledećim koracima biće predstavljena upotreba tog fajla tj. unos fajla u odgovarajuće polje (slika 44).



Slika 44 Unos IOS Images fajla

Prema zahtevima projekta neophodno je povezati virtualne mašine kako bi se omogućila međusobna komunikacija i VoIP telefonija. Da bi to odradili potrebno je koristiti na pravi način ponuđene alate GNS3 softvera. U prvom koraku neophodno je za svaku virtualnu mašinu koristiti alatku Cloud u GNS3. Nakon toga u podešavanjima cloud-a potrebno je postaviti koji interfejs odnosno mrežni adapter ta virtualna mašina koristi. Tako se odradi za svaku virtualnu mašinu. Nakon toga se koristi ponuđeni ruter i konfigurira se da pored svoja dva ima još jedan interfejsa. Na radnoj površini se nalaze nepovezani elementi koje vezemo sa FastEthernet.

Na slici 45 je prikazan postupak dodavanja cloud-a i njegovo povezivanje sa mrežnim adapterom određene virtualne mašine, tako da ta virtualna mašina preko ovog cloud-a komunicira sa ostalim delovima mreže. Nad cloud-om je moguće vršiti određena podešavanja iz liste ponuđenih koja nam se otvara desnim klikom miša na sami cloud. Između ostalog, moguće je menjati ime, simbol i sl.



Slika 45 Povezivanje VMwere mrežnog adaptera sa sloud-om

Ovaj postupak koji je prikazan na prethodnoj slici, potrebno je ponavljati onoliko puta koliko želimo dodati cloud-a. Naravno, jednom cloud-u se dodeljuje jedna virtuelna mašina.

## Konfigurisanje VoIP sistema

### Konfigurisanje Cisco c3700 rutera

Internetworking Operating System (IOS) je porodica vlasničkih mrežnih operativnih sistema koji se koriste na više modela rutera i mrežnih svičeva proizvedenih od strane Cisco Systems. Ovaj sistem predstavlja paket funkcija za rutiranje, preklapanje (switching), umrežavanje i telekomunikacije, integrisanih u operativni sistem sa multitaskingom. Iako IOS sadrži osnovu za kooperativni multitasking, većina njegovih funkcionalnosti je portovana na druge jezgre, poput Linux-a i QNX-a, za upotrebu u Cisco proizvodima.

Nisu svi Cisco mrežni uređaji pokretani IOS-om. Na primer, neki Cisco Catalyst svičevi koriste IOS XE, dok Cisco ASR ruteri rade na IOS XE ili IOS XR, oba zasnovana na Linux-u. Za data-centar okruženja, Cisco Nexus Ethernet svičevi i Cisco MDS Fibre Channel svičevi koriste Cisco NX-OS, koji je takođe Linux-bazirani operativni sistem.

Operativni sistem IOS razvijen je na osnovu koda koji je napisao William Yeager na Univerzitetu Stanford, tokom 1980-ih, za rutere sa 256 kB memorije i slabim procesorskim mogućnostima. Kroz modularna proširenja, IOS je prilagođen naprednijem hardveru i novim mrežnim protokolima. Kada je IOS razvijan, Cisco Systems je bio fokusiran prvenstveno na rutere, ali je kasnijim akvizicijama kompanija u oblasti svičeva (kao što je Kalpana, tvorac prvog Ethernet sviča) proširila ponudu. Pre IOS-a, serija Cisco Catalyst svičeva koristila je CatOS.

IOS CLI (Command-Line Interface) pruža fiksni skup višerečenskih komandi, a dostupne komande zavise od "moda" i nivoa privilegija korisnika. Global Configuration Mode omogućava promene u sistemskoj konfiguraciji, dok Interface Configuration Mode omogućava konfiguraciju pojedinačnih interfejsa. Komande su rangirane od nivoa 0 do 15, pri čemu viši nivo privilegija omogućava pristup naprednim komandama.

Cisco IOS ima monolitnu arhitekturu, zbog ograničenih hardverskih resursa rutera i svičeva iz 1980-ih. Sve operacije imaju direktan pristup hardveru radi optimizacije procesorskog vremena. IOS koristi scheduler "run to completion", što znači da ne prekida aktivne procese – procesi se sami moraju osloboditi da bi omogućili drugim procesima da rade. IOS ne podržava zaštitu memorije između procesa, što znači da jedan proces može prepisati memoriju drugog, što potencijalno može destabilizovati ceo sistem.

U svim verzijama Cisco IOS-a, rutiranje i preklapanje paketa (forwarding) su odvojene funkcije. Routing protokoli (kao što su OSPF, BGP) rade kao procesi unutar IOS-a i ažuriraju Routing Information Base (RIB), koji se zatim obrađuje i kreira Forwarding Information Base (FIB), koji ruter koristi za preusmeravanje paketa. U uređajima sa hardverskim ubrzanjem, kao što su Cisco 12000 serije, IOS izračunava FIB softverski, a zatim ga prebacuje na ASIC ili network processor, koji vrši fizičko preklapanje paketa.

IOS dolazi u obliku pojedinačnih image fajlova, koji su posebno kompajlirani za određene Cisco uređaje. Svaki IOS image sadrži određeni set funkcionalnosti. Nadogradnja sa jedne verzije na drugu zahteva učitavanje novog IOS image fajla i restart sistema.

Od Cisco 1900, 2900 i 3900 serija ISR rutera, Cisco je uveo univerzalne IOS image fajlove, koji uključuju sve funkcionalnosti, a korisnici mogu otključati određene opcije kupovinom dodatnih licenci.

Dostupni feature pack paketi su: Data – MPLS, L2TPv3, Mobile IP, SCTP; Security – VPN, Firewall, IP SLAs, NAC; Unified Communications – VoIP, SIP, CallManager Express.

Setup mode pruža interaktivni meni koji pomaže u kreiranju početnog konfiguracionog fajla za novi mrežni uređaj ili uređaj čiji je startup-config fajl izbrisan iz NVRAM-a. Interaktivni meni vodi korisnika kroz početnu konfiguraciju i koristan je kada korisnik nije upoznat sa Cisco uređajima ili komandnom linijom (CLI) i kada izmene konfiguracije ne zahtevaju visok nivo detalja koji CLI pruža.

Setup mode se može koristiti i za izmenu postojeće konfiguracije.

Ovaj odeljak opisuje kako koristiti System Configuration Dialog za pripremu Cisco mrežnog uređaja za potpunu konfiguraciju i kako izvršiti izmene nakon završetka početne konfiguracije.

Koristili smo terminalski program (npr. HyperTerminal) sa sledećim podešavanjima:

- 9600 baud
- 8 data bits, no parity, 1 stop bit
- No flow control

Zatim smo uneli EXEC mode.

```
Router> enable
Router#
```

Izbrisali postojeća podešavanja u NVRAM.

```
Router# erase startup-config
```

Inicirali AutoInstall

```
Router# reload
```

Zatim smo potvrdili da želimo da nastavimo inicijalnu konfiguraciju.

```
--- System Configuration Dialog--Would you like to enter the initial configuration
dialog? [yes/no]: yes
```

```
--- System Configuration Dialog--Continue with configuration dialog? [yes/no]:
```

Aktivirali Basic management mode.

```
At any point you may enter a question mark '?' for help. Use ctrl-c to abort
configuration dialog at any prompt. Default settings are in square brackets '['. Basic
management setup configures only enough connectivity for management of the system,
extended setup will ask you to configure each interface
on the system Would you like to enter basic management setup? [yes/no]:
```

```
Would you like to enter basic management setup? [yes/no]: yes The screen displays the
following: Configuring global parameters: Enter host name [R1]:
```

Uneli Hostname za ruter

```
Configuring global parameters: Enter host name [R1]: Router The screen displays the
following: The enable secret is a password used to protect access to privileged EXEC and
configuration modes. This password, after entered, becomes encrypted in the
configuration. Enter enable secret:
```

Postavili lozinku

Configure SNMP Network Management? [no]:

IOS zatim daje listu svih raspoloživih interfejsa.

```
Current interface summary Interface Ethernet0/0 IP-Address unassigned OK? Method
Status Prol YES NVRAM administratively down dow Ethernet1/0 Serial2/0 Serial3/0
Loopback0 unassigned unassigned unassigned 1.1.1.1 YES NVRAM administratively
down dow YES NVRAM administratively down dow YES NVRAM administratively
down dow YES NVRAM up Enter interface name used to connect to the management
network from the above interface summary:
```

Zatim doobijamo listu promptpova za podešavanje svih parametara mreže.

```
Configuring interface Ethernet0/0: Configure IP on this interface? [no]: yes IP address for
this interface: 172.17.1.1 Subnet mask for this interface [255.255.0.0] : Class B network
is 172.17.0.0, 16 subnet bits; mask is /16 The following configuration command script
was created: hostname Router enable secret 5 $!$1Gg9$GuxXfUUBBfVqGvIW4psIm1
enable password cts54tnl line vty 0 4 password t!s6gato no snmp-server ! no ip routing !
interface Ethernet0/0 no shutdown ip address 172.17.1.1 255.255.0.0 ! interface
Ethernet1/0 shutdown no ip address ! interface Serial2/0 shutdown no ip address !
interface Serial3/0 shutdown no ip address ! end [0]
```

Go to the IOS command prompt without saving this config. [1] Return back to the setup without saving this config. [2] Save this configuration to nvram and exit.

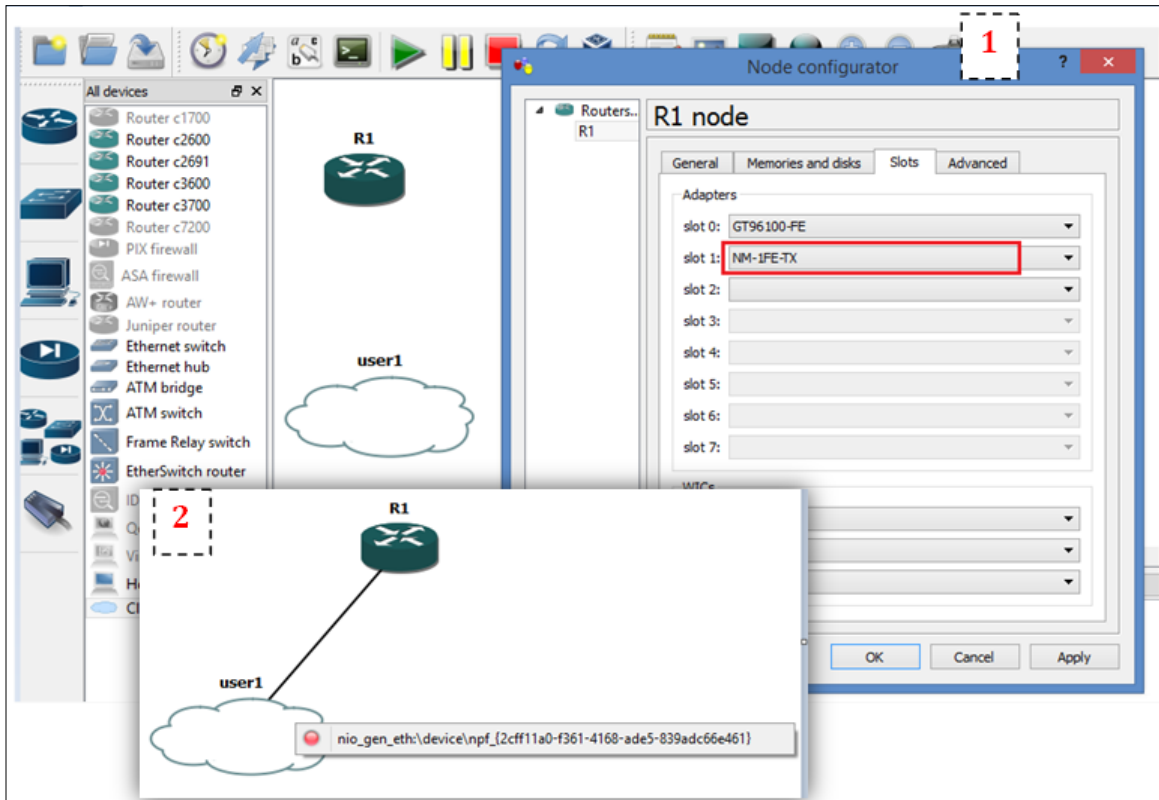
Enter your selection [2]:

Time smo postavili ruter na relanoj mreži, a takođe i podesili virtuelni ruter dobijen snimanjem IOS na virtuelnoj mreži definisanoj na GNS3.

### **Konfigurisanje cisco c3700 rutera za rad u postavljenom cloud sistemu**

После конфигурисања рутера потребно га је повезати са свим претходно конфигурираним cloud-има. Повезивање се врши са FastEthernet линком. Пример повезивања је дат на слици 46 (корак 2).

Posle konfigurisanja rutera potrebno ga je povezati sa svim prethodno konfigurisanim cloud-ima. Povezivanje se vrši sa FastEthernet linkom. Primer povezivanja je dat na slici 46 (korak 2).



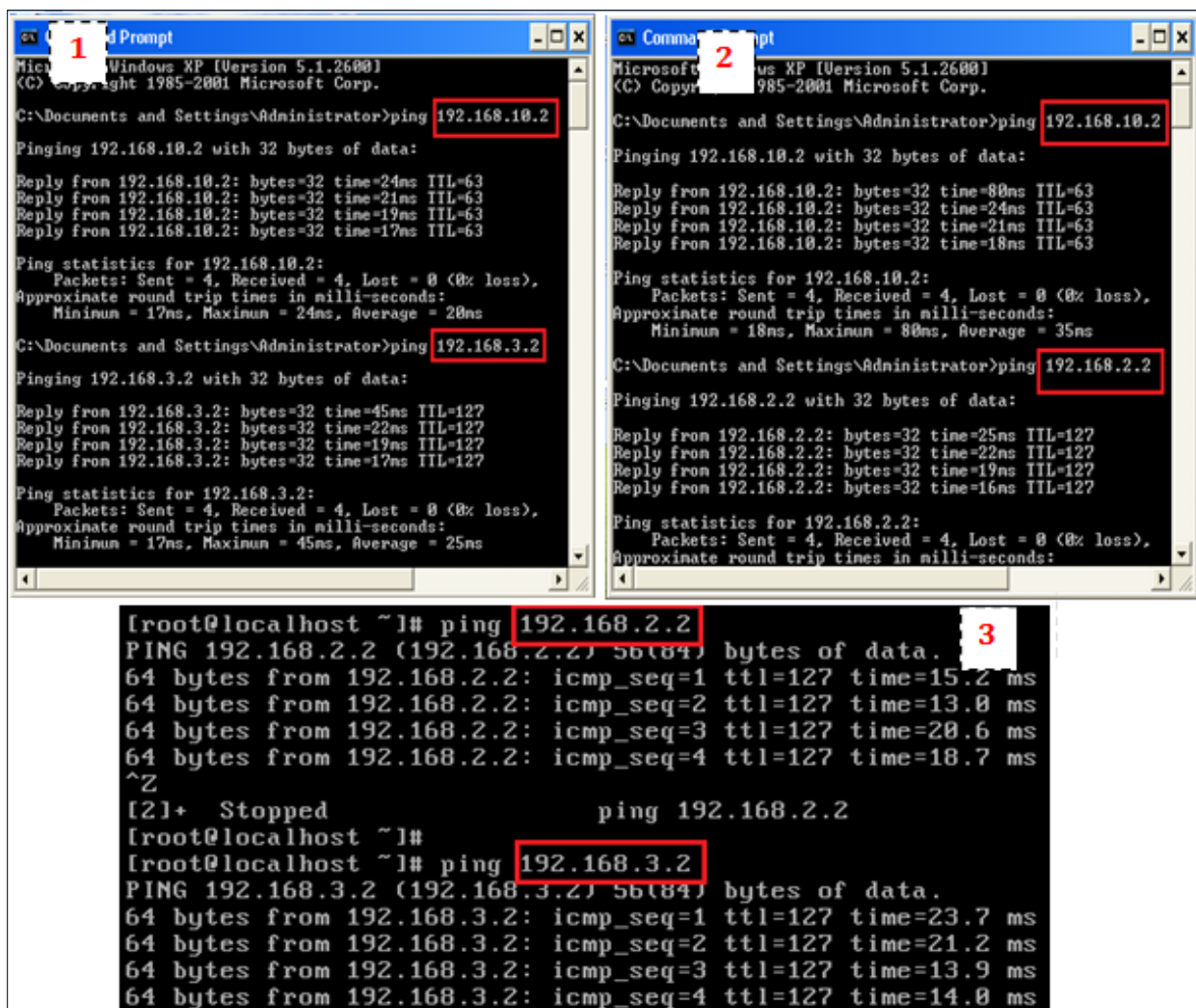
Slika 46 Dodavanje i povezivanje rutera sa cloud-om

#### Провера конекције – пинговање

Kada se proces povezivanja cloud-a sa ruterom završi i izvrše se naophodna podešavanja rutera, potrebno je izvršiti proveru da li mreža uspešno funkcioniše, tj. da li komponente mreže mogu uspostaviti međusobnu komunikaciju. To ćemo izvršiti klasičnom metodom, tzv. pingovanje.

Na slici 47 dat je primer provere mreže pingovanjem. To se vrši tako što u komandnoj liniji svake virtuelne mašine pingujemo preostale dve virtuelne mašine i to ukucamo ping i IP adresu virtelne mašine koju želimo da pingujemo (npr. ping 192.168.10.2):

- 1) Pingujemo server (192.168.10.2) i virtuelnu mašinu user2 (192.168.3.2) sa virtuelne mašine user1;
- 2) Pingujemo server (192.168.10.2) i virtuelnu mašinu user1 (192.168.2.2) sa virtuelne mašine user2;
- 3) Pingujemo virtuelne mašine user1 (192.168.2.2) i user2 (192.168.3.2) sa servera;



Slika 47 Primeri pingovanja između virtuelnih mašina

Pošto su sve kombinacije pingovanja bile uspješne, možemo konstatovati da mreža funkcioniše i da komponente mreže mogu međusobno komunicirati i slati pakete podataka.

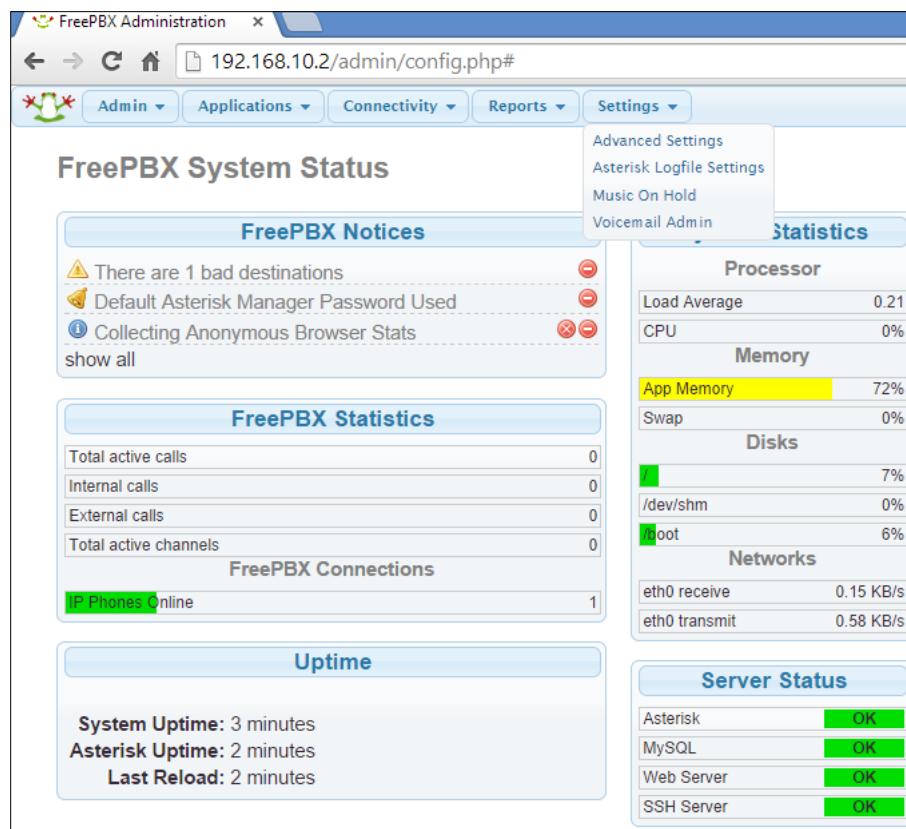
### Konfigurisanje Asterisk servera

Nakon instalacije Asteriska i drugih sistema potrebno je konfigurisati Asterisk server i kreirati nove VoIP naloge, zatim instalirati VoIP telefone na sistemima i nakon toga uspostaviti komunikaciju.

Prilikom instalacije Asterisk servera na virtuelnoj mašini odabrali smo opciju da u sklopu servera instaliramo i kućnu telefonsku centralu (PBX) koja nam omogućava jednostavniji rad i lakše dodavanje i korigovanje naloga.

PBX kućnoj telefonskoj centrali se pristupa tako što se u veb pretraživaču ukuca IP adresa (u našem primeru *192.168.10.2*) koja je dodeljena pri instalaciji asteriska na virtuelnoj mašini (vidi: slika 41).

Na slici 48 je prikazana početna strana PBX sistema.



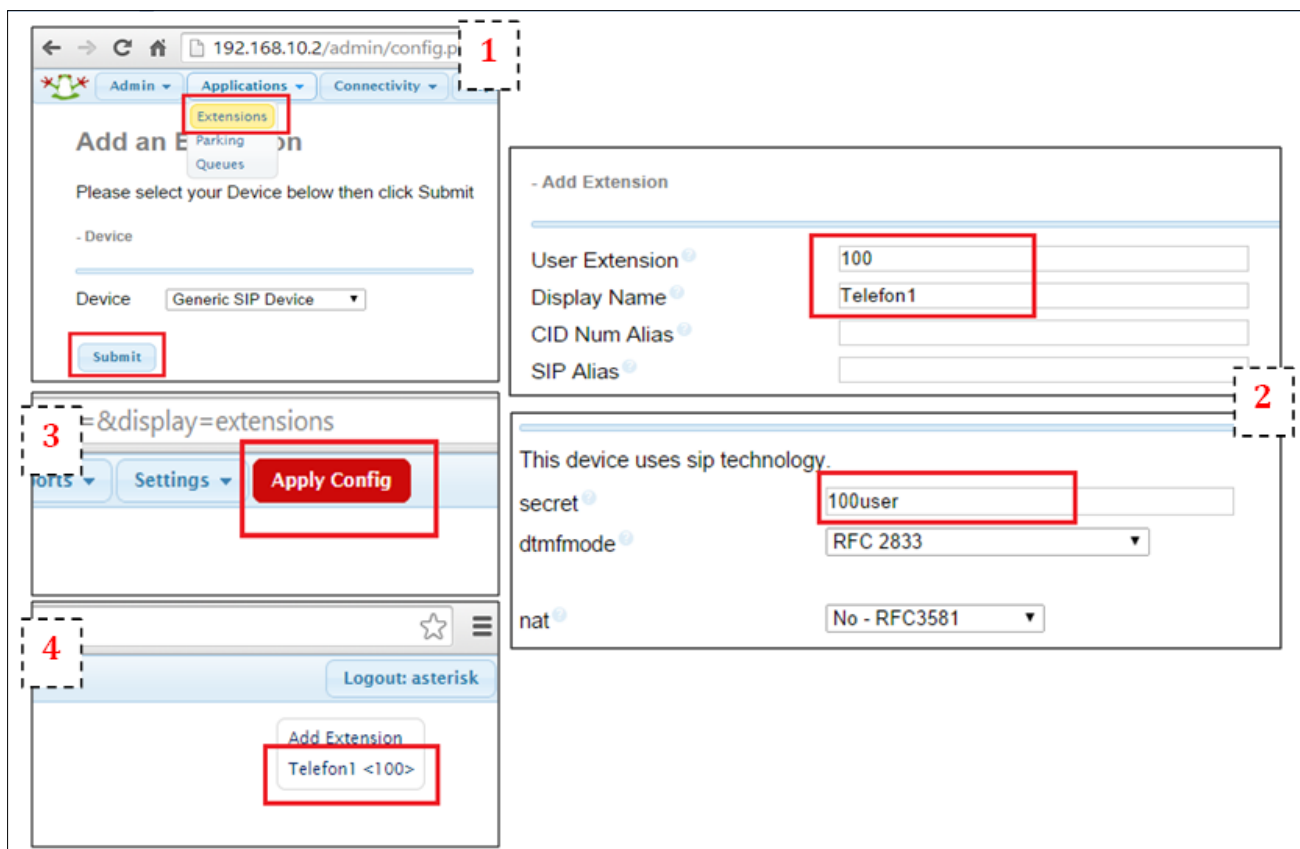
Slika 48 Početna strana PBX centrale

Asterisk server je softverski alat koji u sebi sadrži dosta raznovrsnih funkcija i mogućnosti. Neke od njih će biti prikazane i demonstrirane u ovom radu. Jedna od prvih potrebnih funkcija je kreiranje VoIP naloga. Na narednoj slici su prikazani koraci prilikom kreiranja novog naloga (slika 49).

Proces kreiranja novog SIP VoIP naloga započinje tako što se iz horizontalnog menija biraju opcije Applications -> Extensions gde se otvara novi prozor na kome biramo tip našeg uređaja (Generic SIP Devices, Generic IAX2 Devices, ...), u ovom primeru je to Generic SIP Devices i klikom na dugme Submit (slika 49, korak 1) se prelazi na unos parametara.

Sledeći korak je unošenje potrebnih parametara koji će definisati SIP nalog. Ono što je u ovom primeru predstavljeno (slika 49, korak 2) su sledeći parametri: User Extension – u ovo polje unosimo broj koji će biti dodeljen telefonu nakon kreiranja; Display Name – ime telefona; secret – predstavlja šifru koja se dodeljuje telefonu.

Nakon unetih parametara klikom na Apply Config dodajemo novi nalog (slika 48, korak 3) koji je prikazan u obliku kao na slici 49 (korak 4), i time je završen proces dodavanja naloga. Isti postupak ponoviti onoliko puta koliko je potrebno kreirati naloga na serveru.



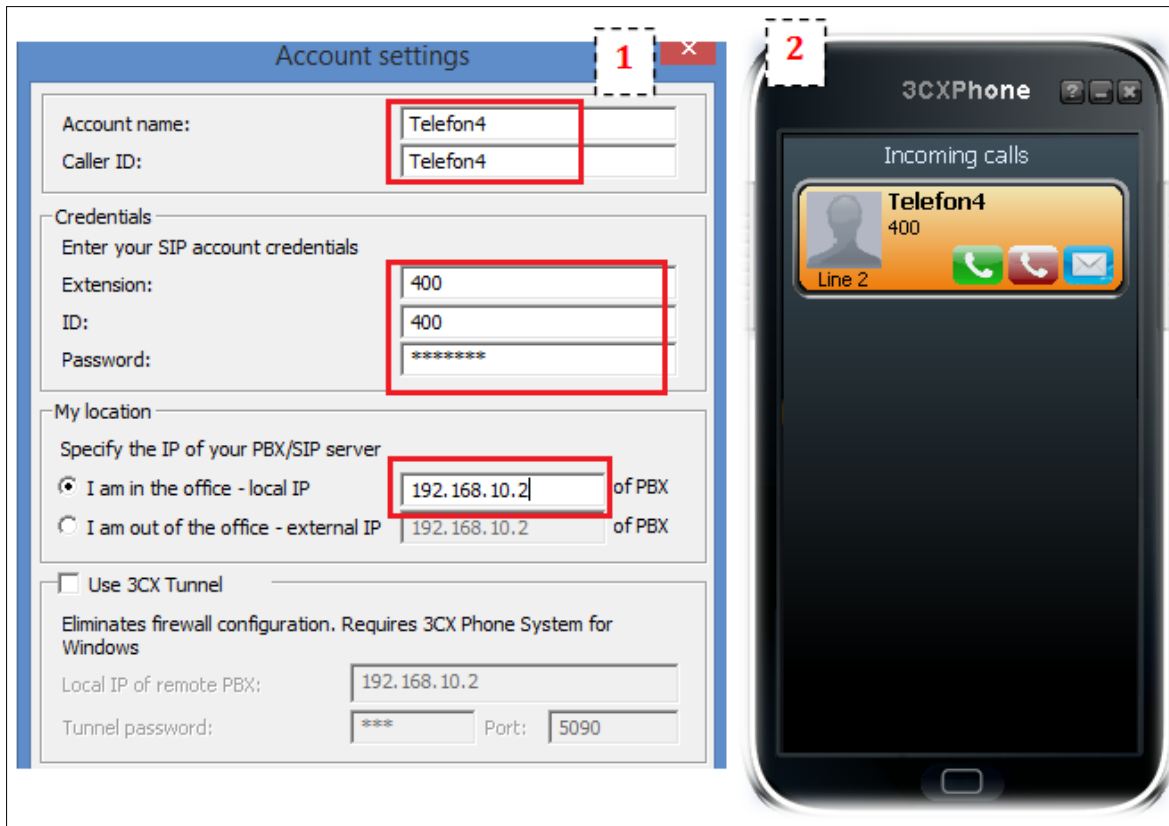
Slika 49 Kreiranje novog SIP naloga

## Instalacija softverskih telefona

Nakon kreiranja SIP VoIP naloga na serveru potrebno je instalirati VoIP (softverske) telefone. Postoji dosta ovakvih softverskih rešenja koja su besplatna, a za ovaj projekat su korišćena dva softverska telefona, i to: X-Lite i 3CX Phone. Ova dva softvera su jednostavna za instalaciju i nije potrebno to grafički predstaviti.

Za svaki VoIP telefon je potrebno napraviti bar po jedan nalog, a moguće je i više, koji će biti povezani sa istim tim nalogom na serveru. Taj postupak će biti prikazan za po jedan naloga na svakom softverskom telefonu.

Primer kreiranja naloga kod ovog softverskog telefona [76] je prikazan na slici 50.

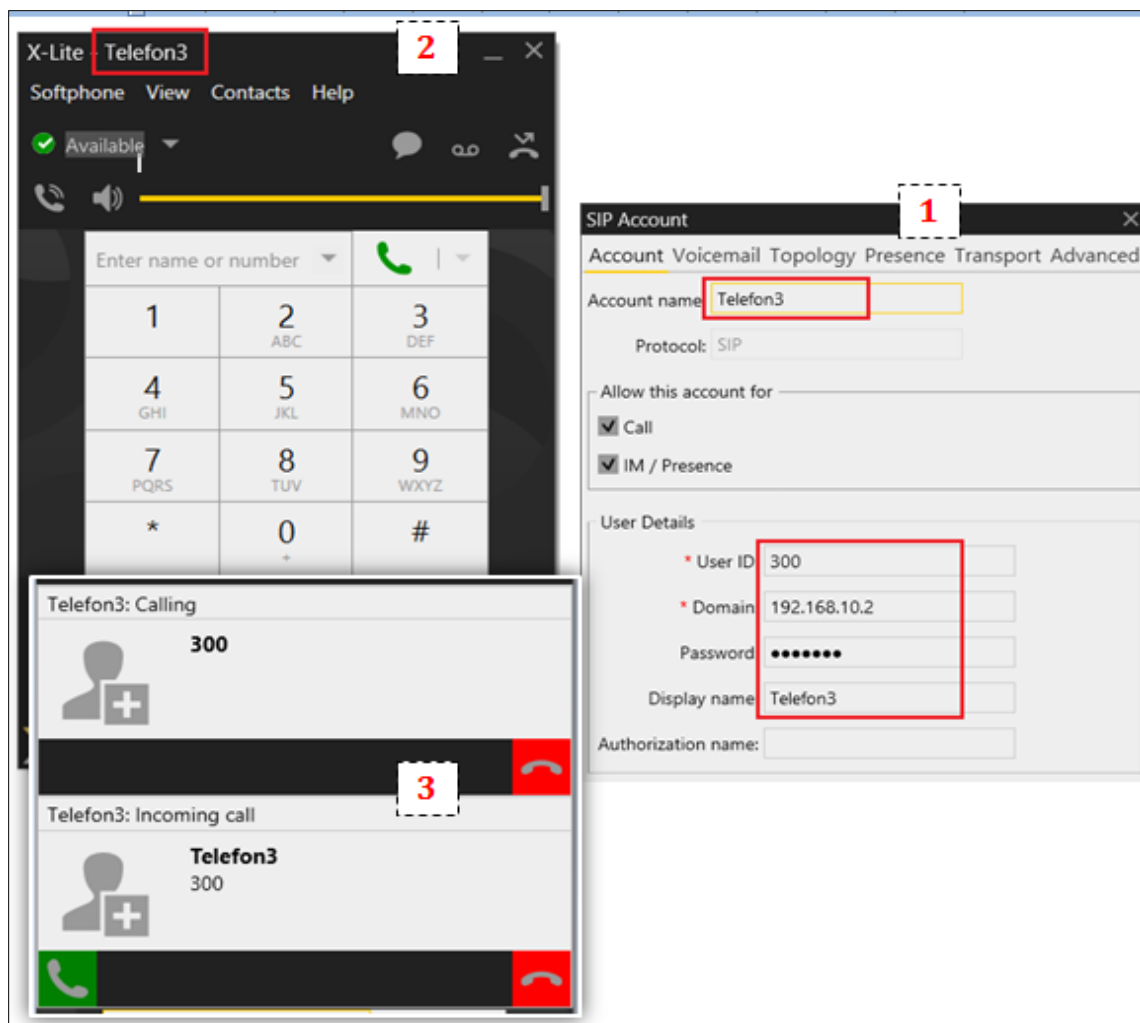


Slika 50 Dodavanje novog naloga i pozivanje samog sebe kod 3CX Phone-a

Na slici 50 (korak 1) su obeležena polja koja je potrebno ispravno popuniti, tj. podaci se moraju poklapati sa odgovarajućim podacima unetim prilikom kreiranja naloga na serveru. Na istoj slici (korak 2) prikazan je izgled ekrana prilikom pozivanja sopstvenog broja.

Na ovom softverskom telefonu moguće je kreirati više različitih naloga koji pre kreiranja moraju biti definisani na serveru, kako bi broj bio u funkciji.

X-Lite softverski telefon



Slika 51 Primer dodavanja novog naloga i pozivanja sopstvenog broja u X-Lite VoIP telefonu

Na slici 51 je prikazan postupak dodavanja novog naloga kod X-Lite softverskog telefona [77]. U prvom koraku je prikazan prozor sa označenim poljima koja je potrebno ispravno popuniti. U ovom primeru je dodat Telefon3 koji je prethodno kreiran na serveru. U koraku 2 vidimo da je nalog Telefon3 uspešno povezan sa serverom i nakon toga vršimo proveru pozivajući samoga sebe što je prikazano na istoj slici, korak 3.

Za svaki novi nalog je potrebno ponoviti isti postupak.

Primer uspostave veze između dva telefona sa različitim virtuelnih memorija

Ovde će biti slikovito prikazano međusobno pozivanje dva telefona (broja) koji su instalirani na dve različite virtuelne mašine, čime će se na najbolji način demonstrirati funkcionalnost celokupnog sistema.

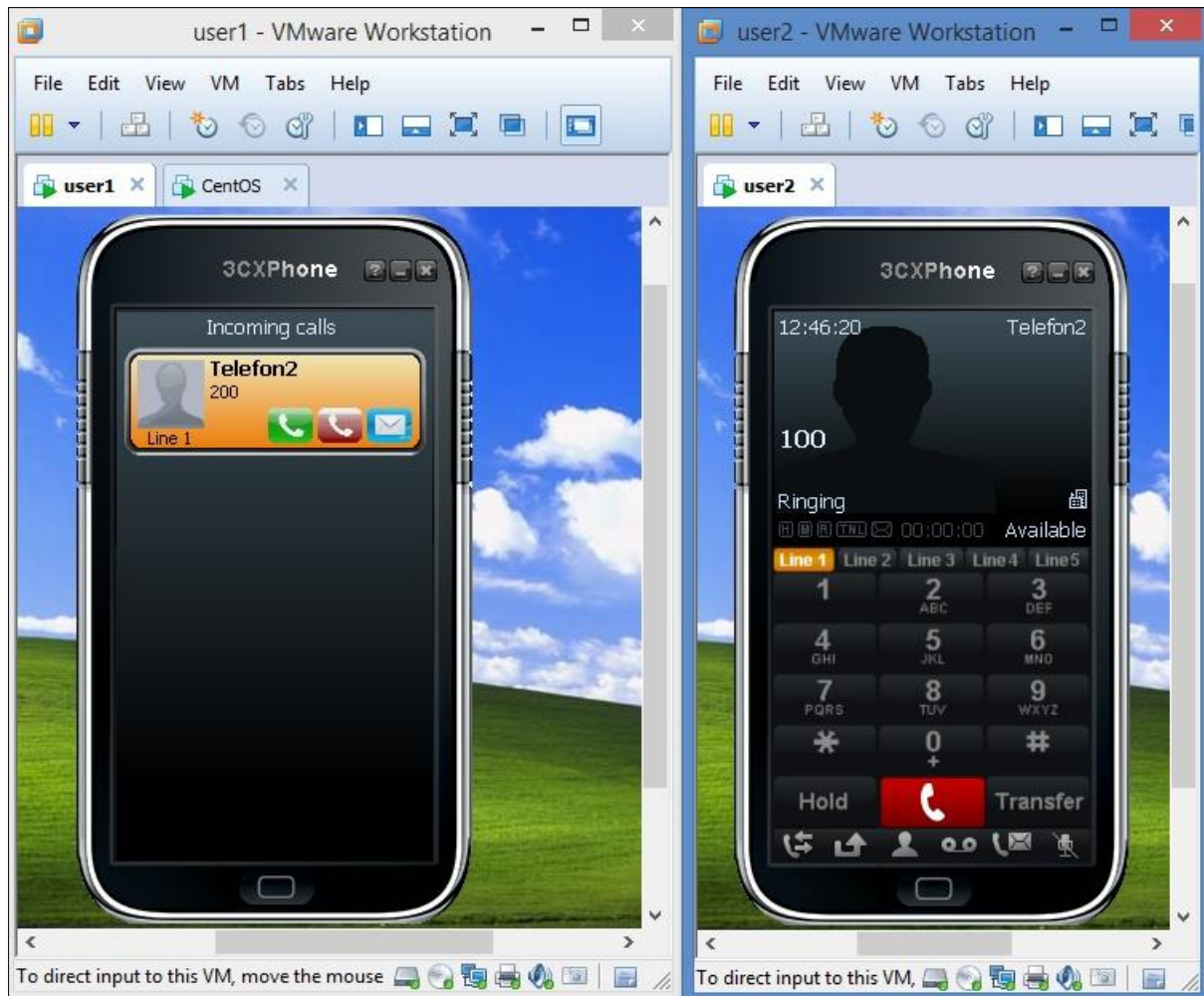
Podsećanja radi, instalirane su tri virtuelne mašine, od kojih jedna predstavlja Asterisk server, a druge dve služe da na njima instaliramo po jedan (po potrebi može i više) softverski telefon na kojima je kreiran po jedan nalog na način koji je ranije predstavljen. Na obe virtuelne mašine su instalirani 3CX Phone softverski telefoni.



Slika 52 Uspostava veze od user1 ka user2 virtuelnoj mašini

Na slici 52 dat je primer uspostave veze od virtuelne mašine user1 ka user2 virtuelnoj mašini tako što unosimo pozivni broj (200) koji je dodeljen virtuelnoj mašini user2. Na ekranu telefona kod virtuelne mašine user2 vidimo da je pristigao poziv sa pozivnog broja 100 (pozivni broj koji je dodeljen virtuelnoj mašini user1), i na nama je da izaberemo opciju odbijanje ili prihvatanje poziva.

Isti postupak, samo sada u suprotnom smeru, tj. poziv se uspostavlja od virtualne mašine user2 ka virtualnoj mašini user1 prikazan je na slici 53.



Slika 53 Uspostava veze od user2 ka user1 virtualnoj mašini

Instalacijom soft telfona uspostavljen je kompletan VoIP sistem i u realnom i u vritulenom okruženju.

## **7.2 Razvoj sistema za obradu digitalnog signala primenom Agilne metodologije**

### **7.2.1. Faza planiranja**

U fazi planiranja želeli smo da signali koji se prenesu simuliranim VoIP sistemom mogu da se dalje obrade metodama i alatima Veštačke inteligencije. Cilj je bio da kroz ovaj proces se razvije razumevanje sistema i pricipi rada DSP i njihovog ukrštanja sa AI. Kao i prethodnom slučaju neophodno je definisati u fazi planiranja planiranu infrastrukturu za implementaciju, odnosno definisati softverske pakete na kojima ćemo instalirati i koje ćemo kroistiti za realizaciju sistema. Zatim moramo uzeti u obzir da smo razvoj planirali u skladu sa agilnim metodologijama, pa je onda neophodno pre implementacije definisati sve kritične elemente agilnog pristupa razvoju [78].

Da bismo se pridržavali DevOps praksi, instalirali smo najnoviju verziju Git-a, 2.43.0. Git je distribuirani sistem za kontrolu verzija dizajniran da rukuje projektima svih veličina – od malih do vrlo velikih – brzo i efikasno. Široko se koristi za kolaborativni rad, naročito u upravljanju izvorom softverskog koda. Trenutno Git ima dominantni udeo od 89,07% na tržištu DevOps alata.

Git je distribuirani sistem kontrole verzija, koji omogućava svakom korisniku da ima kompletnu kopiju celokupnog repozitorijuma na svom računaru [79]. Ova arhitektura poboljšava otpornost na gubitak podataka i omogućava rad članovima tima čak i bez stalne internet veze. Takođe, distribuirana priroda olakšava saradnju unutar tima, jer svaki član može nezavisno da radi na svom delu projekta, a kasnije da integriše promene sa centralnim repozitorijumom [80].

Jedna od ključnih prednosti Git-a je njegova brzina i efikasnost, naročito prilikom izvođenja operacija poput grananja, spajanja i vraćanja na prethodne verzije. Brzo kreiranje i prebacivanje između grana omogućava programerima da eksperimentišu sa novim funkcionalnostima, testiraju ideje bez ometanja glavnog toka rada i brzo reaguju na eventualne greške [81]. Ovaj sistem takođe optimizuje rad na velikim projektima, gde je praćenje i integracija promena kritična za uspešan razvoj softvera.

Da bismo poboljšali vizuelizaciju, sa obzirom na to da Gitz bez GUI može biti konfuzan za ljude koji nemaju iskustvo rada u komandnom interfejsu, odabrali smo Sourcetree Git GUI [82].

Za uspostavljanje neophodne infrastrukture koristili smo GitHub kao platformu za repozitorijume, koji služe kao skladište za stvarni kod i njegovu istoriju verzija. Na svakom razvojnom računaru instaliran je Git, što je omogućilo neometanu interakciju sa centralnim repozitorijumom [83]. Programeri su koristili Git komande putem komandne linije ili grafičkih alata (GUI) kako bi obavili niz operacija kontrole verzija, uključujući zadatke poput kloniranja repozitorijuma, potvrđivanja promena, grananja, spajanja, kao i push/pull koda [84].

Na kraju, smo odabrali alate za upravljanje projektima, koji obuhvataju alate za praćenje problema, alate za upravljanje projektima i komunikacione platforme. Naposljetku smo ih integrisali se sa Git-om i uspostavili centralizovanu tačku za kolaborativne aktivnosti, poput diskusija, praćenja problema i planiranja projekata [85].

Integracija Trello platforme i Git-a pruža jasnu podjelu zadataka i mogućnost efikasnog praćenja napretka, što je ključno za uspješno vođenje projekata. Trello, kao vizuelni alat za upravljanje zadacima, omogućava timu da lako organizuje aktivnosti, postavlja rokove i prati status svakog zadatka putem kartica i listi. Kada se kombinuje sa Git-om, sistemom za kontrolu verzija, dobija se sveobuhvatan pristup planiranju i implementaciji, gde se napredak koda i zadataka sinhronizuje na jednom mestu [86].

Korišćenjem Trello-a u kooperaciji sa Git-om, timovi su imali jasnu komunikaciju i kolaboraciju. Trello je omogućio transparentnost u pogledu raspodele zadataka i trenutnog stanja svakog segmenta projekta, dok je Git pružao detaljan uvid u istoriju izmena koda. Ova kombinacija nam je olakšala identifikaciju eventualnih problema, ubrzavala rešavanje konflikata i omogućavala bržu reakciju na promene, što zajedno doprinelo povećanju efikasnosti celokupnog tima [87].

Sinergija između Trello platforme i Git-a omogućava bolju koordinaciju između različitih sektora tima, kao što su menadžment, dizajn i razvoj. Dok menadžeri mogu koristiti Trello za planiranje, praćenje i postavljanje prioriteta, programeri se oslanjaju na Git kako bi upravljali verzijama koda i pratili napredak svojih zadataka. Ovaj integrisani pristup smanjuje mogućnost komunikacionih propusta, poboljšava transparentnost procesa i omogućava brže donošenje odluka zasnovanih na realnim podacima.

MATLAB je jedan od najrasprostranjenijih softverskih alata u naučnom i inženjerskom svetu, poznat po svojoj sposobnosti da reši složene matematičke, statističke i inženjerske probleme [88]. Njegov modularni dizajn omogućava korisnicima da prošire funkcionalnosti kroz specijalizovane pakete ili "toolbox"-ove, koji su prilagođeni različitim disciplinama. Posebno značajni u savremenim istraživanjima i razvoju su alati za obradu signala i veštačku inteligenciju. Ovi alati omogućavaju analizu, obradu i interpretaciju složenih podataka, čime se otvaraju nove mogućnosti za razvoj inovativnih rešenja u oblastima kao što su telekomunikacije, biomedicina, automatizacija, robotika i mnoge druge [89]. Kroz integraciju različitih metoda i algoritama, MATLAB nudi robusno okruženje za razvoj, testiranje i implementaciju sistema koji mogu automatski da uče i prilagođavaju se dinamičnim uslovima [90].

Jedan od osnovnih paketa koji MATLAB nudi je Signal Processing Toolbox. Ovaj toolbox obuhvata bogat skup funkcija i algoritama koji omogućavaju analizu, obradu i vizualizaciju signalnih podataka. Neke od ključnih karakteristika ovog alata uključuju [91]:

- Filtriranje i transformacije: Kroz razne vrste filtera (npr. niskopropusni, visokopropusni, pojasni i notch filteri), korisnici mogu izolovati relevantne

frekvencijske komponente signala. Takođe, Fourierove transformacije, uključujući brzu Fourierovu transformaciju (FFT), omogućavaju prelazak iz vremenskog u frekvencijski domen radi analize spektra signala.

- Analiza spektra i vremenski-frekvencijska analiza: Alati za spektrogram, analiza valletskih transformacija i drugih metoda pružaju mogućnost detaljnog ispitivanja signala, otkrivanja latentnih obrazaca i promena tokom vremena
- Dizajn i simulacija filtera: Sa ugrađenim funkcijama za dizajn linearnih i nelinearnih filtera, korisnici mogu brzo testirati i prilagoditi filtere prema specifičnim zahtevima aplikacije.

Mnoge od ovih funkcija su optimizovane za rad u realnom vremenu, što omogućava implementaciju u sistemima za automatsko praćenje, dijagnostiku i kontrolu.

Pored Signal Processing Toolbox-a, MATLAB nudi i DSP System Toolbox koji omogućava detaljniju analizu i modeliranje digitalnih signalnih sistema. Ovaj toolbox je posebno koristan za inženjere koji se bave dizajnom, verifikacijom i implementacijom algoritama na realnom hardveru. On pruža mogućnosti za simulaciju, generisanje signala i analizu performansi, čime se omogućava brža iteracija prilikom razvoja sistema.

U oblasti veštačke inteligencije, MATLAB je izgradio snažan eko sistem alata koji podržavaju razvoj, treniranje i evaluaciju modela mašinskog učenja i dubokog učenja. Glavni alati u ovom segmentu uključuju [92]:

Deep Learning Toolbox: Ovaj toolbox omogućava kreiranje i treniranje dubokih neuronskih mreža koristeći unapred definisane arhitekture ili prilagođene modele. Korisnici mogu da implementiraju konvolucione, rekurentne i druge vrste neuronskih mreža, a podrška za GPU akceleraciju omogućava brže treniranje modela. Vizuelni alati poput Deep Network Designer-a omogućavaju interaktivno kreiranje i prilagođavanje arhitektura, što značajno ubrzava proces razvoja [93].

Statistics and Machine Learning Toolbox: Ovaj paket je osmišljen za primenu statističkih metoda i algoritama mašinskog učenja. On obuhvata tehnike kao što su regresija, klasifikacija, klaster analiza, smanjenje dimenzionalnosti i validacija modela. Posebno je koristan za analizu velikih skupova podataka, otkrivanje skrivenih obrazaca i prediktivno modeliranje.

Reinforcement Learning Toolbox: Za aplikacije koje zahtevaju adaptivno ponašanje kroz interakciju sa dinamičnim okruženjem, Reinforcement Learning Toolbox omogućava razvoj modela koji uče putem nagrade i kazne. Ovaj alat se sve češće koristi u robotici, automatizovanim sistemima i simulacijama, gde je potrebno optimizovati akcije na osnovu povratnih informacija iz okruženja [94].

MATLAB alati za veštačku inteligenciju lako se integrišu sa ostalim funkcijama unutar MATLAB okruženja, što omogućava kompleksne analize. Na primer, podaci obrađeni pomoću

Signal Processing Toolbox-a mogu biti direktno korišćeni za treniranje modela u Deep Learning Toolbox-u, čime se ostvaruje sinergija između obrada signala i veštačke inteligencije [95]. Ovakva integracija omogućava razvoj sistema koji automatski mogu prepoznati i klasifikovati obrasce u signalu, kao što su govor, biomedicinski signali ili signali iz industrijskih procesa [96].

MATLAB zajednica aktivno doprinosi razvoju ovih alata, što znači da se novi algoritmi, unapređenja i primene redovno uvode. Time MATLAB ostaje relevantan alat u savremenim istraživanjima i industrijskim primenama, pružajući sve obuhvatno okruženje za razvoj inovacija u oblasti veštačke inteligencije.

Sinergija između alata za obradu signala i onih za veštačku inteligenciju omogućava kreiranje naprednih sistema koji mogu samostalno da uče i prilagođavaju se promenama u okruženju. Na primer, u oblasti biomedicinske dijagnostike, podaci prikupljeni putem elektroencefalograma (EEG) mogu biti obrađeni pomoću Signal Processing Toolbox-a, čime se filtriraju šumovi i ističu značajne frekvencijske komponente. Nakon toga, obrađeni podaci mogu biti korišćeni za treniranje dubokih neuronskih mreža u Deep Learning Toolbox-u, što omogućava preciznu detekciju anomalija ili predviđanje epileptičnih napada [97].

Sličan pristup primenljiv je u telekomunikacijama, gde se signali primljeni iz antena obrađuju radi eliminacije interferencija, a zatim analiziraju pomoću modela mašinskog učenja kako bi se optimizovao prenos podataka i poboljšao kvalitet veze. U industrijskim aplikacijama, kombinacija ovih alata omogućava razvoj sistema za prediktivno održavanje, gde se vibracije i drugi signali prikupljaju u realnom vremenu, obrađuju i analiziraju kako bi se predvideli kvarovi mašina.

Integracija ovih alata takođe omogućava razvoj sofisticiranih sistema za prepoznavanje govora i obrade prirodnog jezika. Podaci o akustičkim signalima se filtriraju i obrađuju, dok se modelima dubokog učenja dodeljuju zadaci prepoznavanja reči, tonaliteta i emocionalnog naboja. Rezultati ovakvih analiza imaju široku primenu u razvoju pametnih asistenata, automatizaciji korisničke podrške i u drugim interaktivnim sistemima.

MATLAB predstavlja moćno okruženje za razvoj rešenja u oblastima obrade signala i veštačke inteligencije. Kroz specijalizovane toolbox-e, kao što su Signal Processing Toolbox, DSP System Toolbox, Deep Learning Toolbox, Statistics and Machine Learning Toolbox i Reinforcement Learning Toolbox, korisnici dobijaju sveobuhvatne alate za analizu, simulaciju i implementaciju naprednih algoritama [98]. Ovi alati omogućavaju ne samo izolovanu obradu i analizu podataka, već i njihovu integraciju u sisteme koji automatski uče i prilagođavaju se, čime se otvaraju nove mogućnosti u industriji, medicini, telekomunikacijama i mnogim drugim oblastima [99].

Kombinacija tehnologija za obradu signala i veštačke inteligencije omogućava razvoj sistema koji su sposobni da se nose sa kompleksnim i promenljivim uslovima u realnom svetu

[100]. Fleksibilnost MATLAB okruženja, uz kontinuirane nadogradnje i podršku aktivne zajednice, čini ove alate idealnim izborom za istraživače i inženjere koji žele da unaprede svoje projekte i implementiraju inovativna rešenja [101]. Na kraju, integrisani pristup koji MATLAB nudi doprinosi bržem razvoju i efikasnijoj primeni tehnologija, što je ključno za konkurentnost u savremenom tehnološkom okruženju [102].

Scrum je agilni okvir koji se široko primenjuje za razvoj složenih proizvoda i usluga, a njegov glavni cilj jeste povećanje transparentnosti, inspekcije i adaptacije kroz iterativne i inkrementalne procese. Time se omogućava brza reakcija na promene i kontinuirano unapređenje radnih procesa. U ovom okruženju, jasno definisane uloge, događaji i artefakti igraju ključnu ulogu u organizaciji i realizaciji projekta, čime se omogućava stvaranje fleksibilnih i samostalnih timova usmerenih na postizanje konkretnih rezultata [103].

Pre početka same implementacije, neophodno je sprovesti pripremne aktivnosti koje uključuju obuku celokupnog tima o osnovnim principima i praksama Scruma. Svi članovi tima moraju se upoznati sa ulogama Scrum Mastera, Product Ownera i razvojnog tima, što omogućava zajedničko razumevanje metodologije. Ujedno, važno je jasno definisati uloge unutar tima, tako da se izabere kvalifikovani Scrum Master koji će facilitirati proces i uklanjati prepreke, dok Product Owner postavlja prioritete i vodi razvoj proizvoda. Jasno definisana vizija projekta i dugoročni ciljevi dodatno doprinose usklađivanju svih članova tima sa očekivanjima [104].

Jedan od ključnih elemenata Scruma je kreiranje Product Backloga, odnosno sveobuhvatne liste funkcionalnosti, zahteva, poboljšanja i ispravki koje treba implementirati. U saradnji sa Product Ownerom, prikupljaju se svi zahtevi i korisničke priče, koje se potom rangiraju prema njihovoj vrednosti i značaju za korisnika i poslovanje. Svaka stavka mora biti dovoljno detaljno opisana, često uz korišćenje tehnika poput „user story“ i „acceptance criteria“, kako bi razvojni tim imao jasnu sliku o očekivanjima i ciljevima [105].

Nakon uspostavljanja Product Backloga, sledi faza planiranja sprinta, koja predstavlja početak svakog iterativnog ciklusa. Tokom ove faze, tim, uz učešće Product Ownera, definiše jasan cilj sprinta i odabire stavke iz Product Backloga koje su realno ostvarive u predviđenom vremenskom okviru, obično između dve i četiri nedelje. Stavke se zatim razlažu na manje zadatke, što omogućava preciznije planiranje i efikasnije praćenje napretka tokom sprinta [106].

Tokom trajanja sprinta, tim se fokusira na realizaciju definisanih zadataka i postizanje zacrtanih ciljeva. Članovi tima samostalno upravljaju svojim zadacima i aktivno saraduju na rešavanju eventualnih prepreka, koristeći vizuelne alate poput Scrum tabli ili softverskih rešenja za praćenje zadataka. Pored toga, dnevni sastanci poznati kao Daily Scrum traju oko petnaest minuta, tokom kojih svaki član iznosi postignuća prethodnog dana, planove za tekući dan i eventualne izazove, čime se omogućava brza sinhronizacija i efikasno rešavanje problema [107].

Na kraju svakog sprinta, tim organizuje sastanak pod nazivom Sprint Review, na kome prezentuje završene zadatke Product Owneru i ostalim zainteresovanim stranama. Ovaj sastanak omogućava dobijanje povratnih informacija, što doprinosi proveru ostvarenja vrednosti i prilagođavanju proizvoda prema potrebama korisnika. Nakon pregleda sprinta, sledi sastanak Sprint Retrospective, gde tim detaljno analizira svoj rad, identifikuje uspešne prakse, kao i izazove i prepreke koje su se pojavile tokom sprinta. Na osnovu ove analize definišu se konkretne akcije za unapređenje procesa u narednim ciklusima.

Završetak sprinta obeležava prelazak u novu fazu pripreme za naredni ciklus. U ovom periodu, Product Backlog se revidira i ažurira na osnovu dobijenih povratnih informacija i novih saznanja, pri čemu se dodaju nove stavke ili se menja prioritet postojećih zadataka. Nakon toga, proces planiranja novog sprinta pokreće se ponovo, čime se uspostavlja kontinuirani ciklus rada i stalnog unapređenja.

Kontinuirano unapređenje predstavlja temelj uspešne implementacije Scruma. Tim redovno prati ključne metrike, kao što su burn-down grafikoni, velocity i broj završenih zadataka, čime se identifikuju eventualni problemi i preduzimaju neophodne korektivne mere. Fleksibilnost i spremnost na promene, zajedno sa otvorenom komunikacijom i aktivnom saradnjom, doprinose stvaranju pozitivne radne dinamike i omogućavaju kontinuirani razvoj tima [108].

Za podršku agilnom procesu, korišćenje adekvatnih alata i tehnologija je od izuzetne važnosti. Softverska rešenja poput Jira, Trello ili Asana omogućavaju efikasnu organizaciju zadataka i praćenje napretka, dok alati za online komunikaciju, kao što su Slack ili Microsoft Teams, olakšavaju brzu razmenu informacija i rešavanje prepreka u realnom vremenu [109]. Osim toga, dokumentovanje svih procesa i odluka doprinosi transparentnosti i omogućava lakše praćenje promena tokom celokupnog projekta [110].

Zaključno, implementacija Scrum metodologije predstavlja transformativan pristup u razvoju proizvoda, koji omogućava timovima da budu fleksibilni, efikasni i usmereni na isporuku maksimalne vrednosti korisnicima [111]. Kroz jasno definisane uloge, iterativne procese i redovne evaluacije, Scrum doprinosi boljoj saradnji i bržem prilagođavanju dinamičnom poslovnom okruženju [112]. Kontinuirani ciklus planiranja, implementacije, pregleda i retrospektive osigurava stalno unapređenje radnih procesa, čime se postavlja čvrsta osnova za dugoročni uspeh i održavanje agilnosti u radu [113-117].

### **7.2.1. Faza implementacije**

Kao i prilikom razvoja VoIP prvi deo faze implementacije se odnosi na instalaciju infrastrukture i softvera neophodnog za razvoj laboratorije. U ovom slučaju deo alata kao što su Trello i GitHub se nalaze na cloudu kao online sistemi i ne zahtevaju posebnu instalaciju. Od neophodne infrastrukture je neophodno postaviti sistem za praćenje verzija i nakon njegove

instalacije i povezivanja sa repozitorijumima neophodno je definisati osnovne zadatke u skladu sa agilnom metodologijom i iste postaviti u sistemima za upravljanje projektima.

Mi smo započeli proces instalacije Git-a na našoj lokalnoj mašini tako što smo posetili zvanični sajt `git-scm.com` kako bismo preuzeli odgovarajući instalacioni fajl prilagođen našem operativnom sistemu. Na Windows platformi smo skinuli instalacioni fajl, pokrenuli instalaciju i pratili korake koje je vodio instalacioni čarobnjak. Tokom instalacije smo obratili pažnju na opcije koje su nam se nudile, kao što su izbor terminala – odlučili smo se za Git Bash – kao i integracija sa Windows komandnom linijom. Ukoliko smo radili na macOS-u, koristili smo Homebrew unosom komande „`brew install git`“, a na Linux distribucijama smo se oslonili na paket menadžere, na primer komandom „`sudo apt-get install git`“ na Debian/Ubuntu sistemima. Time smo osigurali da Git bude ispravno instaliran na našoj mašini i spreman za dalji rad.

Nakon završene instalacije, pristupili smo konfiguraciji Git-a postavljanjem osnovnih globalnih podešavanja. Otvorili smo terminal i uneli komande

```
git config --global user.name 'Naše Ime' i  
git config --global user.email 'nas.email@primer.com',
```

čime smo definisali identitet koji će biti vezan za sve commit-ove koje smo kasnije izvršili. Pored toga, podesili smo podrazumevani editor, kao i ostale opcije poput boja u izlaznim porukama terminala, čime smo osigurali da rad sa Git-om bude prilagođen našem radnom okruženju. Na ovaj način smo kreirali čvrstu osnovu koja je omogućila dalji nesmetan rad sa repozitorijumima.

Sledeći korak koji smo preduzeli bio je generisanje SSH ključeva, što je predstavljalo ključnu aktivnost za uspostavljanje bezbedne veze između naše lokalne mašine i udaljenog repozitorijuma na GitHub-u. Pokrenuli smo terminal i izvršili komandu

```
ssh-keygen -t ed25519 -C 'nas.email@primer.com'.
```

U slučaju da naš operativni sistem nije podržavao ed25519 algoritam, odlučili smo se za RSA unosom „`ssh-keygen -t rsa -b 4096 -C 'nas.email@primer.com'`“. Pri tom smo prihvatili podrazumevanu lokaciju za čuvanje ključeva, a po potrebi smo dodali i lozinku za dodatnu sigurnost.

```
Nakon generisanja ključa, kopirali smo sadržaj javnog ključa pomoću komande  
cat ~/.ssh/id_ed25519.pub
```

(ili odgovarajuće, u zavisnosti od generisanog tipa ključa) i zalepili ga u podešavanja našeg GitHub naloga pod sekcijom SSH and GPG keys. Na ovaj način smo omogućili sigurnu autentifikaciju i olakšali buduće operacije push i pull bez potrebe za stalnim unošenjem korisničkih podataka.

Kada smo završili sa instalacijom i osnovnom konfiguracijom, krenuli smo ka kreiranju i upravljanju repozitorijumima. Ukoliko smo želeli da započnemo novi projekat, inicijalizovali smo lokalni repozitorijum u odgovarajućem direktorijumu pomoću komande

```
git init.
```

Time smo kreirali skrivenu .git fasciklu koja je sadržavala sve metapodatke potrebne za praćenje istorije izmena. Nakon toga, dodavali smo fajlove komandom

```
git add.
```

kako bismo pripremili sve promene za commit, a potom smo izvršavali commit-ove unosom komande

```
git commit -m 'Poruka commit-a'
```

čime smo sačuvali stanje repozitorijuma u određenom trenutku. U slučaju da smo želeli da radimo na već postojećem projektu, klonirali smo repozitorijum sa GitHub-a pomoću komande „git clone URL-repoizitorijuma“, čime smo kreirali lokalnu kopiju cele istorije projekta. Ovim pristupom smo omogućili brzu sinhronizaciju sa udaljenim serverom i kontinuirani rad na projektu.

Posle inicijalizacije lokalnog repozitorijuma, povezali smo ga sa udaljenim repozitorijumom na GitHub-u. Ukoliko smo repozitorijum kreirali lokalno, dodali smo udaljenu vezu unosom komande

```
git remote add origin git@github.com:korisnicko_ime/naziv_repoizitorijuma.git
```

(ili alternativno, koristeći HTTPS putanju). Na ovaj način smo kreirali vezu pod imenom „origin“, što je podrazumevano ime koje Git koristi za dalju komunikaciju sa udaljenim repozitorijumom. Proverili smo uspešnost povezivanja komandom „git remote -v“, koja nam je prikazala sve definisane udaljene adrese. Ova konfiguracija je bila od ključnog značaja jer nam je omogućila da kasnije, prilikom sinhronizacije izmena, jednostavno izvršavamo komande „git push“ i „git pull“ kako bismo održavali konzistentnost između lokalne i udaljene verzije koda.

Nakon što smo uspostavili vezu između lokalnog i udaljenog repozitorijuma, krenuli smo sa svakodnevnim operacijama koje su obeležavale naš rad sa Git-om. Redovno smo vršili commit-ove lokalnih izmena, čime smo sačuvali promene u istoriji projekta, a zatim smo ih sinhronizovali sa GitHub repozitorijumom komandom „git push origin master“ (ili „main“, u zavisnosti od naziva glavne grane). U slučajevima kada smo primetili da se na udaljenom repozitorijumu nalaze nove promene koje nisu bile prisutne u našoj lokalnoj kopiji, prvo smo izvršavali komandu „git pull origin master“ kako bismo sinhronizovali repozitorijum i izbegli eventualne konflikte. Ukoliko su se pojavili konflikti prilikom spajanja različitih verzija koda, koristili smo dostupne Git alate za ručno rešavanje istih, nakon čega smo izvršavali dodatni commit kako bismo potvrdili konačno rešenje. Na ovaj način smo osigurali da je naša istorija promena uvek pregledna i konzistentna, a rad na projektu tekao je bez većih prepreka.

Tokom celog procesa, trudili smo se da održavamo jasnu i preglednu istoriju commit-ova pisanjem deskriptivnih poruka koje su jasno objašnjavale izvršene izmene. Redovno smo koristili komande „git status“ i „git log“ kako bismo imali uvid u trenutno stanje repozitorijuma i istoriju svih izvršenih commit-ova. Ovaj pristup nam je omogućio da pratimo napredak projekta i brzo identifikujemo eventualne greške ili neusklađenosti. Takođe smo se trudili da prilikom uvođenja novih funkcionalnosti kreiramo posebne grane, što je olakšavalo rad na eksperimentalnim ili dodatnim funkcijama bez narušavanja stabilnosti glavne grane. Nakon što smo završili razvoj, te grane smo spajali sa glavnom granom putem pull request-ova, čime smo omogućili pregled i eventualnu reviziju koda od strane celog tima.

```
MINGW64:/d/04_Predmeti/02_GitPredmeti/05_DigObradaAI
Edis@DESKTOP-E9Q0QSF MINGW64 /d/04_Predmeti/02_GitPredmeti/05_DigObradaAI ((9a4c
c71bc...))
$ git log
commit 9a4cc71bc5f5299cb104be855377c6a488d3344e (HEAD)
Merge: fbce069a7 d9b7d82ca
Author: Emin Hamzagic <63869149+EminHamzagic@users.noreply.github.com>
Date: Sat Jan 4 16:58:38 2025 +0100

Merge pull request #10 from EminHamzagic/dzemilova_grana

Smanjenje šuma (Dodavanje šuma + različiti filteri)

commit d9b7d82ca083135c8b50a5c4968d0f20ae1ca9ce (DigObrada/dzemilova_grana)
Author: Emin Hamzagic <hamzagicemin4@gmail.com>
Date: Sat Jan 4 16:57:13 2025 +0100

made some adjustments

commit cf9209c6052893c5c244b7ede4164719bae0f430
Author: Džemil Manić <dzemilmanic@hotmail.com>
Date: Fri Jan 3 17:31:30 2025 +0100

added filters

commit d3127646bd7a543f221b4cbdbc6fb560d21ed2a2
Author: Džemil Manić <dzemilmanic@hotmail.com>
...skipping...
commit 9a4cc71bc5f5299cb104be855377c6a488d3344e (HEAD)
Merge: fbce069a7 d9b7d82ca
Author: Emin Hamzagic <63869149+EminHamzagic@users.noreply.github.com>
Date: Sat Jan 4 16:58:38 2025 +0100

Merge pull request #10 from EminHamzagic/dzemilova_grana

Smanjenje šuma (Dodavanje šuma + različiti filteri)

commit d9b7d82ca083135c8b50a5c4968d0f20ae1ca9ce (DigObrada/dzemilova_grana)
Author: Emin Hamzagic <hamzagicemin4@gmail.com>
Date: Sat Jan 4 16:57:13 2025 +0100

made some adjustments

commit cf9209c6052893c5c244b7ede4164719bae0f430
Author: Džemil Manić <dzemilmanic@hotmail.com>
Date: Fri Jan 3 17:31:30 2025 +0100

added filters

commit d3127646bd7a543f221b4cbdbc6fb560d21ed2a2
Author: Džemil Manić <dzemilmanic@hotmail.com>
Date: Thu Jan 2 21:20:40 2025 +0100

something

commit fbce069a7de64cfad9a7eaac978b6d27d6296eb6
Author: Emin Hamzagic <63869149+EminHamzagic@users.noreply.github.com>
Date: Wed Jan 1 13:34:55 2025 +0100
```

## Slika 54 Git log na lokalnom računaru

Na kraju, posvetili smo značaj kontinuiranog unapređenja naših procesa rada sa Git-om. Redovno smo pratili ključne metrike, poput burn-down grafikona i velocity-ja, kako bismo procenili efikasnost našeg rada i identifikovali eventualne probleme. Uvek smo bili spremni da prilagodimo naše radne procese i ažuriramo alate koje smo koristili, prateći najnovije funkcionalnosti i najbolje prakse iz sveta Git-a i GitHub-a. Takođe smo se oslanjali na bogatstvo resursa koje su pružale zvanična dokumentacija, forumi i zajednice programera, što nam je omogućilo brzo rešavanje tehničkih problema i konstantan napredak u znanju. Ovim pristupom smo stvorili efikasan radni tok koji je obezbedio ne samo kvalitetan razvoj koda, već i bolju kolaboraciju unutar tima, uz visok nivo transparentnosti i kontrole nad projektom.

Gledajući unazad, možemo zaključiti da je naš proces instalacije, podešavanja i integracije Git-a sa GitHub repozitorijumom bio temeljito planiran i uspešno realizovan. Svaki korak koji smo preduzeli – od preuzimanja instalacionih fajlova, preko detaljne konfiguracije, generisanja sigurnosnih ključeva, inicijalizacije lokalnog repozitorijuma, do uspostavljanja veze sa udaljenim repozitorijumom – doprineo je stvaranju stabilne i efikasne radne sredine. Ovaj metodički pristup omogućio nam je da bez problema sinhronizujemo lokalne izmene sa GitHub-om, što je direktno uticalo na kvalitet i organizaciju celokupnog projekta. Kroz dosledno praćenje svih koraka, uspeali smo da postignemo visok nivo automatizacije i sigurnosti u radu sa verzionisanjem koda, čime smo dodatno unapredili timsku saradnju i efikasnost razvoja softverskih rešenja.

Iskustvo koje smo stekli prilikom instalacije i podešavanja Git-a pokazalo se kao neprocenjivo za dalji rad na projektima. Uspostavljanje jasne strukture i korišćenje najboljih praksi u upravljanju izmenama koda omogućilo nam je da se lako prilagodimo dinamičnim zahtevima modernog softverskog razvoja. Na taj način smo stvorili čvrstu osnovu koja nam je omogućila da se usmerimo na razvoj novih funkcionalnosti i poboljšanje postojećih, istovremeno održavajući visok stepen kontrole nad celokupnim procesom razvoja. Ovaj sveobuhvatni pristup je značajno doprineo našem uspehu i osigurao da je svaki naredni projekat realizovan sa maksimalnom efikasnošću i transparentnošću.

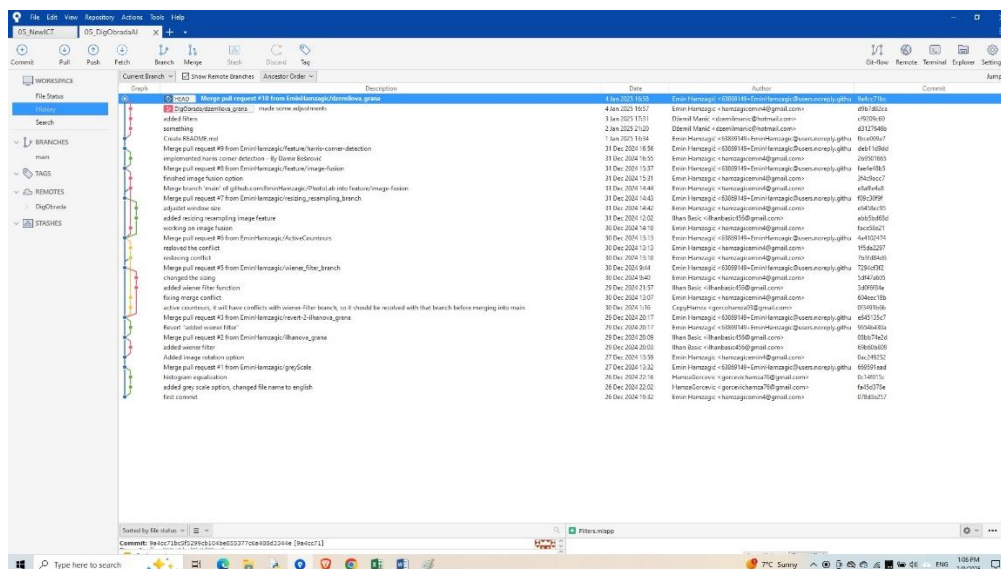
SourceTree je besplatan grafički interfejs razvijen od strane Atlassian-a, namenjen za rad sa Git i Mercurial repozitorijumima. Njegov intuitivan dizajn omogućavao je korisnicima da bez problema prate sve aspekte procesa verzionisanja, od kreiranja grana, preko izvršavanja commit-ova, merge-ovanja i rebase-ovanja, do pregleda istorije izmena. Cilj ovog alata bio je da omogućiti i početnicima i iskusnim programerima da se oslobode kompleksnosti komandne linije, pružajući vizuelan prikaz svih operacija i čime se olakšava upravljanje kodom u razvojnim projektima.

Razvoj SourceTree-ja započeo je sa namerom da pojednostavi rad sa verzionisanjem i da se ponudi alternativa tradicionalnom korišćenju komandne linije. Prva verzija je lansirana kao

alat koji je omogućavao efikasnu kontrolu nad repozitorijumima, a Atlassian je kontinuirano radio na unapređenju njegovih funkcionalnosti. Svaka nova verzija donosila je integraciju sa najnovijim verzijama Git-a, poboljšanja u korisničkom interfejsu i dodatne opcije za rad sa udaljenim repozitorijumima, kao što su Bitbucket i GitHub, čime je alat postajao sve relevantniji u modernom razvoju softvera.

Kroz svoju evoluciju, SourceTree je postao ključni alat u radnom arsenalu mnogih developera, naročito u profesionalnom okruženju gde je brza i efikasna kontrola verzija od izuzetne važnosti. Alat je omogućio korisnicima da lako vizuelizuju tok rada, koristeći grafičke prikaze i interaktivne elemente koji pojednostavljaju identifikaciju i rešavanje potencijalnih konflikata u kodu. Takođe, implementacija funkcija poput drag-and-drop interfejsa i automatizovanih procedura za sinhronizaciju sa udaljenim repozitorijumima dodatno je unapredila svakodnevni rad, čineći upravljanje projektima transparentnijim i intuitivnijim.

Sa kontinuiranim ažuriranjima i uvođenjem novih funkcionalnosti, SourceTree je evoluirao iz jednostavnog GUI alata u kompleksno rešenje koje se savršeno integriše u širi Atlassian ekosistem. Ova integracija omogućila je korisnicima da sinhronizuju svoje radne procese sa drugim Atlassian alatima, što je dodatno poboljšalo saradnju unutar timova i omogućilo brzu implementaciju promena u dinamičnom okruženju razvoja softvera. Današnji SourceTree predstavlja moćan alat koji ne samo da olakšava svako dnevne operacije sa repozitorijumima, već i doprinosi povećanju produktivnosti, pružajući pouzdano rešenje za rad sa verzionisanim kodom u modernim projektima.



Slika 55 Sourcetree kao GUI lokalnog Git-a

Trello je web-bazirani alat za upravljanje projektima koji je revolucionisao način na koji pojedinci i timovi organizuju svoje zadatke, ideje i projekte. Razvijen kao intuitivan sistem zasnovan na metodologiji Kanban, Trello koristi vizuelnu strukturu tabli, lista i kartica kako bi omogućio korisnicima da lako prate napredak, raspoređuju prioritete i komuniciraju unutar tima. Kroz godine, ovaj alat je postao sinonim za fleksibilnost i jednostavnost u organizaciji rada, pružajući rešenja prilagođena kako za lične zadatke, tako i za složene timske projekte.

U svojim počecima, Trello je zamišljen kao sredstvo za pojednostavljenje procesa upravljanja zadacima, naročito u kontekstu agilnog razvoja softvera. Osnivači su želeli da kreiraju platformu koja bi omogućila svim korisnicima – bez obzira na tehničku stručnost – da vizuelno mapiraju tok rada i da lako prate sve promene u realnom vremenu. Ubrzo nakon lansiranja, Trello je stekao veliku popularnost, jer je omogućio ne samo praćenje zadataka, već i celokupnu organizaciju projekata kroz jednostavan i intuitivan interfejs. Ova ideja se pokazala izuzetno privlačnom jer je uklonila potrebu za komplikovanim softverom za upravljanje projektima, nudeći rešenje koje je dostupno širokom spektru korisnika.

Centralni element Trello-a je tabla, koja predstavlja virtuelnu radnu površinu na kojoj se organizuju svi aspekti projekta. Svaka tabla je podeljena na liste, a unutar tih lista nalaze se kartice koje predstavljaju pojedinačne zadatke, ideje ili informacije. Ova hijerarhijska struktura omogućava korisnicima da na jednom mestu vide sve aspekte projekta, od početnih ideja do finalnih zadataka. Kartice se mogu lako premještati između lista, što olakšava praćenje napretka i prilagođavanje prioriteta u realnom vremenu. Na primer, zadatak koji je označen kao "U toku" može se jednostavno prebaciti na listu "Završeno" čim bude kompletiran, što omogućava jasnu i transparentnu evidenciju rada.

Jedna od ključnih karakteristika Trello-a jeste njegova prilagodljivost. Korisnici mogu dodavati različite oznake, rokove, komentare, priloge i checklist-e direktno na kartice, čime se svaki zadatak obogaćuje dodatnim informacijama. Ove opcije omogućavaju detaljnu evidenciju svih aspekata zadatka, uključujući napomene, povezane datoteke i komunikaciju između članova tima. Takođe, mogućnost integracije sa drugim alatima i servisima – kao što su Google Drive, Slack, Dropbox i mnogi drugi – dodatno povećava funkcionalnost Trello-a, omogućavajući korisnicima da sve potrebne informacije i komunikaciju centralizuju na jednom mestu. Upravo zbog ovih karakteristika, Trello se često koristi ne samo u poslovnom okruženju, već i u ličnoj organizaciji, kao što su planiranje putovanja, vođenje budžeta ili čak kao dnevnik ideja.

Vizuelni dizajn Trello-a zasniva se na minimalističkom pristupu koji stavlja akcenat na jednostavnost i preglednost. Interfejs je dizajniran tako da korisniku odmah pruži jasan uvid u sve ključne informacije bez pretrpanosti detaljima. Boje i oznake se koriste kako bi se kartice razlikovale po prioritetu ili statusu, dok se drag-and-drop funkcionalnost koristi za jednostavno reorganizovanje sadržaja. Ova vizuelna jasnoća doprinosi bržem usvajanju alata, čak i za korisnike

koji nisu navikli na kompleksne sisteme za upravljanje projektima. Upravo ta kombinacija estetske privlačnosti i funkcionalnosti čini Trello alatom koji je prepoznatljiv na prvi pogled i lako se prilagođava različitim stilovima rada.

Kroz razvoj, Trello je prošao kroz brojne iteracije koje su unapređivale njegovu funkcionalnost i proširivost. Uvođenjem dodataka, poznatih kao "Power-Ups", korisnicima je omogućeno proširenje osnovnih funkcija alata, što obuhvata napredne mogućnosti za praćenje vremena, analitiku, integraciju sa CRM sistemima, kao i specifične alate za timsku saradnju. Ovi Power-Ups omogućavaju da se Trello prilagodi specifičnim potrebama korisnika, bilo da se radi o vođenju marketingških kampanja, organizaciji događaja ili razvoju softverskih rešenja. Power-Ups su ključni faktor u razvoju Trello-a, jer omogućavaju modularnost i fleksibilnost, čime alat postaje dinamičan i spreman da odgovori na sve izazove modernog poslovanja.

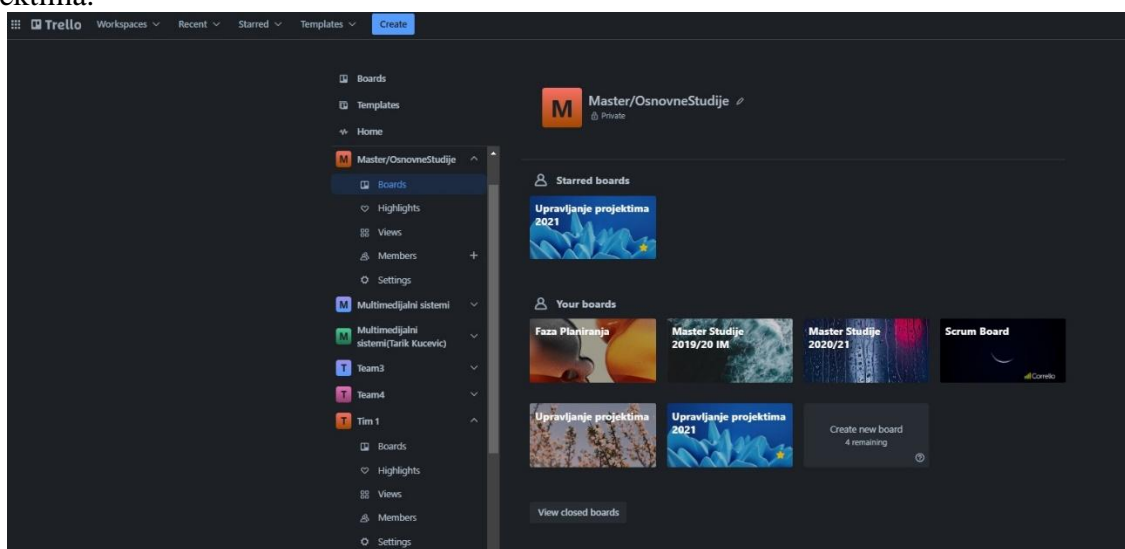
Pored tehničkih karakteristika, Trello je uspeo da stvori snažnu zajednicu korisnika širom sveta. Mnogi timovi i organizacije su prihvatili ovaj alat zbog njegove jednostavnosti i sposobnosti da poveća produktivnost. Kroz forume, blogove i online zajednice, korisnici dele svoje savete, trikove i primere najboljih praksi, što dodatno doprinosi razvoju i popularizaciji alata. Zajednica je postala ključni deo ekosistema Trello-a, jer kroz povratne informacije i diskusije sa developerima, korisnici direktno utiču na dalji razvoj alata. Ovaj model saradnje omogućava stalno unapređenje funkcionalnosti i prilagođavanje novim trendovima u upravljanju projektima.

Trello se pokazao kao izuzetno korisno rešenje u raznim oblastima. U poslovnom svetu, koristi se za planiranje i praćenje timskih zadataka, organizaciju sastanaka, raspodelu resursa i vođenje projekata. Njegova sposobnost da omogući transparentnu komunikaciju i brzu adaptaciju na promene čini ga idealnim alatom za agilne timove koji se suočavaju sa dinamičnim poslovnim okruženjem. Istovremeno, u obrazovanju, Trello se koristi kao pomoćno sredstvo za organizaciju školskih projekata, raspoređivanje zadataka i praćenje napretka učenika. Ljubitelji lične produktivnosti takođe ga koriste za planiranje dnevnih aktivnosti, upravljanje vremenom i postavljanje ličnih ciljeva. Ova široka primena potvrđuje univerzalnost i prilagodljivost alata, koji uspeva da zadovolji potrebe širokog spektra korisnika.

Uprkos svim prednostima, Trello nije bez izazova. Neki korisnici ukazuju na ograničenja u pogledu upravljanja velikim projektima ili složenim zadacima, gde je potrebna dodatna integracija sa specijalizovanim alatima. Ponekad se može desiti da vizuelni pristup, premda vrlo intuitivan, ne obezbedi sve detalje potrebne za kompleksnu analizu podataka. Međutim, ove kritike su uglavnom nadoknađene konstantnim unapređenjima i dodavanjem novih funkcionalnosti putem Power-Ups, kao i aktivnom komunikacijom sa korisničkom zajednicom. Unapređenja se stalno sprovode tako da se obezbedi što veća skalabilnost i prilagodljivost, a razvojni tim redovno objavljuje novosti koje odgovaraju na potrebe tržišta.

Zaključno, Trello predstavlja snažan primer kako jednostavnost i intuitivan dizajn mogu transformisati način na koji organizujemo radne zadatke i projekte. Njegov vizuelni pristup, baziran na tablama, listama i karticama, omogućava korisnicima da brzo steknu pregled nad svim aspektima svog rada, čime se olakšava koordinacija i komunikacija unutar timova. Konstantna evolucija alata, kroz dodavanje novih funkcionalnosti i unapređenja putem Power-Ups, osigurava da Trello ostane relevantan i prilagodljiv potrebama modernog poslovanja. Sa snažnom zajednicom korisnika i aktivnom podrškom developera, Trello je postao neizostavan alat u arsenalu svakog tima koji teži ka većoj produktivnosti, transparentnosti i fleksibilnosti u organizaciji rada.

Ovaj alat nije samo sredstvo za upravljanje zadacima, već i platforma koja inspiriše kreativnost, saradnju i kontinuirano unapređenje. Kroz svoje korisničko iskustvo, Trello nas je naučio da je jasnoća i jednostavnost često ključ uspeha, kako u poslovnom svetu, tako i u ličnom razvoju. Njegova sposobnost da transformiše kompleksne projekte u lako razumljive vizuelne tokove rada omogućava korisnicima da se fokusiraju na suštinske aspekte svog posla, smanjujući stres i povećavajući efikasnost. Upravo zbog toga, Trello je stekao status jednog od najomiljenijih alata za organizaciju, kako među profesionalcima, tako i među entuzijastima lične produktivnosti, ostavljajući trajan uticaj na način na koji pristupamo svakodnevnim zadacima i dugoročnim projektima.



Slika 56 Trello radno okruženje

Naša prva odluka je bila da koristimo MATLAB kao softver temeljac za razvoj virtualne laboratorije zbog njegovih moćnih mogućnosti modeliranja, simulacije i analitike. MATLAB obezbeđuje robusne alate za rešavanje diferencijalnih jednačina, koje su fundamentalne u DSP-u, a njegova kutija alata za diferencijalne jednačine nudi širok spektar numeričkih metoda za

rešavanje i običnih i parcijalnih diferencijalnih jednačina. Ove karakteristike su posebno vredne u oblastima kao što su fizika i inženjering, gde matematičko modeliranje i simulacija igraju ključnu ulogu.

Pored svojih računarskih mogućnosti, MATLAB je veoma efikasan za obradu slika, što je ključni fokus našeg DSP inženjerskog zadatka. Paket sa alatkama za obradu slika pruža napredne funkcije za uvoz, prethodnu obradu i analizu digitalnih slika, čineći osnovu za različite aplikacije. Štaviše, MATLAB podržava širok spektar formata slikovnih datoteka, obezbeđujući besprekornu integraciju sa različitim izvorima podataka i olakšavajući efikasne tokove istraživanja.

Još jedan važan aspekt MATLAB-a je njegov alat za duboko učenje, koji nudi intuitivan okvir za projektovanje i implementaciju dubokih neuronskih mreža. Pored pružanja raznovrsnih prilagodljivih algoritama, MATLAB takođe uključuje unapred obučene modele za različite aplikacije. Najrelevantnija arhitektura za naš inženjerskog zadatka je konvoluciona neuronska mreža (CNN), zajedno sa MATLAB-ovom opsežnom podrškom za različite mrežne strukture.

### **Implementacija Scrum-a.**

Scrum je jedna od najčešće korišćenih agilnih praksi. To je inkrementalni i iterativni pristup razvijen da upravlja složenošću u okruženju stalno promenljivih zahteva za proizvodom. Pored toga, implementacija Scruma je izuzetno efikasna u razvoju novih i inovativnih proizvoda, a koraci definisani u Scrumu se usklađuju sa konstruktivističkim okvirom.

U našem okviru, Scrum će biti korišćen u fazi implementacije razvoja sistema. Prvi korak je definisanje uloga u Scrum razvojnom okruženju: Scrum Master održava Scrum procese; Vlasnik proizvoda definiše potrebe proizvoda; Scrum tim je zadužen za implementaciju i razvoj proizvoda

Vlasnik proizvoda, na početku sprinta, predstavlja grupu funkcionalnosti u vidu backlog-a proizvoda (prioritetna lista osnovnih potreba korisnika). Tim radi na tekućem sprintu. Nijedna osoba nije ovlašćena da menja zadatke u backlog-u dok sprint traje. Nakon završetka sprinta, tim će predstaviti postignute rezultate.

Posao se primarno obavlja u toku sprinta, vremenski ograničenog iterativnog procesa, koji predstavlja period dodeljen timu da završi izabrane zadatke. Dužina sprinta je jedna nedelja, i ta dužina ostaje konstantna do kraja razvoja virtuelne laboratorije. Članovi tima biraju zadatke iz backlog-a proizvoda, verujući da ti zadaci treba da budu završeni unutar jednog sprinta, i generišu sprint backlog sa atributima i funkcijama uključenim u sesiju planiranja sprinta.

Ovo je okvir strukture Scruma implementirane tokom faze implementacije:

Vlasnik proizvoda generiše product backlog (prioritetna lista projektnih zadataka).

Scrum tim sprovodi sesiju planiranja sprinta, gde se neophodni zadaci za završetak stavki sa liste želja dele na male i upravljive blokove.

Scrum tim kreira i ažurira sprint backlog.

Trajanje svakog sprinta odlučuje tim (najčešće trajanje je jedna nedelja).

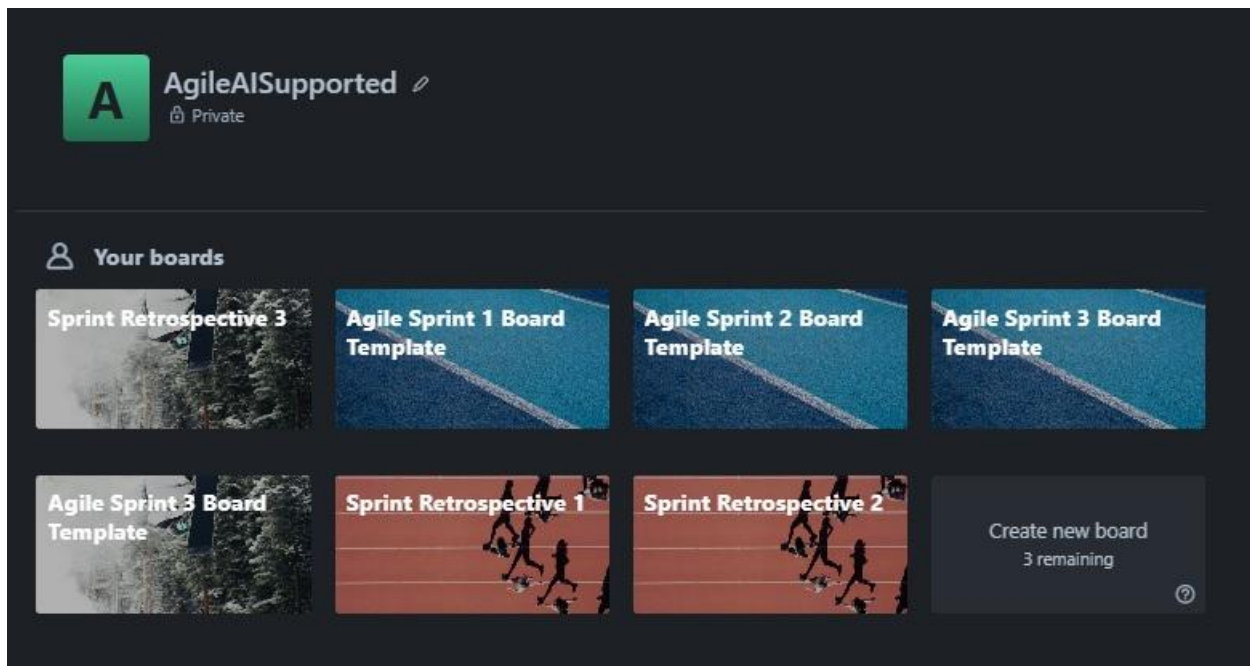
Svakodnevno se održava Scrum sastanak (takođe poznat kao dnevni standup) među članovima tima, gde dele svoja mišljenja i prate napredak proizvoda.

Scrum Master pruža smernice Scrum timu, ohrabruje ih i motiviše da ostanu fokusirani na svoj rad.

Na kraju svakog sprinta, zainteresovane strane i Vlasnik proizvoda održavaju sastanak za pregled.

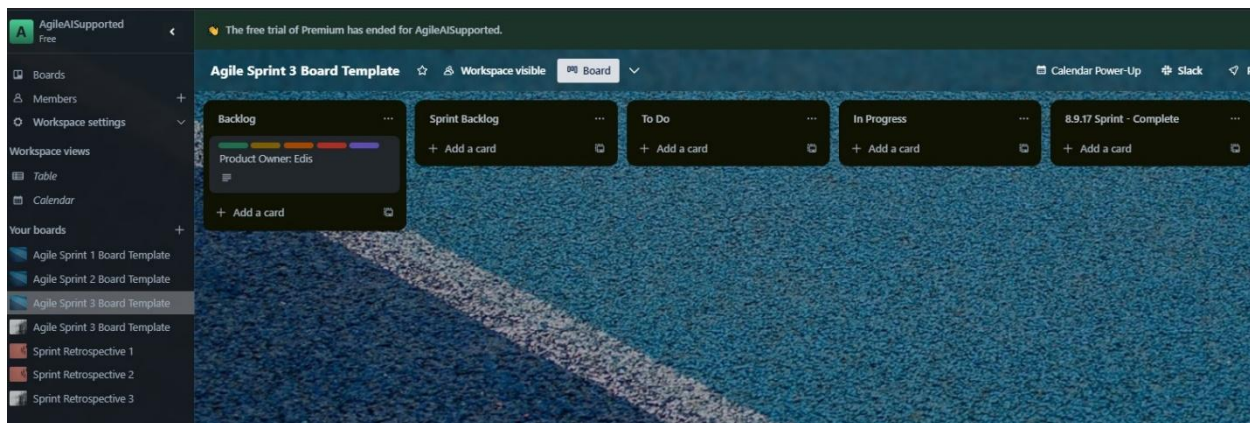
Ovaj ciklički pristup sledi Scrum tim u razvoju softverskog proizvoda. Vlasnik proizvoda, Scrum tim i Scrum Master rade zajednički, pri čemu svaka strana igra ključnu ulogu u postizanju uspeha proizvoda. Rezultati sa sastanka za pregled biće dokumentovani, s naglaskom na pozitivna i negativna iskustva. Ovi podaci će biti analizirani radi poboljšanja procesa učenja i validacije hipoteza.

U treću smo definisali tri elementa sprinta i tri table za retrospektivu to jest za povratne informacije koje su bitne za analizu uspešnosti implementiranja i razvoja proizvoda.



Slika 57 Trello sprint i ploče za povratne informacije

U skladu sa zahtevima Sprinta, Product owner je definisao sledeće kartice u Trello u svaki sprint Backlog (gde će se nalaziti Product backlog), Sprint Backlog (podeljen product backlog na sastavne delove za implementaciju), To Do (Zadaci koji se planiraju za rad), In progress (Yadaci koji su u toku izrade), Completed (Završeni zadaci). Zaostatak proizvoda je pažljivo dizajniran da bude u skladu sa ciljevima učenja inženjerskog zadatka.



Slika 58 Osnovna forma za praćenje Sprinta u Trello

Cilj prvog sprinta bio je da se razvije deo koji će simulirati osnovne koncepte konceptima digitalne obrade signala (DSP) kao i mašinskog učenja koji se primenjuju na analizu slike i zvuka.

Zato su zahtevi za razvoj prvog dela virtuelne laboratorije, bili predstavljanje vremenskog i frekventnog domena, izdvajanje karakteristika i osnove mašinskog učenja. Praktična implementacija laboratorije je postignuta korišćenjem MATLAB-ovih alata za prethodnu obradu signala, ekstrakciju karakteristika i izgradnju osnovnih modela mašinskog učenja.

Ovo je ukupna lista zahteva/funkcionalnosti (user stories) koji se planiraju za projekat. Primeri user story-ja pisani su u formi „Kao [korisnik], želim [funkcija], tako da [cilj].” Uz svaki user story može da se doda prioritet (P1 – najviši, P2 – srednji, P3 – niži) i inicijalna procena napora (story points) ili bilo koja druga metodologija koja se koristi za estimaciju. Product backlog za prvu aktivnost je dat u tabeli.

Tabela 2 Product backlog za prvu aktivnost

Sprint 1			
User story	Kriterijumi prihvatanja	Prioritet	Storz point
Kao korisnik, želim da vidim glavni (početni) ekran aplikacije sa glavnim dugmetom 'SignalProcessingBasics' tako da mogu lako da se krećem kroz aplikaciju.	Postoji glavni prozor aplikacije sa jasno vidljivim naslovom i dizajnom. Glavni ekran sadrži dugme 'SignalProcessingBasics'. Klikom na dugme otvara se novi prozor za obradu slike.	P1	2
Kao korisnik, želim da učitam proizvoljnu sliku sa svog računara i prikažem je u aplikaciji kako bih mogao da obradim sliku.	Postoji opcija da pregledate (ili prevučete i otpustite) da biste učitali sliku. Nakon učitavanja, slika je vidljiva u prozoru za pregled. Aplikacija podržava standardne formate (npr. PNG, JPG, itd.).	P1	3
Kao korisnik, želim da konvertujem učitano sliku u različite formate boja (RGB, HSV, CIE Lab, ICbCr) kako bih mogao da vidim razlike u predstavljanju i pripremim sliku za dalju obradu.	Nakon učitavanja, korisnik može izabrati željeni format. Aplikacija prikazuje konvertovanu sliku u novom prozoru ili pored originala u istom prozoru.	P2	5
Kao korisnik, želim da normalizujem sliku (pretvorim je u nijanse sive) i vidim je	Postoji opcija za normalizaciju slike. I originalna i normalizovana slika se prikazuju jedna pored druge.	P2	3

pored originala kako bih mogao da uporedim razlike.			
Kao korisnik, želim da poboljšam kontrast slike pomoću metode Histogram Ekualization kako bih mogao bolje da istaknem detalje.	Korisnik bira akciju Histogram Ekualization. Dobijena slika se prikazuje pored originala radi poređenja.	P2	5
Kao korisnik, želim da dodam određeni nivo buke soli i bibera, a zatim da je uklonim koristeći različite tehnike filtriranja kako bih mogao da razumem efikasnost različitih filtera.	Korisnik bira intenzitet šuma koji će se dodati slici. Svaki filter se može primeniti pojedinačno. Slika sa šumom i filtrirana slika se prikazuju radi poređenja.	P1	13
Kao korisnik, želim da rotiram sliku za proizvoljan ugao (stepeni) kako bih mogao da ispravim orijentaciju.	Postoji interfejs za podešavanje ugla rotacije. Prikazuju se originalne i rotirane slike.	P3	3
Kao korisnik, želim da promenim veličinu (skal) slike i opciono izaberem metod ponovnog uzorkovanja kako bih mogao da je prilagodim različitim potrebama.	Korisnik određuje željene dimenzije ili faktor skaliranja. Prikazuju se originalne i obrađene slike. Korisnik može izabrati metod interpolacije (npr. najbliži, bilinearni, bikubni).	P3	5
Kao korisnik, želim da pokušam da otklonim zamućenje slike pomoću Vinerovog filtera kako bih mogao da uklonim efekte zamućenja.	Postoji opcija da primenite Viener filter. Prikazuju se originalne i obrađene slike.	P2	8
Kao korisnik, želim da primenim segmentaciju slike koristeći metodu aktivnih kontura (zmije) tako da mogu da izolujem regione od interesa.	Postoji opcija za pokretanje Active Contours. Korisnik može da definiše početnu oblast (ili parametre) za konturu. Rezultat segmentacije se prikazuje pored originala.	P3	8
Kao korisnik, želim da primenim Harris Corner Detector za detekciju karakteristika (tačaka) na slici tako da mogu da identifikujem ključne tačke.	Korisnik bira parametre Harrisovog algoritma (npr. prag). Dobijene tačke (uglovi) se prikazuju na slici.	P3	5
Kao korisnik, želim da izvršim fuziju slika koristeći Vavelet transformaciju tako da mogu	Aplikacija omogućava učitavanje dve slike. Postoji opcija za Vavelet fuziju. Dobijena spojena slika se prikazuje pored originala.	P3	8

spojiti informacije iz različitih izvora slika.			
---	--	--	--

Drugi sprint se fokusira na razvoj tehnike obrade slike i klasifikaciju slika korišćenjem konvolucionih neuronskih mreža (CNN). Na osnovu prethodnih prethodnih primera za obradu slika, primenićemo CNN arhitekture za mašinsko učenje koristeći standardne modele za trening kao što je AlekNet. Product backlog za drugi sprint je dat u sledećoj tabeli.

Tabela 3 Product backlog za drugi sprint

ID	Karakteristika	Korisnička priča	Kriterijumi prihvatanja	Prioritet	Procena (SP)
1	Početni ekran	Kao korisnik, želim da vidim dugme „Duboko učenje za analizu slike“ na glavnom ekranu kako bih mogao da pristupim svim funkcijama vezanim za CNN.	Glavni prozor aplikacije postoji. Dugme „Duboko učenje za analizu slike“ je vidljivo i funkcionalno. Klikom na dugme otvara se novi ekran sa opcijama dubokog učenja.	P1	2
2	Uvod u CNN	Kao korisnik, želim da naučim osnove CNN-a, uključujući njihovu arhitekturu i funkcionalnost, da razumem duboke neuronske mreže za obradu slika.	Opisni tekst (ili video/slajdovi) o osnovama CNN-a. Ilustracije CNN slojeva (konvolucija, objedinjavanje, potpuno povezano).	P2	3
3	CNN Arhitektura	Kao korisnik, želim da istražim različite CNN arhitekture (LeNet, AlekNet, VGG, ResNet) da bih razumeo koji model najbolje odgovara mom projektu.	Prikaz ključnih informacija o svakoj arhitekturi (broj slojeva, karakteristike). Detaljni pregled strukture (LeNet-5 ima 7 slojeva, AlekNet 8, VGG-16, VGG-19).	P2	5
4	Skupovi podataka (učitavanje &	Kao korisnik, želim da učitam različite skupove podataka (MNIST, Fashion-MNIST, CIFAR,	Lista podržanih skupova podataka sa opcijama preuzimanja/otpremanja. Funkcije za promenu veličine i normalizaciju. Prikaz osnovne	P1	8

	prethodna obrada)	ImageNet, itd.) i da ih unapred obradim (promena veličine, normalizacija) za obuku CNN modela.	statistike skupa podataka (broj slika, dimenzije, broj klasa).		
5	Obuka CNN-a	Kao korisnik, želim da konfiguriram parametre obuke (optimizator, brzinu učenja, epohe, itd.) i obučim CNN model na izabranom skupu podataka.	UI za određivanje parametara obuke. Dugme za početak treninga sa trakom/grafikonima napretka. Prikaz gubitka, tačnosti i drugih metrika tokom obuke.	P1	13
6	Klasifikacija slika	Kao korisnik, želim da klasifikujem slike koristeći obučeni CNN model da bih identifikovao njihovu kategoriju.	Mogućnost izbora unapred obučene CNN arhitekture ili učitavanja prilagođenog modela. Prikažite rezultate klasifikacije (npr. najbolji 1, najbolji 5 razredi). Vizuelna potvrda predviđanja (npr. "cat", "dog", "shirt").	P2	5
7	Detekcija objekata	Kao korisnik, želim da detektujem objekte na slikama i lokalizujem ih koristeći granične okvire sa YOLO ili brži R-CNN.	Mogućnost izbora algoritma (YOLO, brži R-CNN). Prikaz graničnog okvira na slikama. Verovatnoća (ocena pouzdanosti) za svaki otkriveni objekat.	P2	8
8	Semantička segmentacija	Kao korisnik, želim da primenim semantičku segmentaciju na slike da dodelim klasu svakom pikselu (npr. pozadina, put, automobil, pešak).	Podrška za U-Net, SegNet ili sličnu arhitekturu. Prikaz rezultata segmentacije (maskirane slike, segmenti označeni bojama). Sposobnost upoređivanja segmentirane slike sa originalom.	P3	8
9	Evaluacija metrike	Kao korisnik, želim da vidim i razumem ključne metrike evaluacije kao što su tačnost, preciznost, opoziv, F1 rezultat,	Izračunajte i prikažite ove metrike nakon obuke ili zaključivanja. Grafičko predstavljanje (matrica konfuzije, grafovi preciznog opoziva, itd.). Kratki opisi/opisi za	P2	5

		IoU i matrica konfuzije.	svaku metriku radi boljeg razumevanja.		
10	Korisnik CNN Kreator (opciono)	Kao napredni korisnik, želim da prilagodim svoju CNN arhitekturu (broj i vrste slojeva) za eksperimentisanje.	UI za dodavanje konvolucionih i objedinjenih slojeva, definisanje veličina filtera, itd. Mogućnost čuvanja prilagođenog modela kao JSON, MAT datoteke, itd. Integracija sa inženjerskim zadatkom CNNs modulom za korišćenje prilagođene arhitekture.	P3	8

Poslednja u poslednjoj nedelji razvijamo naprednim tehnike veštačke inteligencije, posebno multimodalne sistemima veštačke inteligencije koji integrišu podatke o slici i zvuku. Učesnici će istraživati generativne tehnike veštačke inteligencije.

Tabela 4 Generativne tehnike veštačke inteligencije

1	Osnovni okvir neuronske mreže	Implementirati osnovne komponente mreže (slojevi, aktivacione funkcije, itd.) za podršku prilagođenim simulacionim modelima.	P1	5
2	Simulacija GAN-a	Razviti simulaciju za generativnu suparnišu mrežu, uključujući kodiranje generatora, diskriminatora i petlje za treniranje.	P1	8
3	Simulacija VAE-a	Implementirati simulaciju varijacionog autoenkodera sa enkoderom, dekoderom i uzorkovanjem latentnog prostora za generisanje slika.	P2	5
4	Simulacija difuzionog modela	Implementirati simulacione rutine za generativne modele zasnovane na difuziji, obuhvatajući i proces unapred (dodavanje šuma) i unazad (uklanjanje šuma).	P2	8
5	Modul za vizuelizaciju	Kreirati korisnički interfejs ili MATLAB skripte za prikaz generisanih slika, napretka u treniranju.	P1	3

Ovime smo završili prvu fazu planiranja i možemo da pređemo na ključnu fazu, implementacije i sakupljanja rezultata istraživanja.

## 8. REZULTATI ISTRAŽIVANJA

Kako smo se dosada držali Akcionpg istraživanja čije faze su do sada bile u skladu sa osnovnim koracima agilnog razvoja siftvera. Da bi prikupili podatke za poslednju fazu alanilze istoh u sledećoj fazi koja je kod razvoja simulacije VoIP sistema, faza primene, dok kod razvoja softvera za obradu signala to je faza sprovođenja sprintova. U ovom poglavlju opisaćemo korake preduzete u ovoj fazi i tip podataka i rezultata koji su prikupljeni implementacijom istih.

### 8.1 Analiza rada simuliranog VoIP sistema

Tim\_B, zadužen za Virtual VoIP, instalirao je serverku mašinu, simulator mreže i softver za virtualne računare-klijente s osam virtualnih mašina. JVirtualni mašine bile su povezani preko virtualne mreže, a instaliran je PBX progrsam. Na klijentskim mašinama instalirani su softfoni 3CXPhone.

Tim\_C, zadužen za Real VoIP, instalirao je računasku mrežu od osam računara. Na serveru sa PBX-om instaliran je softver, a na klijentskim računarima također su instalirani softfoni 3CXPhone.

Oba tima koristila su istu topologiju i postavke za rad VoIP sistema. Za pokretanje ova dva sistema i pripremu detaljne dokumentacije o koracima instalacije, ispunjeni su IQ zahtjevi GLP/GMP. Na temelju ovoga, Tim\_D je dovršio konfiguraciju PBX-a, konfiguraciju korisničkih uređaja, vezu sa inetnetom, konferencijske mogućnosti, nadzor sistema i sigurnost.

IQ testovi dokazali su da je oprema ispravno instalirana i funkcioniра. Sada je potrebno testirati jesu li instalirani hardver i softver prikladni za predviđenu upotrebu. U oba slučaja, u VirtualVoIP i RealVoIP sustavima, instaliran je program za praćenje mrežnog prometa, te je napravljen plan testiranja I poređenja testiranih poziva.

Ista konfiguracija korištena je u VirtualVoIP i RealVoIP sustavima i programima za praćenje mrežnog saobraćaja.

prvi korak je bio uspostva veze između dva telefona sa različitim virtuelnih mašina.

Ovde će biti slikovito prikazano međusobno pozivanje dva telefona (broja) koji su instalirani na dve različite virtuelne mašine, čime će se na najbolji način demonstrirati funkcionalnost celokupnog sistema.

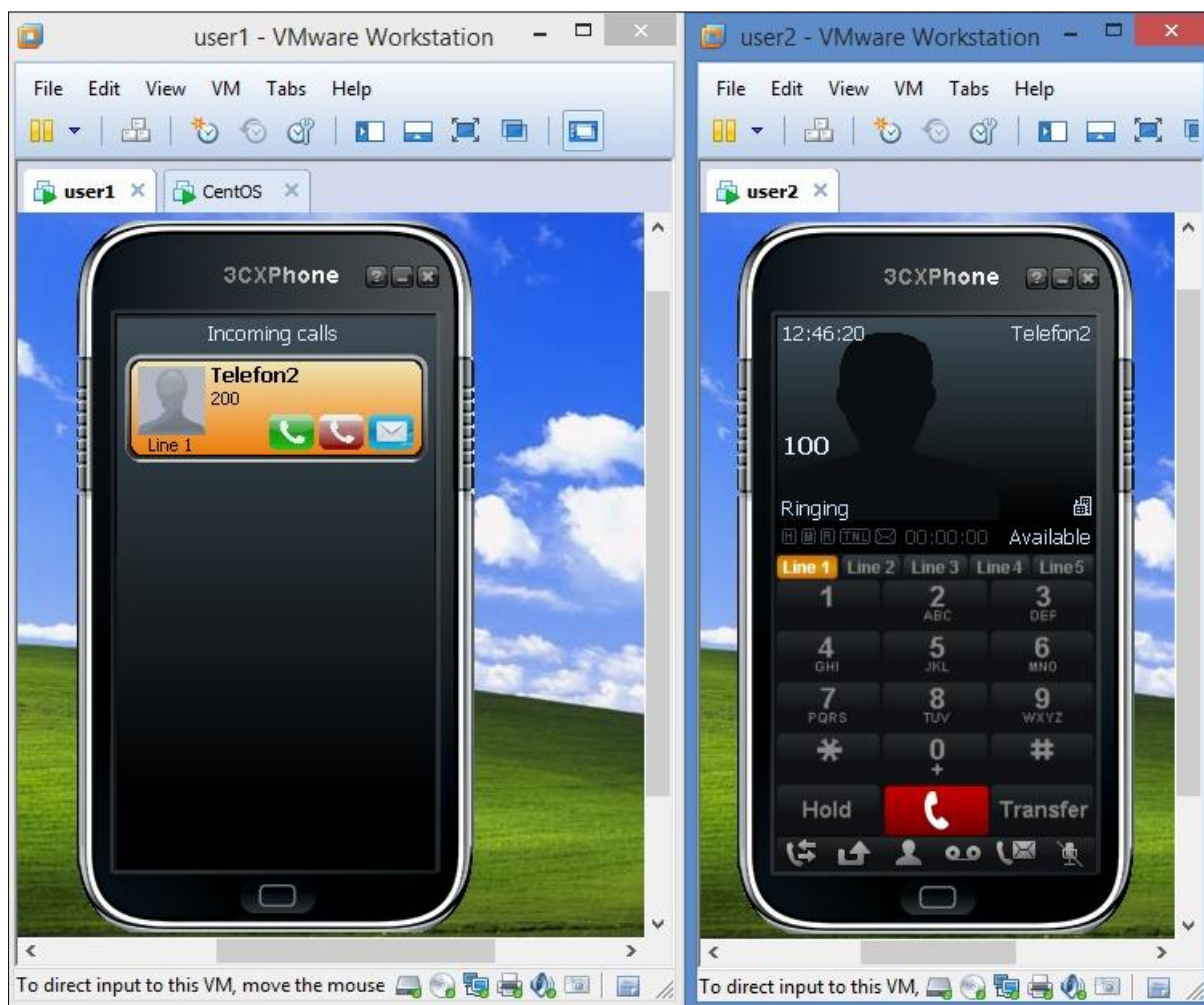
Podsećanja radi, instalirane je deset virtuelnih mašina, od kojih jedna predstavlja Asterisk server, a druge dve služe da na njima instaliramo po jedan (po potrebi može i više) softverskih telefona na kojima je kreiran po jedan nalog na način koji je ranije predstavljen. Na obe virtuelne mašine su instalirani 3CX Phone softverski telefoni.



Slika 59 Uspostava veze od user1 ka user2 virtuelnoj mašini

Na slici je dat primer uspostave veze od virtuelne mašine user1 ka user2 virtuelnoj mašini tako što unosimo pozivni broj (200) koji je dodeljen virtuelnoj mašini user2. Na ekranu telefona kod virtuelne mašine user2 vidimo da je pristigao poziv sa pozivnog broja 100 (pozivni broj koji je dodeljen virtuelnoj mašini user1), i na nama je da izaberemo opciju odbijanje ili prihvatanje poziva.

Isti postupak, samo sada u suprotnom smeru, tj. poziv se uspostavlja od virtuelne mašine user2 ka virtuelnoj mašini user1 prikazan je na slici 60.



Slika 60 Uspostava veze od user2 ka user1 virtuelnoj mašini

Programski alat Wireshark koristili smo za analizu mrežnih paketa. Radi se o alatu koji hvata podatke koji u paketima putuju mrežom i prikazuje ih na najdetaljniji mogući način. U prošlosti, alati slični Wireshark-u su bili skupi i najčešće komercijalni. Dolaskom alata Wireshark na tržište situacija se promenila. Wireshark je danas verovatno najbolji besplatni i open source alat dostupan na tržištu. Neki od primera korišćenja ovog alata su:

- Otklanjanje problema na mreži,
- Analiza sigurnosnih nedostataka,
- Razvoj i implementacija novih protokola i
- Učenje o mrežnim protokolima.

Wireshark je tzv. „cross-platform“ mrežni alat, predstavlja da može raditi na različitim platformama. Osim što radi na operativnom sistemu Microsoft Windows, podržan je i na različitim Unix operativnim sistemima među kojima su Linux, Mac OS X, BSD i Solaris. Takođe, postoji i verzija bez grafičkog interfejsa (engl. Graphical User Interface) nazvana TShark. Wireshark i Tshark su besplatni alati.

Na slici 61 je prikazana analiza protoka saobraćaja prilikom uspostave veze između dve virtuelne mašine, tj. kada VoIP telefon instaliran na virtuelnu mašinu user1 poziva VoIP telefon instaliran na virtuelnu mašinu user2.

Na slici možemo videti korake prilikom uspostave veze i to:

User1 (čija je IP adresa 192.168.2.2) šalje zahtev INVITE sip:200@192.168.10.2:5060 prema serveru kojom želi da uspostavi sesiju sa telefonom koji ima broj 200;

Server mu odgovara statusnom porukom 401;

User1 šalje ACK zahtev prema serveru i ponavlja INVITE zahtev iz koraka1;

Sada server prosleđuje zahtev INVITE sip:200@192.168.3.2:1035 prema User2 virtuelnoj mašini (čija je IP adresa 192.168.3.2);

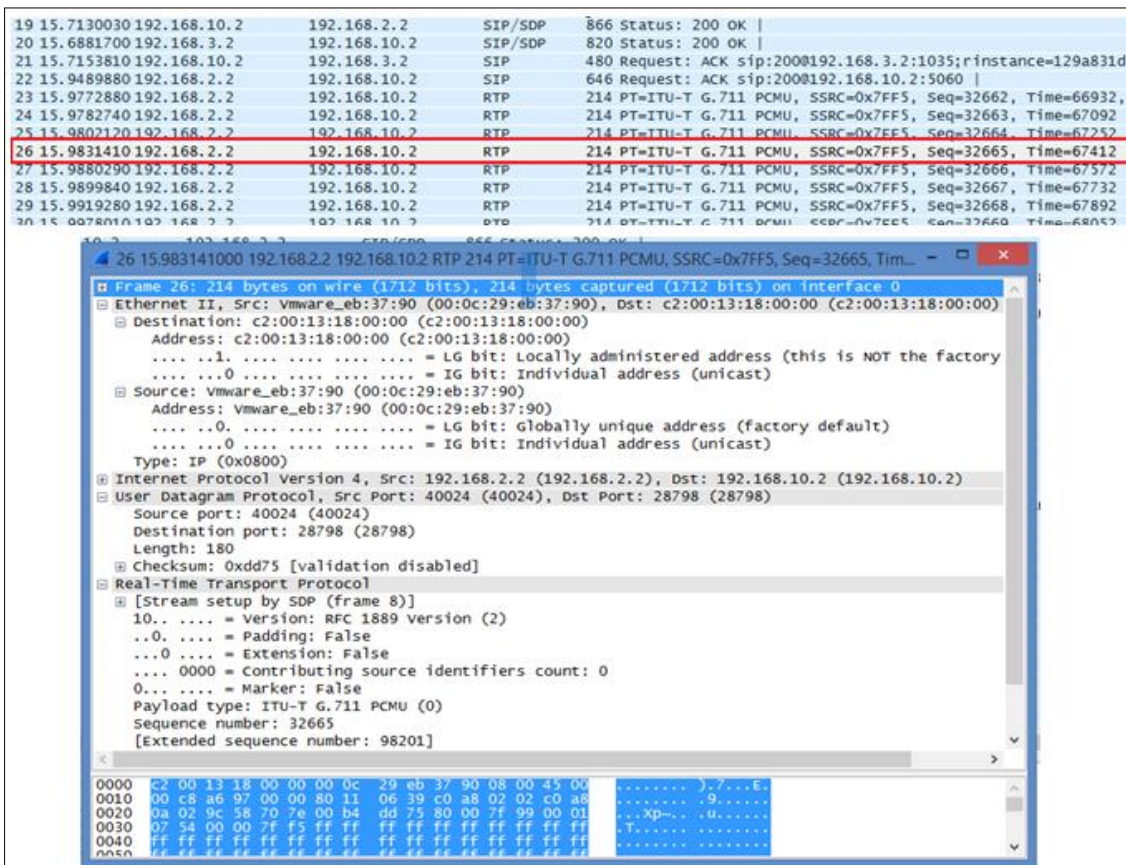
U narednom koraku User2 obaveštava server da je prihvatio poziv i da „zvoni“, server posle obaveštava User1 da je uspostavio vezu sa User2;

Kada User2 prihvati poziv i server obavesti User1 da je User2 prihvatio poziv, statusnom porukom 200 OK, počinje razmena paketa posredstvom RTP protokola.

Time	Source	Destination	Protocol	Length	Info
5	10.9013220	192.168.2.2	SIP/SDP	1023	Request: INVITE sip:200@192.168.10.2:5060
6	10.9265770	192.168.10.2	SIP	600	Status: 401 Unauthorized
7	10.9276780	192.168.2.2	SIP	386	Request: ACK sip:200@192.168.10.2:5060
8	11.0289810	192.168.2.2	SIP/SDP	1185	Request: INVITE sip:200@192.168.10.2:5060
9	11.2298840	192.168.10.2	SIP/SDP	913	Request: INVITE sip:200@192.168.3.2:1035;rinstance=129a831d7c38f357
10	11.3465270	192.168.3.2	SIP	447	Status: 180 Ringing
11	11.3926600	192.168.10.2	SIP/SDP	913	Request: INVITE sip:200@192.168.3.2:1035;rinstance=129a831d7c38f357
12	11.3943660	192.168.3.2	SIP	447	Status: 180 Ringing
13	11.2290750	192.168.2.2	SIP	541	Status: 100 Trying
14	11.3919320	192.168.10.2	SIP	557	Status: 180 Ringing
15	11.5309830	192.168.10.2	SIP/SDP	913	Request: INVITE sip:200@192.168.3.2:1035;rinstance=129a831d7c38f357
16	11.5338940	192.168.3.2	SIP	447	Status: 180 Ringing
17	11.5303350	192.168.10.2	SIP	557	Status: 180 Ringing
18	15.6736400	192.168.2.2	UDP	46	Source port: mtqp Destination port: sip
19	15.7130030	192.168.10.2	SIP/SDP	866	Status: 200 OK
20	15.6881700	192.168.3.2	SIP/SDP	820	Status: 200 OK
21	15.7153810	192.168.10.2	SIP	480	Request: ACK sip:200@192.168.3.2:1035;rinstance=129a831d7c38f357
22	15.9489880	192.168.2.2	SIP	646	Request: ACK sip:200@192.168.10.2:5060
23	15.9772880	192.168.2.2	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x7FF5, Seq=32662, Time=66932, Mark
24	15.9782740	192.168.2.2	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x7FF5, Seq=32663, Time=67092
25	15.9802120	192.168.2.2	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x7FF5, Seq=32664, Time=67252
26	15.9831410	192.168.2.2	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x7FF5, Seq=32665, Time=67412
27	15.9880290	192.168.2.2	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x7FF5, Seq=32666, Time=67572
28	15.9899840	192.168.2.2	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x7FF5, Seq=32667, Time=67732
29	15.9919280	192.168.2.2	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x7FF5, Seq=32668, Time=67892
30	15.9978010	192.168.2.2	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x7FF5, Seq=32669, Time=68052

Slika 61 Analiza protoka saobraćaja prilikom uspostave veze između user1 i user2 virtuelnih mašina

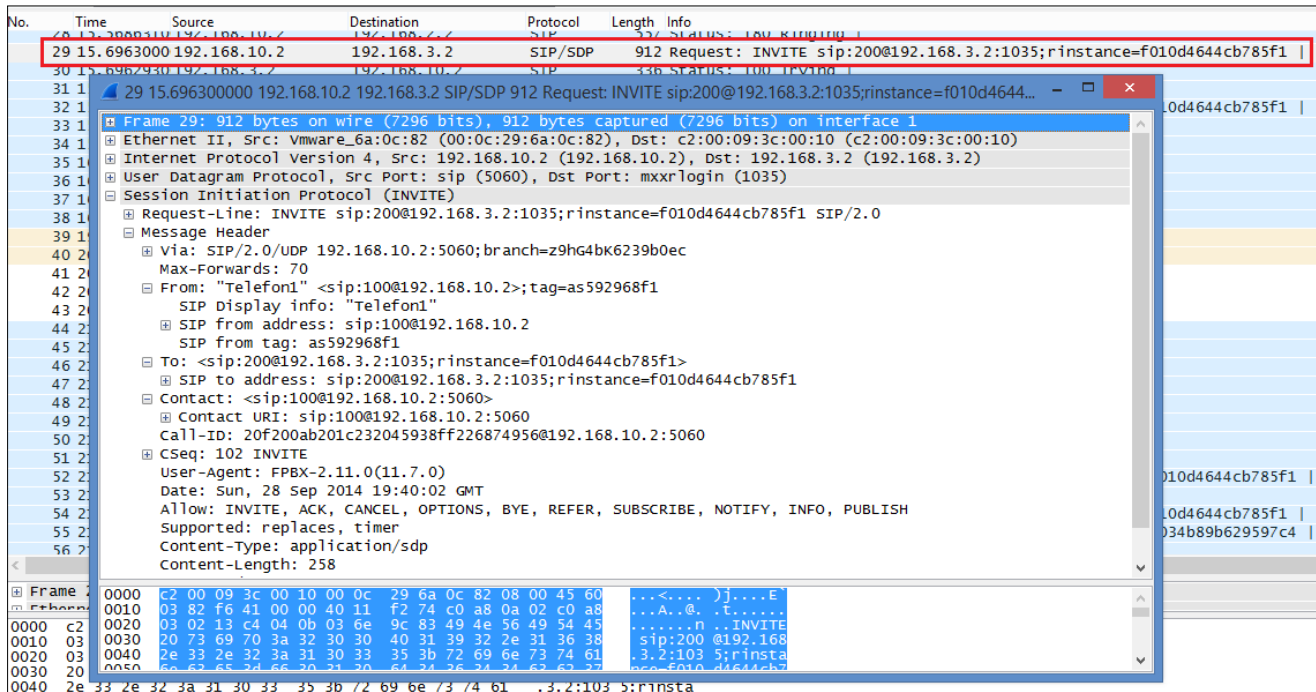
Na slici 62 dat je prikaz jedne RTP poruke. Prikazana su polja koja čine ovu poruku. Između ostalog možemo videti da poruka sadrži Ethernet polje što nam govori da se koristi Ethernet veza između izvora i odredišta (user1 i servera). Naredno polje je Internet Protocol Version 4 koje nam govori da se taj protokol koristi između izvora (192.168.2.2) i destinacije (192.168.10.2). Sledeće polje je User Datagram Protocol što nam govori da ovaj protokol koristi za prenos podataka između izvorišnog porta (Src port) i destinacionog porta (Dst port). Vidimo da ovo polje u sebi sadrži i podatke o izvorišnom i odredišnom portu, kao i o dužini. Poslednje polje nam kazuje da se koristi Real-Time Transport Protocol što nam kazuje da se veza uspostavila i da traje razmena paketa, između izvora i odredišta, u realnom vremenu.



Slika 62 Prikaz jedne RTP poruke

Slično kao i na prethodnoj slici, i na slici 63 je dat primer jedne SIP poruke, u ovom slučaju je to SIP INVITE poruka koju server (192.168.10.2) šalje virtualnoj mašini user2 (192.168.3.2). Struktura polja je slična kao i na slici 36, ono što je uočljivo kao razlika je to što umesto RTP protokola imamo SIP protokol. Na slici se može videti da je u pitanju INVITE poruka i da je upućena na adresu 192.168.3.2.

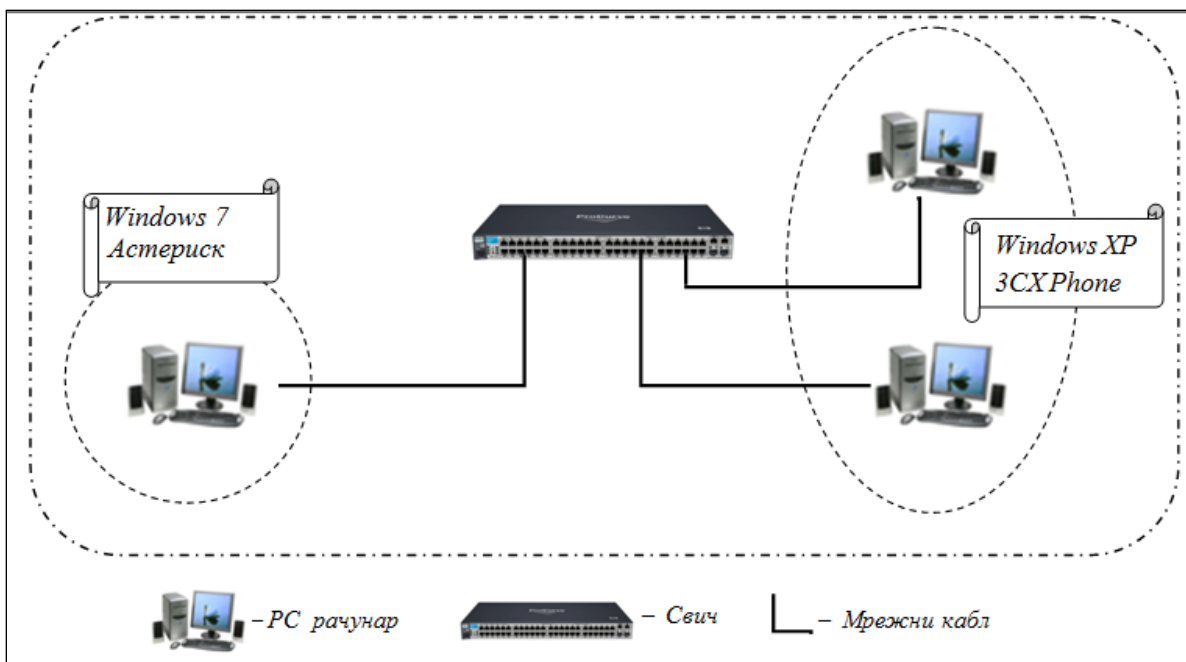
Kao što je i u prethodnim poglavljima opisano, sada to možemo videti i na slici 38, SIP ima zaglavlje i telo poruke. U zaglavlju vidimo da INVITE poruka dolazi sa adrese 192.168.2.2 i da je upućena na adresu 192.168.3.2 i to preko servera čija je adresa 192.168.10.2.



Slika 63 Prikaz jedne SIP poruke

Pošto smo projektovali sistem i uspešno izvršili simulaciju, preostaje nam još da to i fizički implementiramo. Dakle, u ovom odeljku će biti opisan postupak implementacije ovog sistema (projekta).

Šematski prikaz mreže i elementa u istoj dat je na sledećoj slici (slika 64).



Slika 64 Šematski prikaz umrežavanja elemenata sistema

Kao što se vidi na slici 64, umreženo je deset PC računara posredstvom mrežnih kablova na svič. Za razliku od softverske implementacije gde je svaka virtuelna mašina pripadala zasebnoj mreži i ruterom bile povezane, ovde imamo slučaj da svi računari pripadaju istoj mreži i povezani su preko sviča.

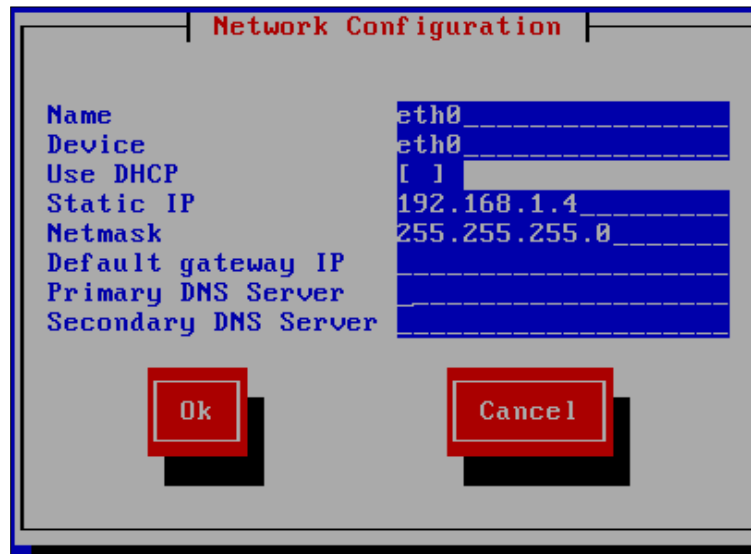
Da bi ceo sistem mogao uspešno da funkcioniše, kao i kod softverske simulacije ovog sistema, neophodno je da svaki računar poseduje neophodan softver. Na slici 39 je označeno da je na jednom računaru instaliran operativni sistem Windows 7, a na preostala dva računara je instaliran operativni sistem Windows XP.

Potreban softver koji je instaliran na računaru koji sadrži Windows 7 operativni sistem je: VMware workstatin, Wireshark i Cent OS. Na preostala dva računara, koja rade na Windows XP, je od dodatnog softvera, instaliran samo 3CX Phone softverski telefon.

Pošto je u prethodnom odeljku opisan postupak instalacije i podešavanja svih navedenih programa, u ovom delu neće biti predstavljen postupak njihove instalacije.

Ono što je potrebno uraditi jeste instalirati VMware workstatin na Windows 7 operativnom sistemu i kreirati novu virtuelnu mašinu, na način koji je predstavljen u prethodnom odeljku. Zatim je potrebno na toj virtuelnoj mašini instalirati Cent OS operativni sistem u sklopu

koga dolazi Asterisk server i neophodno mu je dodeliti IP adresu (192.168.1.4) – slika 65. Potrebno je instalirati i Wireshark koji će služiti za praćenje i analiziranje paketa.



Slika 65 Podešavanje IP adrese na Asterisk serveru

Na preostala dva računara neophodno je instalirati 3CX Phone softverski telefon i potrebno je podesiti lokalne IP adrese, na jednom 192.168.1.1 i na drugom 192.168.1.2. Postupak podešavanja lokalne IP adrese kod Windows XP operativnog sistema je prikazan u prethodnom odeljku.

Sledeći korak je pokretanje PBX telefonske centrale i kreiranje telefonskih brojeva koje je potrebno kasnije dodeliti softverskim telefonima. Prikaz kreiranih telefona dat je na slici 66.

Kreirane naloge, po izboru, je potrebno dodeliti softverskim telefonima koji će biti instalirani na preostala dva računara.



Slici 66 Kreirani telefonski nalozi na Asterisk serveru

Pošto su na serveru kreirani nalozi, potrebno je sada kreirati na softverski telefonima bar po jedan nalog kako bi komunikacija mogla da se obavi.

Na slici 67 je prikazana analiza mreže u Wireshark-u prilikom pozivanja između dva softverska telefona čije su IP adrese 192.168.1.1 i 192.168.1.2. Naravno, komunikacija ide preko servera čija IP adresa je 192.168.1.4.

No.	Time	Source	Destination	Protocol	Length	Info
16	21.0146590	192.168.1.1	192.168.1.4	SIP	707	Request: SUBSCRIBE sip:unknown@192.168.1.4:5060, in-dialog
17	21.0150130	192.168.1.4	192.168.1.1	SIP	608	Status: 401 Unauthorized (0 bindings)
18	21.0152160	192.168.1.4	192.168.1.1	SIP	621	Status: 401 unauthorized
19	21.1161890	192.168.1.1	192.168.1.4	SIP	764	Request: REGISTER sip:192.168.1.4:5060
20	21.1166690	192.168.1.4	192.168.1.1	SIP	631	Request: OPTIONS sip:3000@192.168.1.1:1333;rinstance=a5bc4e5e3cd252a7
21	21.1168510	192.168.1.4	192.168.1.1	SIP	650	Status: 200 OK (1 bindings)
22	21.1169830	192.168.1.1	192.168.1.4	SIP	707	Request: SUBSCRIBE sip:unknown@192.168.1.4:5060, in-dialog
23	21.1172310	192.168.1.4	192.168.1.1	SIP	590	Status: 200 OK
24	21.1174100	192.168.1.4	192.168.1.1	SIP	747	Request: NOTIFY sip:3000@192.168.1.1:1333;rinstance=a5bc4e5e3cd252a7
25	21.2180280	192.168.1.4	192.168.1.1	SIP	747	Request: NOTIFY sip:3000@192.168.1.1:1333;rinstance=a5bc4e5e3cd252a7
26	21.2189360	192.168.1.1	192.168.1.4	SIP	593	Status: 200 OK
27	21.2189370	192.168.1.1	192.168.1.4	SIP	450	Status: 200 OK
28	21.2189380	192.168.1.1	192.168.1.4	SIP	450	Status: 200 OK
49	38.2703040	192.168.1.4	192.168.1.2	SIP/SDF	917	Request: INVITE sip:2000@192.168.1.2:1193;rinstance=d61f3e3a2f492d0d
50	38.2705450	192.168.1.4	192.168.1.5	SIP	558	Status: 180 Ringing
51	38.3708940	192.168.1.4	192.168.1.2	SIP/SDF	917	Request: INVITE sip:2000@192.168.1.2:1193;rinstance=d61f3e3a2f492d0d
52	38.3718110	192.168.1.2	192.168.1.4	SIP	444	Status: 180 Ringing
53	38.3724630	192.168.1.4	192.168.1.5	SIP	558	Status: 180 Ringing
54	38.3730270	192.168.1.2	192.168.1.4	SIP	444	Status: 180 Ringing
55	42.6969090	192.168.1.2	192.168.1.4	SIP/SDF	817	Status: 200 OK
56	42.6973010	192.168.1.4	192.168.1.2	SIP	480	Request: ACK sip:2000@192.168.1.2:1193;rinstance=d61f3e3a2f492d0d
57	42.6977460	192.168.1.4	192.168.1.5	SIP/SDF	869	Status: 200 OK
58	42.7493650	192.168.1.2	192.168.1.4	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x2EA6, Seq=2995, Time=36059, Mark

Slika 67 Analiza mreže prilikom pozivanja sa IP adrese 192.168.1.1 prema IP adresi 192.168.1.2

Na slici se vidi kako, u prvom delu slike, komuniciraju IP adrese 192.168.1.1 i 192.168.1.4, odnosno softverski telefon kome je dodeljen nalog pod nazivom prof (2000) i server kome se on obraća radi uspostave veze. Pošto su telefonski nalog prof i server razmenili poruke da je zahtev ispravan i da je sve u redu (200 OK), server se obraća drugom telefonskom nalogu

pod nazivom Izo (3000) koji se nalazi na IP adresi 192.168.1.2. U donjem delu je prikazana komunikacija između 192.168.1.2 i 192.168.1.4 koja je obostrana, koja se na kraju završava porukom 200 OK što nam govori da je zahtev uspešno prihvaćen, tj. da je komunikacija između dva telefonska broja uspešno uspostavljena.

79	42.9308110	192.168.1.2	192.168.1.4	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x2EA6, Seq=3004, Time=37499
80	42.9309320	192.168.1.4	192.168.1.5	UDP	214	Source port: 18278 Destination port: 40044
81	42.9485400	192.168.1.2	192.168.1.4	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x2EA6, Seq=3005, Time=37659
82	42.9486720	192.168.1.4	192.168.1.5	UDP	214	Source port: 18278 Destination port: 40044
83	42.9718690	192.168.1.2	192.168.1.4	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x2EA6, Seq=3006, Time=37819
84	42.9719910	192.168.1.4	192.168.1.5	UDP	214	Source port: 18278 Destination port: 40044
85	42.9894180	192.168.1.2	192.168.1.4	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x2EA6, Seq=3007, Time=37979
86	42.9895530	192.168.1.4	192.168.1.5	UDP	214	Source port: 18278 Destination port: 40044
87	43.0070380	192.168.1.2	192.168.1.4	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x2EA6, Seq=3008, Time=38139
88	43.0071640	192.168.1.4	192.168.1.5	UDP	214	Source port: 18278 Destination port: 40044
89	43.0304570	192.168.1.2	192.168.1.4	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x2EA6, Seq=3009, Time=38299
90	43.0305880	192.168.1.4	192.168.1.5	UDP	214	Source port: 18278 Destination port: 40044
91	43.0480560	192.168.1.2	192.168.1.4	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x2EA6, Seq=3010, Time=38459
92	43.0481990	192.168.1.4	192.168.1.5	UDP	214	Source port: 18278 Destination port: 40044
93	43.0714580	192.168.1.2	192.168.1.4	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x2EA6, Seq=3011, Time=38619
94	43.0715870	192.168.1.4	192.168.1.5	UDP	214	Source port: 18278 Destination port: 40044
95	43.0891970	192.168.1.2	192.168.1.4	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x2EA6, Seq=3012, Time=38779
96	43.0893380	192.168.1.4	192.168.1.5	UDP	214	Source port: 18278 Destination port: 40044
97	43.1124830	192.168.1.2	192.168.1.4	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x2EA6, Seq=3013, Time=38939
98	43.1126140	192.168.1.4	192.168.1.5	UDP	214	Source port: 18278 Destination port: 40044
99	43.1300730	192.168.1.2	192.168.1.4	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x2EA6, Seq=3014, Time=39099
100	43.1301980	192.168.1.4	192.168.1.5	UDP	214	Source port: 18278 Destination port: 40044
101	43.1476700	192.168.1.2	192.168.1.4	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x2EA6, Seq=3015, Time=39259

Slika 68 Prikaz razmene paketa tokom komunikacije između telefonskog naloga Izo i servera

Na slici 68 je dat prikaz razmene paketa tokom komunikacije između telefonskog naloga Izo i servera. Sličan način komunikacije je opisan u odeljku 3 koji se tiče softverske implementacije, tako da ovaj nećemo detaljno analizirati.

Slika 69 predstavlja prekid ranije uspostavljene komunikacije. Ovim je zaokružen jedan proces komunikacije između tročlanog sistema koga sačinjavaju dva softverska telefona i Asterisk server. Iz priloženih slika se vidi da mreža funkcioniše i da je softverska podrška pravilno instalirana i podešena, što nam potvrđuje da je implementacija sistema uspešno obavljena.

691	49.0073060	192.168.1.4	192.168.1.5	UDP	214	Source port: 18278 Destination port: 40044
692	49.0090860	192.168.1.5	192.168.1.4	SIP	646	Request: BYE sip:2000@192.168.1.4:5060
693	49.0095410	192.168.1.4	192.168.1.5	SIP	512	Status: 200 OK
694	49.0102740	192.168.1.4	192.168.1.2	SIP	515	Request: BYE sip:2000@192.168.1.2:1193;rinstance=d61f3e3a2f492d0d
695	49.0315510	192.168.1.2	192.168.1.4	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x2EA6, Seq=3309, Time=86299
696	49.1105520	192.168.1.4	192.168.1.2	SIP	515	Request: BYE sip:2000@192.168.1.2:1193;rinstance=d61f3e3a2f492d0d
697	49.1115710	192.168.1.2	192.168.1.4	SIP	436	Status: 200 OK
698	49.1115720	192.168.1.2	192.168.1.4	SIP	436	Status: 200 OK
699	52.4554200	Vmware_d8:4c:d3	FujitsuT_d3:96:5e	ARP	42	who has 192.168.1.5? Tell 192.168.1.4
700	52.4554390	FujitsuT_d3:96:5e	Vmware_d8:4c:d3	ARP	42	192.168.1.5 is at 00:19:99:d3:96:5e
701	53.6279860	FujitsuT_d3:96:5e	Vmware_d8:4c:d3	ARP	42	who has 192.168.1.4? Tell 192.168.1.5
702	53.6281150	Vmware_d8:4c:d3	FujitsuT_d3:96:5e	ARP	42	192.168.1.4 is at 00:0c:29:d8:4c:d3
703	56.3142100	192.168.1.5	192.168.1.4	SIP	766	Request: REGISTER sip:192.168.1.4:5060
704	56.3142480	192.168.1.5	192.168.1.4	SIP	700	Request: SUBSCRIBE sip:unknown@192.168.1.4:5060 in dialog

Slika 69 Prekid ranije uspostavljene komunikacije

Isti postupak je ponovljen i urelanoj mreži i snimljen je istovetan saobraćaj što nam govori da je naš virtuelni sistem u potpunosti ispunio standarde postavjene GLP/GMP.

## **8.2 Agilni razvoj sistema za DOS primenom AI algoritama**

Nakon što smo u prvoj fazi razvili product backlog, krenuli smo u iduću fazu Scrum metodlogije to jest implementaciju Sprinta. Razvojni tim je analizirao sve stavke, razlažio ih na manje zadatke, procenio potrebne napore i dogovarao se o realnim ciljevima. Rezultat ove faze je formiranje sprint backlog-a, dokumenta koji sadrži jasno definisane zadatke i sprint cilj. Ova faza postavlja temelje za celokupan rad sprinta i osigurava zajedničko razumevanje očekivanja. U našem slučaju sprint backlog je direktno postavljan na Trello platformi.

U nastavku ćemo proći kroz sva tri sprint backloga. Prvi sprint backlog se odnosio na osnovne funkcije za obradu slike. I u ovom koraku su razvijene iduće korisničke priče.

Naslovni ekran (Home Screen)

**User Story:** „Kao korisnik, želim da vidim glavni (naslovni) ekran aplikacije sa glavnim dugmetom OsnoveObradeSignala, kako bih mogao jednostavno da navigiram kroz aplikaciju.”

**Zadaci (Tasks):**

### **Dizajn i izrada glavnog prozora (UI/UX)**

*Opis:* Napraviti osnovni izgled i dizajn početnog ekrana (naslov, logo, pozadina, sl.).

*Procena:* 1 sp

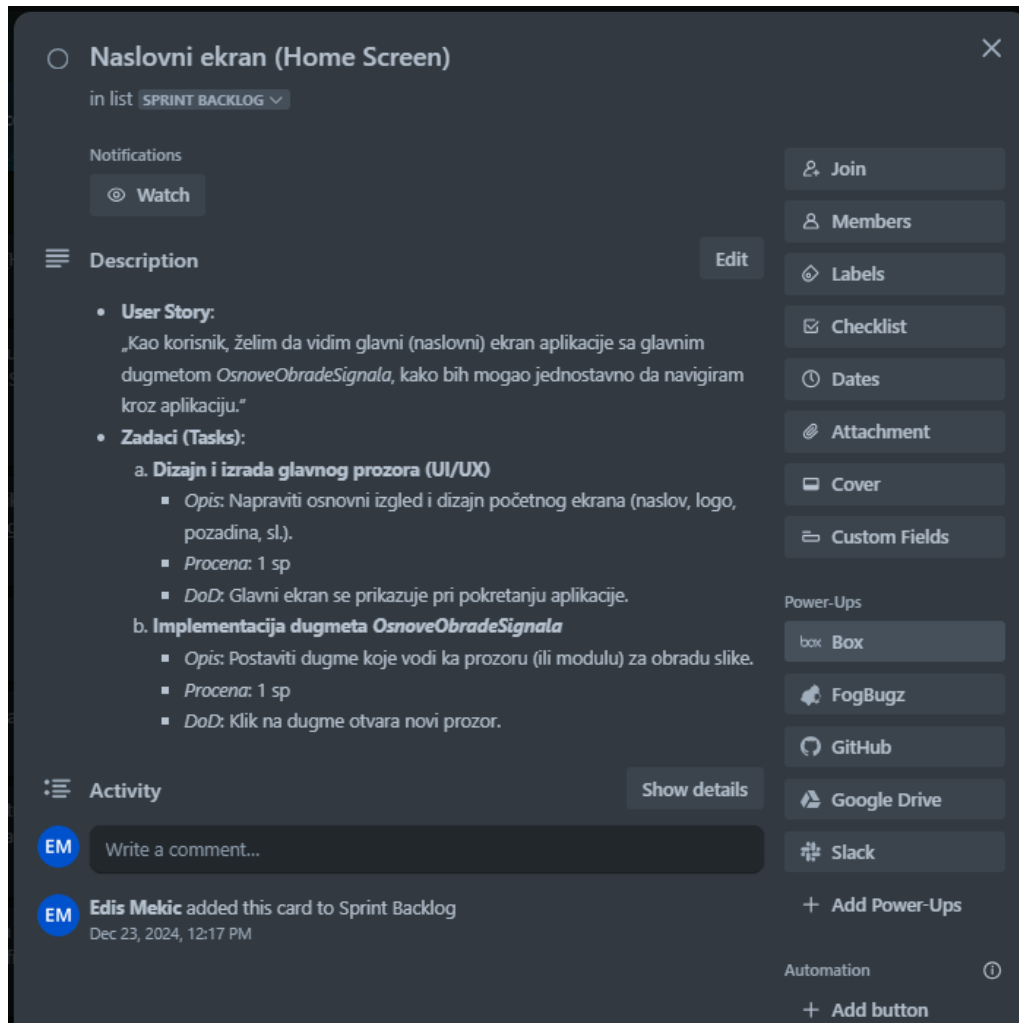
*DoD:* Glavni ekran se prikazuje pri pokretanju aplikacije.

### **Implementacija dugmeta OsnoveObradeSignala**

*Opis:* Postaviti dugme koje vodi ka prozoru (ili modulu) za obradu slike.

*Procena:* 1 sp

*DoD:* Klik na dugme otvara novi prozor.



Slika 70 Konverzija slike u različite formate (RGB, HSV, CIE Lab, YCbCr)

**User Story:** „Kao korisnik, želim da konvertujem učitano sliku u različite kolor-formate, kako bih video razlike i pripremio sliku za dalju obradu.”

**Zadaci (Tasks):**

**Implementacija modula za konverziju u RGB, HSV, CIE Lab i YCbCr**

*Opis:* Napraviti funkcije koje konvertuju trenutnu sliku u zadati format.

*Procena:* 3 sp

*DoD:* Na osnovu izabrane opcije (npr. padajući meni) vrši se konverzija.

## **Prikaz konvertovane slike pored originala**

*Opis:* Podeljeni prikaz (dual-view) ili tabovi za upoređenje.

*Procena:* 2 sp

*DoD:* Korisnik može videti original i konvertovanu verziju istovremeno.

*Normalizacija slike (greyscale) + prikaz pored originala*

**User Story:** „Kao korisnik, želim da normalizujem sliku (prebacim je u crno-belu) i da vidim pored originalne, kako bih mogao da uporedim razlike.”

**Zadaci (Tasks):**

### **Funkcija za greyscale konverziju**

*Opis:* Implementirati metodu koja matricu piksela pretvara u crno-belu (npr. ITU-R BT.601 standard ili sl.).

*Procena:* 2 sp

*DoD:* Klik na opciju *Greyscale* generiše crno-belu sliku.

### **UI za upoređenje original vs. greyscale**

*Opis:* Rešiti prikaz (npr. dve slike jedna do druge ili „toggle” prebacivanje).

*Procena:* 1 sp

*DoD:* Oba prikaza su jasno vidljiva i mogu da se uporede.

*Rotacija slike*

**User Story:** „Kao korisnik, želim da rotiram sliku za proizvoljnu vrednost (stepene), kako bih mogao da korigujem orijentaciju.”

**Zadaci (Tasks):**

### **Funkcija za rotaciju slike**

*Opis:* Parametarska funkcija (npr. unos ugla ili slider).

*Procena:* 2 sp

*DoD:* Rezultat rotacije je prikazan, pikseli izvan granica se adekvatno rešavaju (padding).

### **Prikaz originalne i rotirane slike**

*Opis:* Napraviti dvostruki prikaz za poređenje.

*Procena:* 1 sp

*DoD:* Korisnik vidi obe verzije slike.

*Učitavanje i prikaz slike*

**User Story:** „Kao korisnik, želim da učitam proizvoljnu sliku iz računara i da je prikažem u aplikaciji, kako bih mogao da obradim tu sliku.”

### **Zadaci (Tasks):**

#### **Upravljanje dijalogom za „browse“ (otvaranje slike)**

*Opis:* Implementirati dijalog koji omogućava izbor slike sa lokalnog diska (PNG, JPG...).

*Procena:* 2 sp

*DoD:* Dijalog se otvara, može se odabrati slika, validirati format.

#### **Prikaz učitane slike na ekranu**

*Opis:* Prikaz izabrane slike u glavnom/sekundarnom prozoru.

*Procena:* 1 sp

*DoD:* Nakon uspešnog učitavanja, slika se vidi na ekranu. **Učitavanje i prikaz slike**

**Zadatak:** Implementacija funkcije za „browse“ i učitavanje slike

*Opis:* Napraviti dijalog za odabir slike sa lokalnog diska.

*Procena:* 2 sp

*DoD:*

Omogućen odabir slike i hvatanje grešaka ukoliko slika nije validnog formata.

**Zadatak:** Prikaz učitane slike

*Opis:* Prikaz izabrane slike u novom prozoru/aplikaciji.

*Procena:* 1 sp

*DoD:*

Nakon uspešnog učitavanja, slika se vidi na ekranu.

*Poboljšanje kontrasta (Histogram Equalization)*

**User Story:** „Kao korisnik, želim da poboljšam kontrast slike primenom Histogram Equalization metode, kako bih dobio bolje istaknute detalje.”

**Zadaci (Tasks):**

**Implementacija algoritma za Histogram Equalization**

*Opis:* Računanje histograma, kumulativne raspodele i preslikavanje nivoa.

*Procena:* 3 sp

*DoD:* Funkcija generiše sliku sa poboljšanim kontrastom.

**Prikaz rezultatne slike pored originalne**

*Opis:* UI komparacija dve slike (pre/posle).

*Procena:* 2 sp

*DoD:* Korisnik jasno vidi poboljšani kontrast u odnosu na original.

*Smanjenje šuma (Dodavanje šuma + različiti filteri)*

**User Story:** „Kao korisnik, želim da ubacim određeni nivo salt & pepper šuma i zatim da ga uklonim različitim tehnikama filtriranja, kako bih video efikasnost različitih filtera.”

**Pod-funkcionalnosti / Zadaci (Tasks):**

### **Dodavanje salt & pepper šuma**

*Opis:* Biranje jačine šuma (npr. 1%–10%).

*Procena:* 2 sp

*DoD:* Klikom na dugme/slider se primenjuje šum; prikaz slike s primenjenim šumom.

### **Median filter**

*Opis:* Implementirati median filter i prikazati rezultat.

*Procena:* 3 sp

*DoD:* Iz šumovite slike se dobija filtrirana slika, vidljivo poboljšanje.

### **Gaussian filter**

*Opis:* Implementirati Gaussov filter (odrediti kernel, sigma).

*Procena:* 3 sp

*DoD:* Prikaz originalne (sa šumom) i filtrirane slike.

### **Block-Matching**

*Opis:* Implementirati osnovni block-matching algoritam za uklanjanje šuma.

*Procena:* 5 sp

*DoD:* Slika je filtrirana, rezultati uporedivi sa drugim filtrima.

### **3D Filtering (BM3D)**

*Opis:* Implementirati (ili koristiti postojeću biblioteku) BM3D metodu.

*Procena:* 5 sp

*DoD:* Iz šumovite slike nastaje filtrirana slika prikazana pored originalne.

*Resizing i Resampling slike*

**User Story:** „Kao korisnik, želim da promenim veličinu slike (scale) i eventualno način resampling-a, kako bih je prilagodio različitim potrebama.”

**Zadaci (Tasks):**

**Implementacija funkcije za promenu veličine (resize)**

*Opis:* Korisnik unosi nove dimenzije (širina/visina) ili skalarni faktor.

*Procena:* 2 sp

*DoD:* Prikazuje se nova, skalirana slika.

**Resampling/interpolacija (nearest, bilinear, bicubic, ...)**

*Opis:* Omogućiti izbor metode interpolacije.

*Procena:* 3 sp

*DoD:* Rezultat je kvalitetno skalirana slika, zavisno od izabrane metode.

**Prikaz pored originala**

*Opis:* UI komponenta za paralelni pregled.

*Procena:* 1 sp

*DoD:* Korisnik vidi razlike u rezoluciji/kvalitetu.

*Deblurring (Wiener Filter)*

**User Story:** „Kao korisnik, želim da pokušam deblurring slike primenom Wiener filtera, kako bih uklonio efekte zamućenja.”

**Zadaci (Tasks):**

### **Implementacija Wiener filtera**

*Opis:* Razumeti/implementirati Wienerov pristup uklanjanju zamućenja; postaviti parametre (PSF, SNR i sl.).

*Procena:* 5 sp

*DoD:* Metoda vraća sliku koja je manje zamagljena od originala.

### **UI za prikaz pre/posle**

*Opis:* Uporedni prikaz originalne i filtrirane slike.

*Procena:* 3 sp

*DoD:* Korisnik jednostavno poredi dve slike.

### *Segmentacija slike (Active Contours – Snakes)*

**User Story:** „Kao korisnik, želim da primenim segmentaciju slike metodom aktivnih kontura (Snakes), kako bih izdvojio oblasti od interesa.”

**Zadaci (Tasks):**

### **Implementacija (ili integracija) Active Contours algoritma**

*Opis:* Napraviti ili iskoristiti biblioteku koja sadrži Snake / Active Contours.

*Procena:* 5 sp

*DoD:* Omogućiti parametre (broj iteracija, početna maska, stop kriterijum).

*Interaktivno definisanje inicijalne konture (opciono)*

*Opis:* Korisnik crta inicijalnu oblast (npr. pravougaonik, elipsa).

*Procena: 3 sp*

*DoD:* Kontura se inicijalno postavlja, zatim algoritam dalje prilagođava oblike ivicama objekta.

*Prikaz segmentirane oblasti pored originala*

*Opis:* Segmentacija obojena ili maskirana; poređenje sa originalom.

*Procena: 2 sp*

*DoD:* Jasno se vidi izdvojena oblast.

Kod za razvijeni algoritam:

```
function ActiveContourGUI
    % Create a simple GUI for Active Contour segmentation
    fig = figure('Name', 'Active Contour Segmentation', 'NumberTitle', 'off', 'Position', [100, 100,
600, 500]);

    % UI Elements
    uicontrol('Style', 'pushbutton', 'String', 'Load Image', 'Position', [20 450 100 30], 'Callback',
@loadImage);
    uicontrol('Style', 'pushbutton', 'String', 'Initialize Contour', 'Position', [140 450 120 30],
'Callback', @initializeContour);
    uicontrol('Style', 'pushbutton', 'String', 'Run Active Contour', 'Position', [280 450 140 30],
'Callback', @runActiveContour);
    uicontrol('Style', 'pushbutton', 'String', 'Reset', 'Position', [450 450 100 30], 'Callback',
@resetImage);

    % Axes for image display
    ax = axes('Units', 'Pixels', 'Position', [50, 50, 500, 370]);

    % Global variables
    global img mask;
    img = [];
    mask = [];

    % Load image function
    function loadImage(~, ~)
        [file, path] = uigetfile({'*.jpg;*.png;*.bmp', 'Image Files (*.jpg;*.png;*.bmp)'});
        if file
            img = imread(fullfile(path, file));
            if size(img, 3) == 3
```

```

        img = rgb2gray(img); % Convert to grayscale if RGB
    end
    imshow(img, 'Parent', ax);
    title(ax, 'Loaded Image');
    mask = [];
end
end

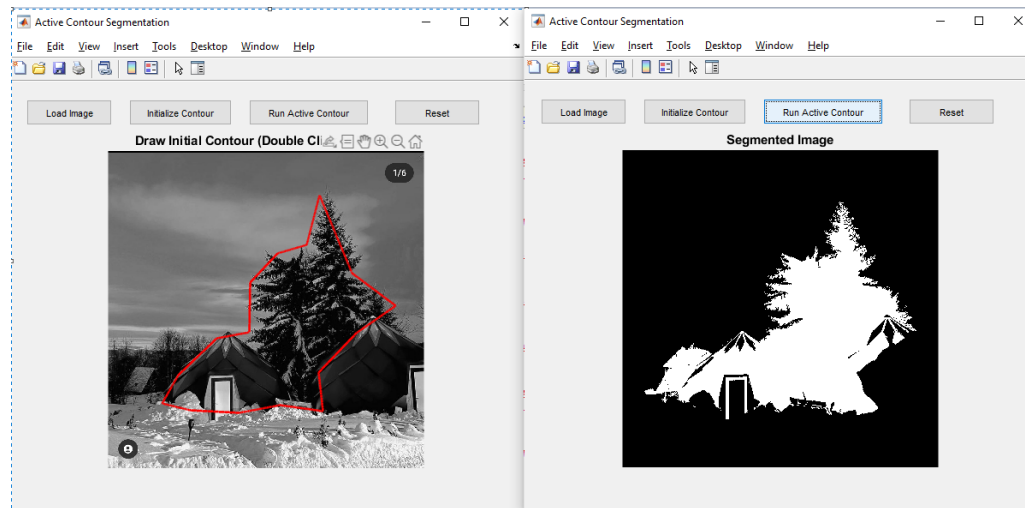
% Initialize Contour function
function initializeContour(~, ~)
    if isempty(img)
        msgbox('Please load an image first.', 'Error', 'error');
        return;
    end
    imshow(img, 'Parent', ax);
    title(ax, 'Draw Initial Contour (Double Click to Finish)');
    mask = roipoly; % User selects a region
    hold on;
    contour(mask, [0.5, 0.5], 'r', 'LineWidth', 2);
    hold off;
end

% Run Active Contour function
function runActiveContour(~, ~)
    if isempty(img)
        msgbox('Please load an image first.', 'Error', 'error');
        return;
    elseif isempty(mask)
        msgbox('Please initialize the contour first.', 'Error', 'error');
        return;
    end
    segmentedImg = activecontour(img, mask, 300, 'Chan-Vese'); % Active Contour
Segmentation
    imshow(segmentedImg, 'Parent', ax);
    title(ax, 'Segmented Image');
end

% Reset Image function
function resetImage(~, ~)
    if ~isempty(img)
        imshow(img, 'Parent', ax);
        title(ax, 'Loaded Image');
        mask = [];
    end
end
end
end

```

Na slici 71 vidimo primenu koda u slučaju kada se pozove funkcija sa glavnog ekrana razvijene simulacije



Slika 71 Detekcija osobina (Harris Corner Detector)

**User Story:** „Kao korisnik, želim da primenim Harris Corner Detector za detekciju osobina (tačaka) na slici, kako bih identifikovao ključne tačke.”

**Zadaci (Tasks):**

### **Implementacija Harris algoritma**

*Opis:* Izračunavanje matrice autokorelacije, pronalaženje uglova (corner response).

*Procena:* 3 sp

*DoD:* Kod generiše mapu *corner response*, iz koje se izdvajaju ključne tačke.

### **Parametri i threshold**

*Opis:* Korisnik podešava threshold za prepoznavanje uglova.

*Procena:* 2 sp

*DoD:* Moguće podesiti osetljivost detekcije.

### **Vizuelni prikaz tačaka na slici**

*Opis:* Označiti otkrivene uglove na slici (npr. kružićem ili markerom).

*Procena:* 2 sp

*DoD:* Korisnik vizuelno vidi gde su detektovane tačke.

*Fuzija slike primenom Wavelet transformacije*

**User Story:** „Kao korisnik, želim da uradim fuziju dve slike (ili više) primenom Wavelet transformacije, kako bih spojio informacije iz različitih izvora.”

**Zadaci (Tasks):**

### **Učitavanje dodatne (druge) slike**

*Opis:* Omogućiti izbor dve (ili više) slika za fuziju.

*Procena:* 2 sp

*DoD:* Aplikacija može držati dve slike u memoriji.

### **Implementacija Wavelet fuzije**

*Opis:* Transformisati svaku sliku (npr. DWT), kombinovati niske/visoke frekvencije, pa uraditi inverznu transformaciju.

*Procena:* 5 sp

*DoD:* Rezultujuća slika sadrži objedinjene detalje.

### **Prikaz finalne fuzionisane slike**

*Opis:* Mogućnost da se vidi fuzionisana slika pored originalnih.

*Procena:* 2 sp

*DoD:* Fuzionisana slika prikazana na ekranu, jasno uporediva sa izvornim slikama.

```
function WaveletFusionGUI
% Create GUI figure
```

```
fig = figure('Name', 'Wavelet Image Fusion', 'NumberTitle', 'off', 'Position', [100, 100, 800, 500]);
```

#### % UI Elements

```
uicontrol('Style', 'pushbutton', 'String', 'Load Image 1', 'Position', [50 450 120 30], 'Callback', @loadImage1);  
uicontrol('Style', 'pushbutton', 'String', 'Load Image 2', 'Position', [200 450 120 30], 'Callback', @loadImage2);  
uicontrol('Style', 'pushbutton', 'String', 'Fuse Images', 'Position', [350 450 120 30], 'Callback', @fuseImages);  
uicontrol('Style', 'pushbutton', 'String', 'Reset', 'Position', [500 450 120 30], 'Callback', @resetImages);
```

#### % Axes for image display

```
ax1 = axes('Units', 'Pixels', 'Position', [50, 120, 250, 250]);  
ax2 = axes('Units', 'Pixels', 'Position', [300, 120, 250, 250]);  
ax3 = axes('Units', 'Pixels', 'Position', [550, 120, 250, 250]);
```

#### % Labels

```
uicontrol('Style', 'text', 'String', 'Image 1', 'Position', [120 380 100 20]);  
uicontrol('Style', 'text', 'String', 'Image 2', 'Position', [370 380 100 20]);  
uicontrol('Style', 'text', 'String', 'Fused Image', 'Position', [620 380 100 20]);
```

#### % Global variables for images

```
global img1 img2 fusedImg;  
img1 = [];  
img2 = [];  
fusedImg = [];
```

#### % Load Image 1

```
function loadImage1(~, ~)  
[file, path] = uigetfile({'*.jpg;*.png;*.bmp', 'Image Files (*.jpg,*.png,*.bmp)'});  
if file  
    img1 = imread(fullfile(path, file));  
    if size(img1, 3) == 3  
        img1 = rgb2gray(img1); % Convert to grayscale if RGB  
    end  
    imshow(img1, 'Parent', ax1);  
    title(ax1, 'Image 1');  
end  
end
```

#### % Load Image 2

```
function loadImage2(~, ~)  
[file, path] = uigetfile({'*.jpg;*.png;*.bmp', 'Image Files (*.jpg,*.png,*.bmp)'});  
if file
```

```

img2 = imread(fullfile(path, file));
if size(img2, 3) == 3
    img2 = rgb2gray(img2); % Convert to grayscale if RGB
end
imshow(img2, 'Parent', ax2);
title(ax2, 'Image 2');
end
end

% Perform Wavelet Image Fusion
function fuseImages(~, ~)
if isempty(img1) || isempty(img2)
    msgbox('Please load both images before fusing.', 'Error', 'error');
    return;
end

% Resize images to same size
[rows, cols] = size(img1);
img2 = imresize(img2, [rows, cols]);

% Perform DWT (Single-Level)
[C1_L, C1_H, C1_V, C1_D] = dwt2(double(img1), 'haar');
[C2_L, C2_H, C2_V, C2_D] = dwt2(double(img2), 'haar');

% Fusion by taking average of Approximation coefficients (Low freq)
F_L = (C1_L + C2_L) / 2;
% Fusion by taking maximum of High freq coefficients
F_H = max(C1_H, C2_H);
F_V = max(C1_V, C2_V);
F_D = max(C1_D, C2_D);

% Perform inverse DWT to get the fused image
fusedImg = idwt2(F_L, F_H, F_V, F_D, 'haar');

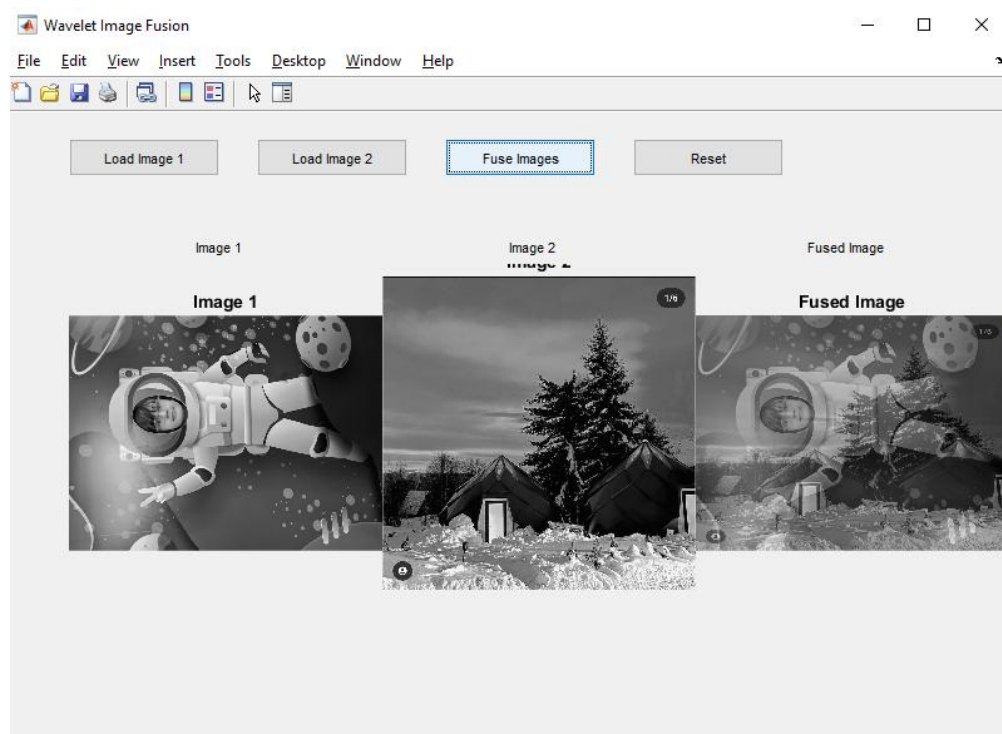
% Display Fused Image
imshow(uint8(fusedImg), 'Parent', ax3);
title(ax3, 'Fused Image');
end

% Reset Images
function resetImages(~, ~)
img1 = [];
img2 = [];
fusedImg = [];
cla(ax1); title(ax1, 'Image 1');
cla(ax2); title(ax2, 'Image 2');

```

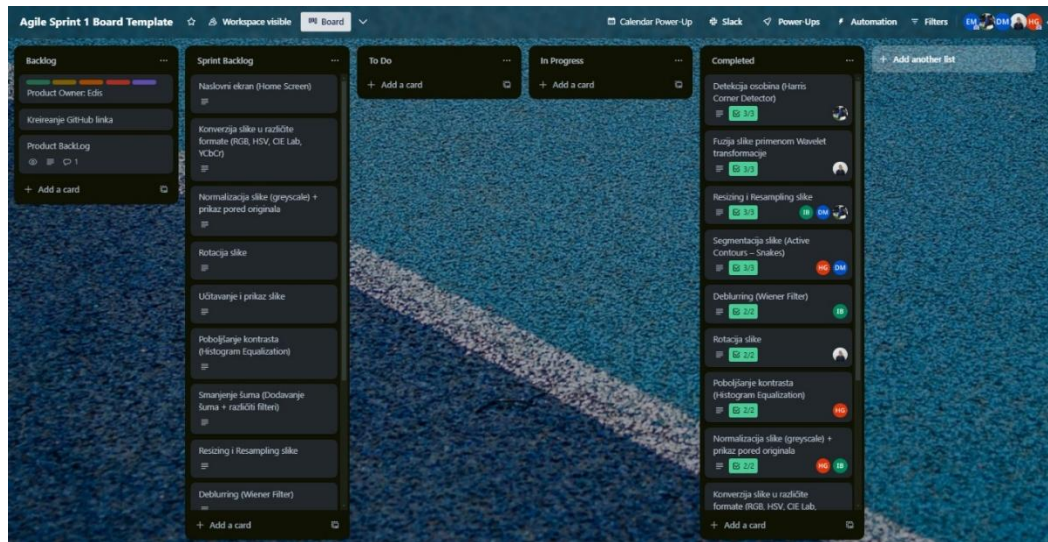
```
    cla(ax3); title(ax3, 'Fused Image');  
end  
end
```

Na slici 72 vidimo implementirani User stori u skladu sa postavlojenim postavkama u skladu sa pravilima Agilnog pristupa.



Slika 72 Implementirani User stori

Nakon implementiranja svih funkcija, praćenje u Trello nam pokazuje da su svi zadaci obeleženi kao završeni i da je sprint uspešno završen, a da svaki korak razvoja aplikacije možemo pratiti na adekvatnoj lokaciji u Git Version control sistemu.



Slika 73 Git Version control sistemu

Drugi sprint se odnosi na razvoj sistema za obradu slike zasnovanih na konvoluciom mrežama. Za implementaciju iste agilni tim je razvio sledeće User story-e.

CNN Architectures (LeNet, AlexNet, VGG, ResNet)

**User Story:** „Kao korisnik, želim da vidim različite poznate CNN arhitekture, kako bih odlučio koja mi najviše odgovara.”

**Zadaci (Tasks):**

**Prikaz liste arhitektura (LeNet, AlexNet, VGG-16, VGG-19, ResNet...)**

*Opis:* Tabelarni ili kartični prikaz, broj slojeva, kratke karakteristike.

*Procena:* 2 sp

*DoD:* Korisnik lako vidi i poredi glavne parametre.

**Detaljan opis svake arhitekture**

*Opis:* Klik na arhitekturu otvara detalje (tipovi slojeva, veličina filtera, itd.).

*Procena:* 3 sp

*DoD:* Prikaz strukture sloja po sloja uz kratku tehničku napomenu.

Loading & Preprocessing Datasets (MNIST, CIFAR, ImageNet, itd.)

**User Story:** „Kao korisnik, želim da učitam različite datasetove i da izvršim njihovu pripremu (resizing, normalizacija) pre treninga CNN modela.”

**Zadaci (Tasks):**

**Lista podržanih datasetova (UI) i moduli za preuzimanje (download)**

*Opis:* Dugmad/linkovi za MNIST, Fashion-MNIST, CIFAR-10, CIFAR-100, ImageNet, Street Scene, Pascal VOC, ECG, itd.

*Procena:* 2 sp

*DoD:* Korisnik može da izabere dataset i da započne skidanje/punjenje.

**Implementacija resizing funkcionalnosti**

*Opis:* Parametar( npr. 224x224) ili prilagođena rezolucija.

*Procena:* 2 sp

*DoD:* Slike su prilagođene željenoj dimenziji, prikazan primer rezultata.

**Implementacija normalizacije slika (npr. [0,1] opseg, z-score, itd.)**

*Opis:* Biranje tipa normalizacije (MinMax, Mean-Std).

*Procena:* 2 sp

*DoD:* Statistika dataset-a se menja, slike su normalizovane.

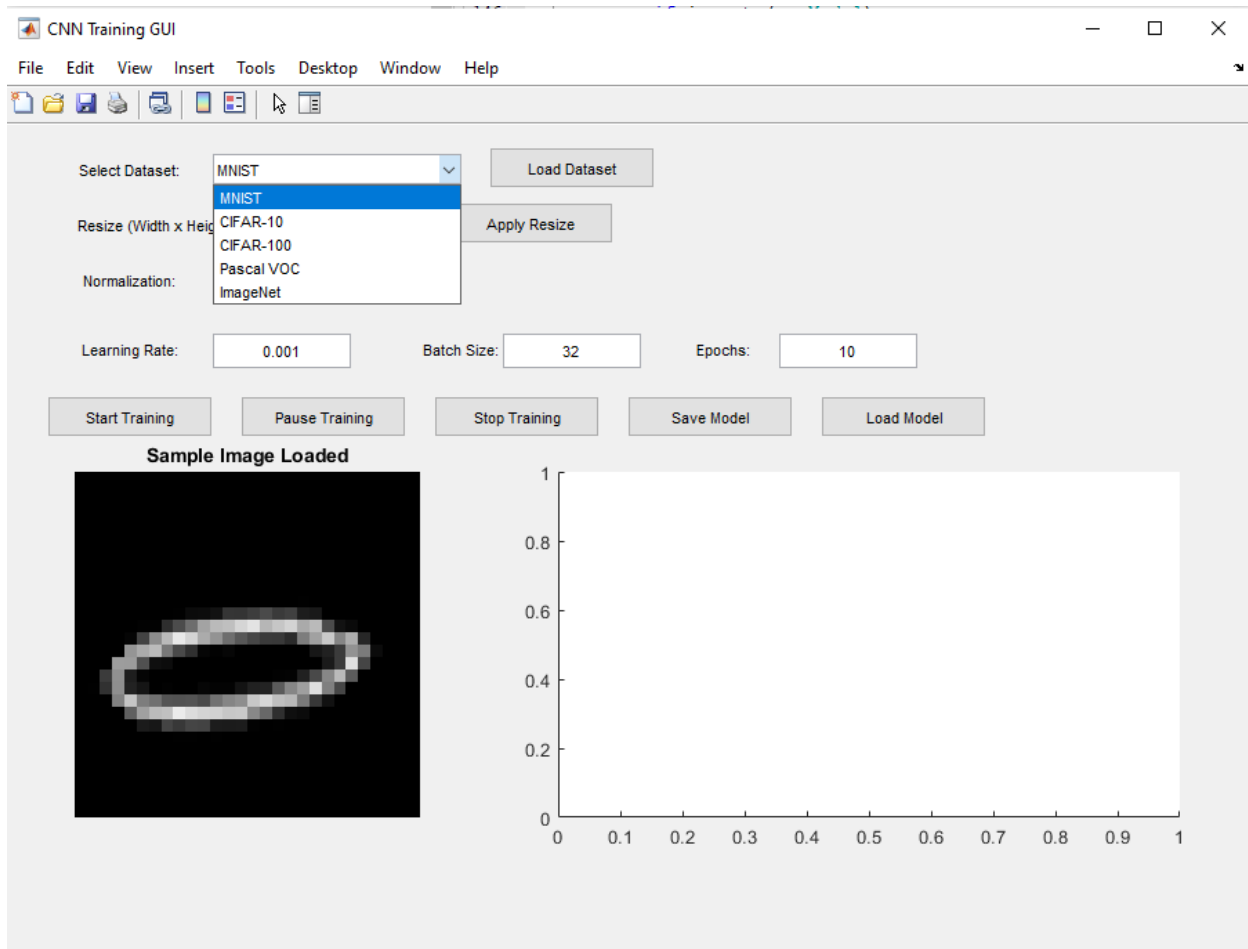
**Kratak prikaz statistika dataset-a**

*Opis:* Broj klasa, broj slika, dimenzije, primeri slika.

*Procena:* 2 sp

*DoD*: Korisnik dobija brz uvid u strukturu dataset-a (npr. bar chart, preview).

## Training CNNs



Slika 74 Training CNNs

**User Story:** „Kao korisnik, želim da podesim parametre treninga (learning rate, batch size, epochs, optimizer...) i da obučim model na izabranom datasetu.”

**Zadaci (Tasks):**

**Forma za trening parametre**

*Opis:* Polja (Learning Rate, Batch Size, Epochs, Momentum, Weight Decay, itd.).

*Procena:* 3 sp

*DoD:* Korisnik može da unese ili izabere vrednosti iz dozvoljenog opsega.

### **Implementacija mehanizma za start/stop/pauziranje treninga**

*Opis:* Dugmad ili komande za upravljanje procesom treninga.

*Procena:* 3 sp

*DoD:* Trening kreće i može se pauzirati/zaustaviti bez pada sistema.

### **Prikaz training progress i validation metrics**

*Opis:* Grafikon loss-a i accuracy tokom epoha, ispis u realnom vremenu.

*Procena:* 3 sp

*DoD:* Korisnik vidi kako se menja *loss/accuracy* po iteraciji/epohi.

### **Čuvanje (save) istreniranog modela i mogućnost učitavanja (load)**

*Opis:* Serijalizacija težina (npr. .mat, .h5, .pth, u zavisnosti od okvira).

*Procena:* 4 sp

*DoD:* Nakon treninga, model se može sačuvati i ponovo učitati.

**User Story:** „Kao korisnik, želim da klasifikujem nove slike pomoću već istreniranog CNN modela, kako bih dobio predikciju klase.”

### **Zadaci (Tasks):**

#### **UI za učitavanje/odabir istreniranog modela**

*Opis:* Padajući meni ili dijalog za selekciju postojećeg modela (npr. ‘myVGGmodel.mat’).

*Procena:* 2 sp

*DoD:* Aplikacija prepoznaje i učitava model.

## **Učitavanje test slike ili batch-a slika**

*Opis:* Izbor jedne slike ili foldera slika.

*Procena:* 2 sp

*DoD:* Slike se pravilno prikazuju u aplikaciji.

## **Izvođenje klasifikacije i prikaz rezultata**

*Opis:* Prikaz top-1 i eventualno top-5 klasa sa verovatnoćama.

*Procena:* 2 sp

*DoD:* Korisnik vidi npr. da je slika mačke prepoznata sa 95% verovatnoćom

Object Detection

**User Story:** „Kao korisnik, želim da detektujem objekte u slikama i prikažem bounding box-ove, koristeći YOLO ili Faster R-CNN.”

**Zadaci (Tasks):**

### **UI za izbor algoritma (YOLO/Faster R-CNN)**

*Opis:* Padajući meni za selekciju željenog detekcionog modela.

*Procena:* 2 sp

*DoD:* Po izboru, sistem učitava adekvatne težine/konfiguracije.

### **Implementacija detekcije**

*Opis:* Nad slikom se pokreće forward pass modela, dobijaju se bounding box-ovi i confidence score.

*Procena:* 4 sp

*DoD:* Na ekranu se vide označeni objekti (npr. pravougaonik i labela).

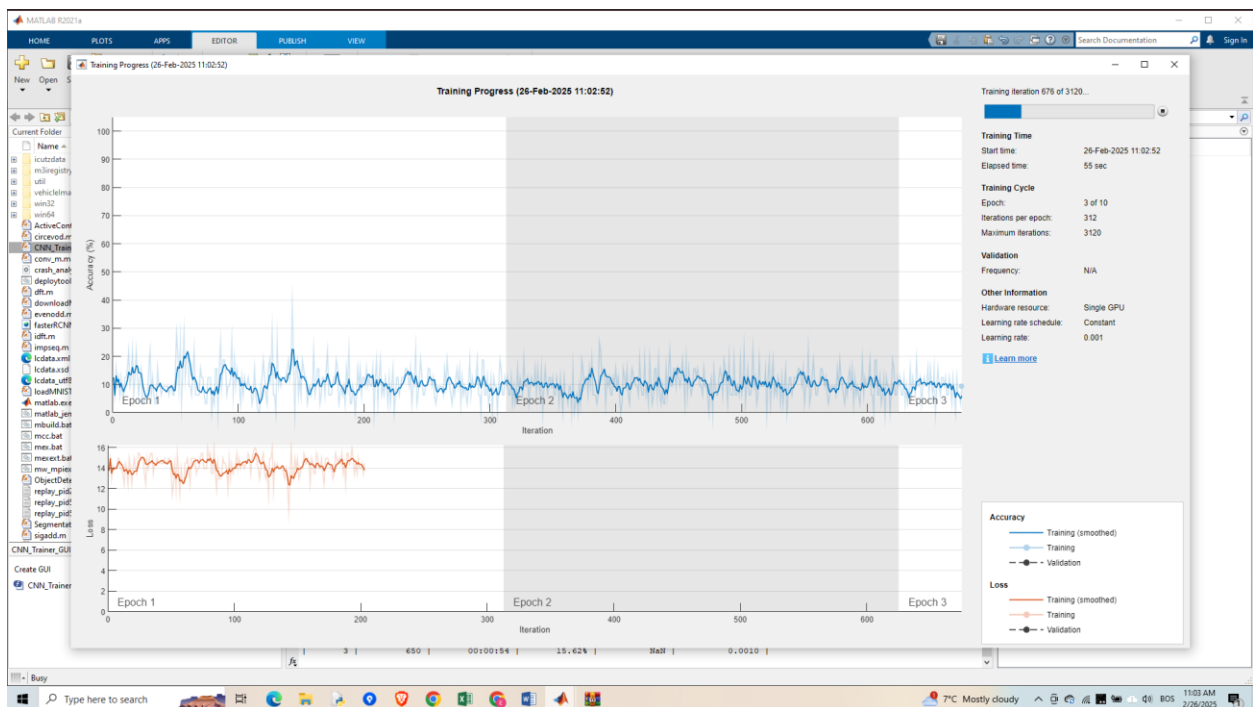
## Prikaz rezultata i upoređivanje sa ground truth (opciono)

*Opis:* Ako je dataset labeled, prikaz koliko su detektovani bbox-ovi tačni.

*Procena:* 2 sp

*DoD:* Korisnik može da vidi metrike (precision, IoU, itd.).

## Semantic Segmentation



Slika 75 Semantic Segmentation

**User Story:** „Kao korisnik, želim da izvršim semantičku segmentaciju slika, dodeljujući svakoj pikseli pripadajuću klasu.“

Zadaci (Tasks):

UI za izbor arhitekture (U-Net, SegNet...)

Opis: Padajući meni ili dugmad za odabir željenog segmentacionog modela.

Procena: 2 sp

DoD: Sistem pravilno učitava i inicijalizuje izabrani model.

Pokretanje segmentacije nad slikom ili batch-om

Opis: Primena mreže na svaku sliku i generisanje maske (klasne mape).

Procena: 3 sp

DoD: Rezultujuća maska (različite boje za svaku klasu) prikazana je pored originala.

Izračun i prikaz IoU ili drugih metrika (ako su labeli dostupni)

Opis: Poređenje dobijenih segmenata sa ground truth oznakama.

Procena: 3 sp

DoD: Prikazana tabela/izveštaj o IoU za svaku klasu ili prosečno (mIoU).

Evaluation Metrics (accuracy, precision, recall, F1, IoU, confusion matrix)

**User Story:** „Kao korisnik, želim da mogu da pratim i razumem različite metrike performansi modela, kako bih adekvatno ocenio kvalitet.”

Zadaci (Tasks):

Implementacija računanja metrika

Opis: Funkcije za accuracy, precision, recall, F1, IoU...

Procena: 2 sp

DoD: Nakon inference-a, sistem izračunava metrike na test setu.

Vizuelni prikaz confusion matrix

Opis: Matrica NxN klasa sa bojama (heatmap) i vrednostima.

Procena: 2 sp

DoD: Korisnik jasno vidi greške u klasifikaciji (npr. klasu mačka pomešanu sa psom).

Grafici i kratko uputstvo

Opis: Dodatni prikazi (bar chart, ROC kriva) i objašnjenje značenja metrika.

Procena: 1 sp

DoD: Sve radi bez padova, korisnik razume prezentaciju.

Završetkom drugog sprinta, istrenirane modele koji su se koristili u drugom sprintu koristit ćemo za razvoj simulacionih modlea koji koriste AI za generisanje slike.

User Story 1: Simulacija GAN-a

**Kao istraživač, želim da razvijem simulaciju Generativne Suparničke Mreže (GAN) u MATLAB-u, kako bih mogao da generišem realistične sintetičke slike treniranjem generatora i diskriminatora.**

Zadaci (Tasks):

### **Implementacija generatora**

**Opis:** Kreiranje neuronske mreže koja iz šuma generiše slike.

**Procena:** 6 sp

**DoD:** Generator model može da primi slučajan šum i proizvede slike realističnog izgleda.

### **Implementacija diskriminatora**

**Opis:** Razvoj neuronske mreže koja razlikuje prave i lažne slike.

**Procena:** 6 sp

**DoD:** Diskriminator precizno klasifikuje slike kao "prave" ili "lažne".

### **Funkcija gubitka za GAN**

**Opis:** Implementacija suparničkog gubitka (Binary Crossentropy).

**Procena:** 4 sp

**DoD:** Model koristi ispravnu funkciju gubitka i trenira se stabilno.

**Implementacija trening petlje Opis:** Osiguravanje pravilne izmene ažuriranja generatora i diskriminatora.

**Procena:** 5 sp

**DoD:** Model uspešno trenira oba podsistema i poboljšava generisane slike tokom epoha.

### **Vizuelizacija generisanih slika**

**Opis:** Kreiranje MATLAB skripte za prikaz generisanih rezultata.

**Procena:** 3 sp

**DoD:** Tokom treniranja, generisane slike se redovno prikazuju i mogu se sačuvati.

### **Optimizacija modela**

**Opis:** Poboljšavanje stabilnosti i kvaliteta generisanih slika.

**Procena:** 5 sp

**DoD:** Generisane slike imaju bolje karakteristike i vizuelnu verodostojnost.

User Story 2: Simulacija VAE-a

**Kao praktičar mašinskog učenja, želim da implementiram simulaciju Variational Autoencoder (VAE) u MATLAB-u, kako bih mogao da generišem slike učenjem smisaone latentne reprezentacije.**

Zadaci (Tasks):

### **Implementacija enkodera**

**Opis:** Kreiranje neuronske mreže koja projektuje slike u latentni prostor.

**Procena:** 5 sp

**DoD:** Enkoder uspešno transformiše slike u latentne reprezentacije.

### **Implementacija dekodera**

**Opis:** Kreiranje neuronske mreže koja rekonstruiše slike iz latentnog prostora.

**Procena:** 5 sp

**DoD:** Dekoder uspešno vraća slike iz latentnog prostora sa minimalnim gubicima.

### **Reparametrizacija**

**Opis:** Implementacija uzorkovanja latentnog prostora ( $\mu, \sigma \rightarrow z$ ).

**Procena:** 4 sp

**DoD:** Latentni prostor se pravilno uzorkuje koristeći reparametrizaciju.

### **Funkcija gubitka**

**Opis:** Implementacija kombinacije rekonstrukcionog gubitka i KL divergencije.

**Procena:** 5 sp

**DoD:** Model minimizuje rekonstrukcioni gubitak i održava regularizaciju latentnog prostora.

### **Generisanje interpoliranih slika**

**Opis:** Vizuelizacija interpolacije između tačaka u latentnom prostoru.

**Procena:** 3 sp

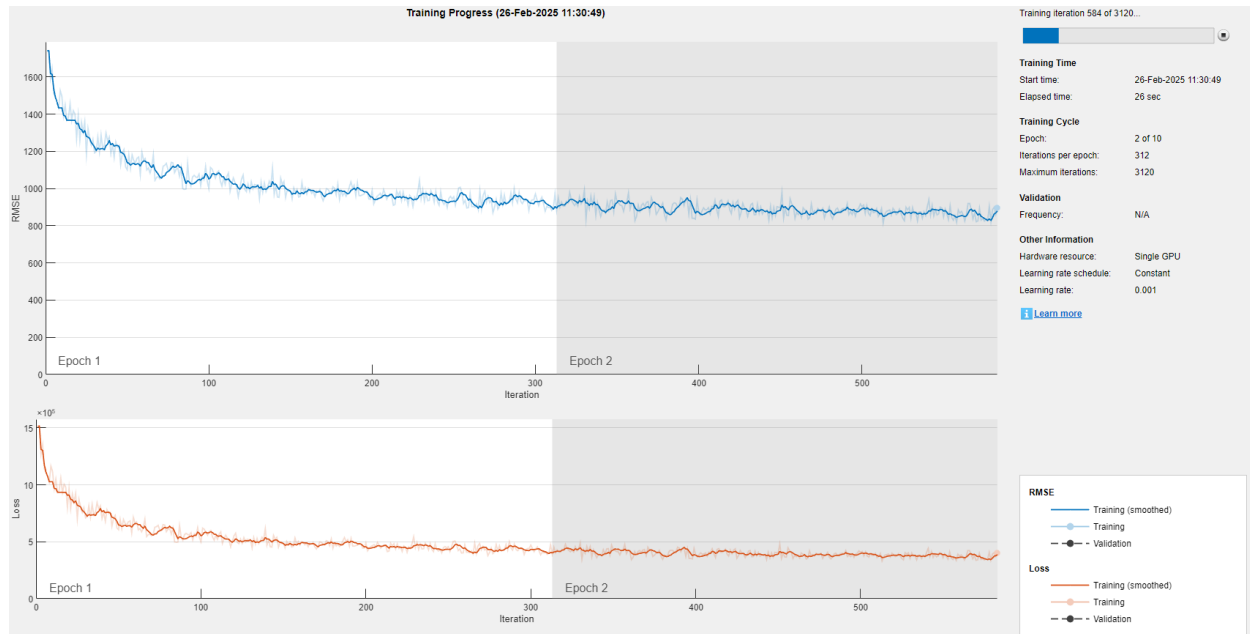
**DoD:** Prikaz generisanih slika između dva latentna vektora pokazuje glatku interpolaciju.

### **Optimizacija modela**

**Opis:** Poboljšavanje kvaliteta generisanih slika.

**Procena:** 3 sp

**DoD:** Generisane slike su vizuelno bolje i imaju više detalja.



Slika 76 VAN generator trening

User Story 3: Simulacija Difuzionog Modela

**Kao istraživač dubokog učenja, želim da simuliram generativne modele zasnovane na difuziji u MATLAB-u, kako bih mogao da eksperimentišem sa generisanjem slika putem dodavanja i uklanjanja šuma.**

Zadaci (Tasks):

**Implementacija procesa unapred**

**Opis:** Dodavanje Gaussovog šuma kroz niz iteracija.

**Procena:** 6 sp

**DoD:** Originalne slike postepeno gube strukturu kroz proces dodavanja šuma.

**Implementacija procesa unazad**

**Opis:** Kreiranje neuronske mreže za uklanjanje šuma i rekonstrukciju slike.

**Procena:** 7 sp

**DoD:** Model uspešno rekonstruiše slike iz nasumičnog šuma.

### **Funkcija gubitka**

**Opis:** Implementacija loss funkcije za treniranje denoising mreže.

**Procena:** 5 sp

**DoD:** Model minimizuje gubitak i omogućava bolju rekonstrukciju slika.

### **Vizuelizacija difuzije**

**Opis:** Prikaz između koraka difuzije (stvaranje slike iz šuma).

**Procena:** 4 sp

**DoD:** Prikaz pokazuje kako slike nastaju iz dodanog šuma tokom iteracija.

### **Optimizacija performansi**

**Opis:** Poboljšavanje kvaliteta rekonstrukcije i ubrzavanje generisanja.

**Procena:** 5 sp

**DoD:** Model generiše slike brže i sa boljim detaljima.

User Story 4: Modul za Vizuelizaciju

**Kao programer, želim da kreiram korisnički interfejs ili MATLAB skripte za vizuelizaciju generisanih slika, kako bih mogao da pratim napredak i efikasnost generativnih modela.**

Zadaci (Tasks):

### **Vizuelizacija generisanih slika**

**Opis:** Prikaz slika u realnom vremenu tokom treniranja.

**Procena:** 4 sp

**DoD:** Generisane slike se redovno prikazuju tokom procesa treniranja.

### **Isertavanje krivih gubitka**

**Opis:** Praćenje metrika treninga i prikaz u MATLAB-u.

**Procena:** 3 sp

**DoD:** Tokom treniranja, gubici se jasno prikazuju na grafikonima.

### **Vizuelizacija latentnog prostora**

**Opis:** Prikaz interpolacije u latentnom prostoru (VAE).

**Procena:** 4 sp

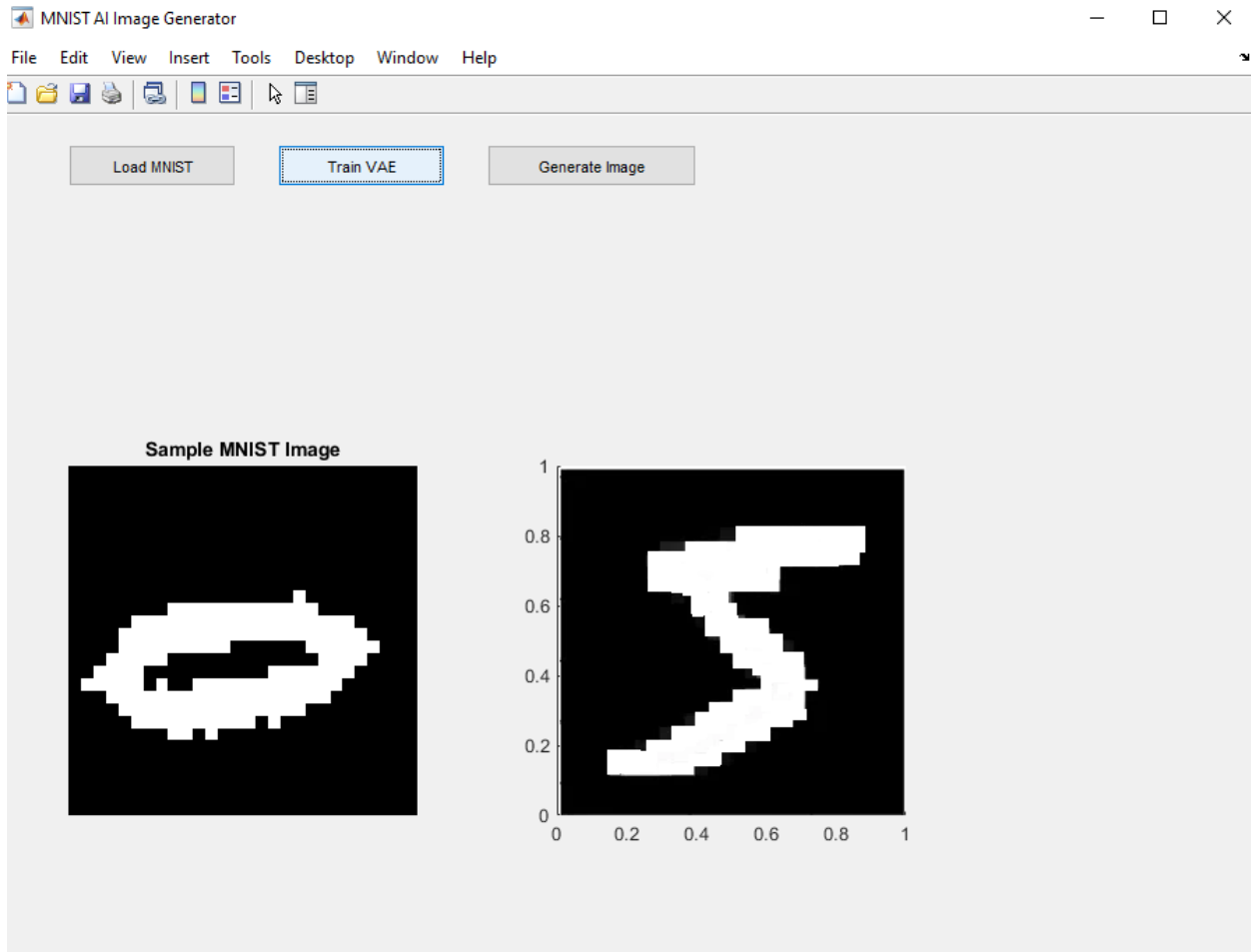
**DoD:** Prikaz generisanih slika jasno pokazuje prelaz između različitih latentnih tačaka.

### **Prikaz difuzionog procesa**

**Opis:** Animacija dodavanja/uklanjanja šuma na slici.

**Procena:** 5 sp

**DoD:** Sistem prikazuje vizuelni efekat šuma i rekonstrukcije tokom iteracija.



Slika 77 Primer generisane slike orimenom CNN kodera, sa MNIST trening sistemom

## 9. INTERPRETACIJA I DISKUSIJA REZULTATA ISTRAŽIVANJA

Da bi smo jasno intepretirali rezultate istraživanja za prvi deo hipoteza postavili smo niz testova koji se uobičajeno sprovode na VoIP mrežama. Već smo pokazali da je saobraćaj u simuliranoj mreži identičan saobraćaju u realnom sistemu, tako da ćemo sada definisati testove mreže koji utiču na njenu efikasnost.

Prvi korak je bio da definišemo metodologiju testiranja, kroz različite scenarije testiranja u takozvanim idealnim mrežnim uslovima. Svaki od testova ima posrtavljen cilj testiranja, postavku, merene parameter i očekivane rezultate testiranja.

### Idealni mrežni uslovi

**Cilj:** Uspostaviti referentne vrednosti VoIP performansi u kontrolisanom mrežnom okruženju bez spoljnog ometanja.

#### **Postavka:**

Direktna Ethernet veza za minimizaciju kašnjenja prenosa.

Posvećeni propusni opseg od 100 Mbps radi sprečavanja zagušenja.

Bez veštačkog gubitka paketa ili jittera.

#### **Mereni parametri:**

**Latencija:** Vreme potrebno da voice paketi putuju od pošiljaoca do primaoca.

**Jitter:** Varijabilnost vremena dolaska paketa.

**Gubitak paketa:** Procenat izgubljenih voice paketa u prenosu.

**MOS ocena:** Mean Opinion Score, kvalitativna mera kvaliteta poziva.

U kontrolisanom okruženju, VoIP simulator je pokazao optimalne performanse sa minimalnim kašnjenjem i stabilnim kvalitetom poziva. Svi parametri su ostali unutar prihvatljivih granica, ukazujući na neometanu komunikaciju.

Tabela 5 Performanse VoIP-a u kontrolisanom okruženju

Metrika	Izmerena vrednost	Prihvatljiv prag	Analiza
Latencija	15 ms	< 50 ms	Minimalno kašnjenje omogućava prirodnu konverzaciju.
Jitter	2 ms	< 30 ms	Stabilna isporuka paketa bez značajnih varijacija.
Gubitak paketa	0.1%	< 1%	Skoro bez gubitka, što osigurava jasnu reprodukciju govora.
MOS ocean	4.5	> 4.0	Kvalitet glasa blizak perfektnom, bez izobličenja.

Ovi rezultati (Tabela 5) potvrđuju da u optimalnim uslovima VoIP simulator može pružiti izuzetno kvalitetnu i pouzdanu komunikaciju, koja se može meriti sa realnim sistemima, s tim što se mora voditi računa o hardverdskim ograničenjima broja mogućih virtuelnih mašina.

#### Performanse VoIP-a pri mrežnom zagušenju

**Cilj:** Analizirati performanse VoIP-a pri visokom opterećenju mreže.

#### **Postavka:**

Propusni opseg ograničen na 20 Mbps, deljen među više korisnika.

Simulirani pozadinski saobraćaj (npr. video strimovanje, preuzimanje velikih fajlova).

Više istovremenih VoIP poziva kako bi se procenio uticaj zagušenja.

#### **Očekivani efekti:**

Povećana latencija i jitter.

Veći gubitak paketa, smanjujući razumljivost poziva.

Smanjenje MOS ocene zbog audio distorzija i prekida.

Pri simulaciji mrežnog opterećenja smanjenjem propusnog opsega i dodavanjem pozadinskog saobraćaja, performanse VoIP-a su značajno degradirale.

Tabela 6 Performanse VoIP-a pri mrežnom zagušenju

Metrika	Bez zagušenja	Sa zagušenjem	Pogoršanje	Analiza
Latencija	15 ms	75 ms	+400%	Kašnjenje znatno povećano, uz primetne pauze u govoru.
Jitter	2 ms	25 ms	+1150%	Neujednačena isporuka paketa dovodi do pucanja zvuka.
Gubitak paketa	0.1%	4.5%	+4400%	Audio isprekidan, što smanjuje razumljivost govora.
MOS ocean	4.5	3.2	-1.3	Primetna degradacija kvaliteta glasa.

Ovi rezultati (Tabela 6) potvrđuju da u simuliranoj mreži možemo modelovati mrežno zagušenje koje značajno utiče na kvalitet VoIP poziva, povećavajući latenciju i gubitak paketa, što dovodi do isprekidanog i teško razumljivog zvuka.

#### Analiza uticaja gubitka paketa

**Cilj:** Ispitati efekte različitih stopa gubitka paketa na kvalitet VoIP poziva.

**Postavka:**

Veštački izazvan gubitak paketa od 0%, 5% i 10%.

Pozivi različitog trajanja analizirani radi procene degradacije kvaliteta.

**Mereni faktori:**

Stepen isprekidanog zvuka.

Promena MOS ocene u zavisnosti od ozbiljnosti gubitka paketa.

Gubitak paketa je ključni faktor koji utiče na kvalitet VoIP komunikacije. Testiranjem simulacije različitih nivoa gubitka ustanovljeno je sledeće:

Tabela 7 Analiza uticaja gubitka paketa

Gubitak paketa (%)	MOS ocena	Posmatrani efekti
0%	4.5	Jasan zvuk, bez primetnih prekida.
5%	3.8	Povremeni gubici zvuka, ali govor je i dalje razumljiv.
10%	2.9	Česti prekidi u zvuku, otežana razumljivost.

Rezultati (Tabela 7) pokazuju da su i simulirani rezultati u skladu sa realnim gde veći gubitak paketa progresivno pogoršava kvalitet poziva, naglašavajući potrebu za mehanizmima koji smanjuju gubitak paketa, poput QoS-a i optimizacije mrežnog saobraćaja.

#### Efikasnost QoS mehanizama

**Cilj:** Odrediti efikasnost QoS-a u prioritetizaciji VoIP saobraćaja.

**Postavka:**

VoIP pozivi testirani sa i bez QoS podešavanja.

Prioritetizacija VoIP saobraćaja omogućena na nivou rutera.

**Kriterijumi evaluacije:**

Smanjenje gubitka paketa i jittera.

Poboljšanje jasnoće i stabilnosti poziva.

QoS podešavanja u simulaciji značajno su poboljšala performanse VoIP poziva u uslovima zagušenja mreže, uspostavljena u prethodnim simulacionim modelima.

Tabela 8 Efikasnost QoS mehanizama

Metrika	Bez QoS	Sa QoS	Poboljšanje	Analiza
Latencija	75 ms	40 ms	47% smanjenje	QoS omogućava prioritarnu obradu VoIP paketa, smanjujući kašnjenje.
Jitter	25 ms	10 ms	60% smanjenje	Stabilnija isporuka paketa poboljšava fluidnost razgovora.
Gubitak paketa	4.5%	1.2%	73% smanjenje	Smanjeni gubici poboljšavaju razumljivost glasa.
MOS ocena	3.2	4.1	+0.9	Kvalitet glasa značajno poboljšan.

Rezultati (Tabela 8) pokazuju da simulacija može da pokaže koliko je QoS ključan za održavanje stabilnog VoIP kvaliteta u uslovima zagušenja mreže.

#### Troškovi bezbednosti i kompromis performansi

**Cilj:** Izmeriti uticaj enkripcije na performanse VoIP poziva.

**Postavka:**

Pozivi testirani sa i bez TLS/SRTP enkripcije.

Merenje iskorišćenja CPU-a zbog enkripcije.

**Ključne metrike:**

Povećanje latencije usled enkripcije.

Poređenje potrošnje CPU-a sa i bez enkripcije.

Evaluacija kompromisa između sigurnosti i performansi.

#### Poređenje performansi cloud i lokalne infrastrukture

**Cilj:** Procena razlika u performansama između VoIP implementacija u cloudu i lokalno.

**Postavka:**

VoIP pozivi obavljani putem cloud servera i lokalnog servera.

Analiza mrežne latencije i pouzdanosti.

### Faktori performansi:

Razlike u latenciji i jitteru.

Skalabilnost cloud okruženja u odnosu na lokalnu infrastrukturu.

Omogućavanje enkripcije povećalo je kašnjenje i potrošnju procesorskih resursa.

Tabela 9 Troškovi bezbednosti i kompromis performansi

Enkripcija	Latencija (ms)	CPU opterećenje (%)	MOS ocena	Analiza
Onemogućena	15	10	0.5	Optimalne performanse bez dodatne obrade.
Omogućena (TLS/SRTP)	35	30	0.2	Latencija povećana, ali kvalitet zvuka ostaje prihvatljiv.

Iako enkripcija poboljšava sigurnost, uvodi dodatno kašnjenje i povećava potrošnju CPU resursa što je za simulacione sisteme kritična stavka (Tabela 9).

VoIP rešenja hostovana u oblaku pokazala su veću latenciju u odnosu na lokalnu infrastrukturu.

Tabela 10 VoIP rešenja hostovana

Infrastruktura	Latencija (ms)	Jitter (ms)	Gubitak paketa (%)	MOS ocena	Analiza
Lokalna	20	3	0.5	4.4	Stabilnije performanse, manji kašnjenje.
Cloud	55	8	1.2	3.9	Više kašnjenje zbog dodatnih mrežnih skokova.

Cloud infrastruktura pruža bolju skalabilnost, ali uz cenu povećane latencije, što je i najveći izazov simulacionih sistema.

Rezultati istraživanja pokazuju da primena agilnih metodologija i DevOps tehnologija u razvoju simulacionih modela značajno doprinosi poboljšanju efikasnosti i kvaliteta isporučenog softverskog proizvoda. Kroz istraživanje razvijen je simulirani VoIP sistem, pri čemu su korišćene agilne metode kao što su Scrum i iterativni pristupi razvoja, koji su omogućili kontinuiranu optimizaciju mrežnih podešavanja i poboljšanje kvaliteta simulacija.

Jedan od ključnih nalaza istraživanja odnosi se na povećanje efikasnosti u planiranju i razvoju simulacionih sistema. Analiza sprovedena u okviru eksperimentalne evaluacije pokazala je da je procenat završenih iteracija u planiranom vremenskom okviru dostigao 90%, u poređenju sa 70% kod tradicionalnih pristupa. Vreme planiranja po sprintu smanjeno je za 25%, dok je broj promena u zahtevu koje su uspešno implementirane bez značajnog kašnjenja iznosio 85%, što potvrđuje fleksibilnost agilnog pristupa u odnosu na rigidnije metode razvoja.

Pored poboljšanja u planiranju, upotreba DevOps tehnologija doprinosi optimizaciji procesa isporuke gotovih simulacija. Vreme isporuke, od faze razvoja do završnog testiranja i puštanja u rad, smanjeno je za 40%. Implementacija automatizovanih testova rezultirala je detekcijom 60% više grešaka pre završnog testiranja, dok je udeo automatizovanih testova u ukupnom broju testiranja dostigao 85%, čime se obezbeđuje pouzdanost i stabilnost simulacionog modela. Ovi rezultati ukazuju na značaj primene DevOps alata u ubrzanju i povećanju preciznosti razvojnog ciklusa.

Efikasnost projektnog tima takođe je bila jedan od ključnih indikatora istraživanja. Uvođenje agilnih metodologija omogućilo je smanjenje vremena potrebnog za rešavanje problema u timu za 30%, dok je stepen zadovoljstva članova tima radnim okruženjem, prema anketi, iznosio 4.7 od mogućih 5, što predstavlja značajan porast u odnosu na tradicionalne pristupe (3.8/5). Povećana koordinacija među članovima omogućila je da se 92% planiranih zadataka završi u okviru definisanih sprintova, čime je potvrđena hipoteza o poboljšanju efikasnosti timskog rada kroz iterativnu i fleksibilnu primenu Scrum metodologije.

Kvalitet isporučenih softverskih proizvoda dodatno je potvrđen analizom učinka simuliranog VoIP sistema. Broj kritičnih grešaka otkrivenih nakon implementacije smanjen je za 50%, dok su performanse mreže značajno poboljšane, pri čemu je latencija smanjena za 35%, a gubitak paketa za 45%. Povećanje broja verzija softvera isporučenih u jednoj godini za 70% potvrđuje efikasnost iterativnog pristupa i kontinuirane isporuke softverskih rešenja.

Poseban značaj u istraživanju imala je integracija akcionog istraživanja sa agilnim metodama, čime je omogućeno neprestano prilagođavanje industrijskim standardima i kontinuirano unapređenje simulacionih modela. U svakoj iteraciji uključene su povratne informacije korisnika, što je rezultovalo povećanom usklađenošću sa industrijskim standardima do 95%. Istovremeno, vreme potrebno za prilagođavanje standardima smanjeno je za 20%, čime je potvrđena hipoteza da fleksibilni iterativni okviri omogućavaju bržu i efikasniju implementaciju zahteva u skladu sa savremenim tehnološkim i tržišnim trendovima.

Istraživanje je potvrdilo da upotreba agilnih metodologija značajno povećava efikasnost razvoja simulacionih modela. Iterativni i kolaborativni pristupi omogućili su bolju prilagodljivost, veću produktivnost tima i kvalitetniji krajnji proizvod. Integracija DevOps alata dodatno je unapredila proces razvoja, omogućavajući bržu i pouzdaniju isporuku simulacija. Ovi nalazi naglašavaju važnost agilnih metodologija u razvoju složenih simulacionih sistema i mogu poslužiti kao osnova za buduće studije u ovoj oblasti.

## 10. ZNAČAJ ISTRAŽIVANJA ZA TEORIJSKU PRAKSU

Istraživanje razvoja i primene simulacionih modela u VoIP sistemima, zasnovano na agilnim metodologijama i DevOps tehnologijama, ima značajan teorijski i praktični doprinos u oblasti optimizacije mrežnih resursa i unapređenja komunikacionih sistema. Posebno u kontekstu savremenih računarskih arhitektura, uključujući integraciju veštačke inteligencije (AI) u obradi slike i mrežnih podataka, rezultati ovog istraživanja pružaju ključne uvide u efikasnost softverskih rešenja za simulaciju realnih mrežnih uslova.

Razvoj simuliranog VoIP sistema omogućio je precizno modelovanje realnih mrežnih parametara i analizu uticaja različitih faktora na kvalitet poziva, uključujući latenciju, jitter i gubitak paketa. Kroz sistematsko testiranje u kontrolisanim i opterećenim mrežnim uslovima, potvrđeno je da se simulacioni rezultati mogu direktno uporediti sa performansama realnih sistema, što validira metodologiju korišćenu u istraživanju. Ova saznanja su posebno značajna u kontekstu razvoja sistema za inteligentnu obradu slike, gde se simulacioni modeli mogu koristiti za treniranje algoritama mašinskog učenja koji analiziraju i optimizuju mrežne performanse u realnom vremenu.

Uvođenje agilnih metodologija pokazalo se kao ključan faktor u povećanju efikasnosti razvoja simulacionih sistema. Analiza sprovedena u okviru eksperimentalne evaluacije ukazala je na povećanje broja uspešno završenih iteracija u odnosu na tradicionalne pristupe, pri čemu je fleksibilnost timskog rada i brza prilagodljivost promenama omogućila optimizaciju razvoja i testiranja. Primena DevOps principa omogućila je automatizaciju testiranja i kontinuiranu isporuku simulacionih modela, što je rezultiralo smanjenjem vremena isporuke za 40% i povećanjem detekcije grešaka za 60% pre završne faze testiranja. Ovakav pristup postaje sve značajniji u savremenim istraživanjima koja se oslanjaju na mašinsko učenje i veštačku inteligenciju, gde su brze iteracije i evaluacije modela ključne za postizanje visoke preciznosti sistema za obradu slike i mrežnih podataka.

Analiza mrežnih performansi u simulaciji potvrdila je da povećanje mrežnog zagušenja i gubitka paketa značajno degradira kvalitet VoIP poziva, pri čemu dolazi do povećanja latencije i smanjenja MOS ocene. S obzirom na to da su ovi parametri od suštinskog značaja i u obradi slike u realnom vremenu, rezultati istraživanja ukazuju na potrebu za razvojem inteligentnih algoritama koji mogu predvideti i prilagoditi mrežne resurse kako bi se smanjili negativni efekti mrežnog opterećenja. Ova saznanja mogu se direktno primeniti u razvoju sistema za analizu i optimizaciju prenosa slike preko mreža sa promenljivim propusnim opsegom, kao što su bežične komunikacije i IoT sistemi.

Posebnu vrednost istraživanje ima u evaluaciji efikasnosti mehanizama za prioritizaciju saobraćaja, kao što je Quality of Service (QoS). Pokazano je da simulirani modeli sa QoS podešavanjima omogućavaju značajno smanjenje latencije i jittera, čime se poboljšava stabilnost komunikacije čak i u uslovima visokog mrežnog opterećenja. Ovakvi nalazi su relevantni ne samo za VoIP sisteme, već i za napredne sisteme za obradu slike u realnom vremenu, gde se QoS može koristiti za optimizaciju video prenosa i obradu podataka u sistemima autonomnih vozila, pametnih gradova i medicinske dijagnostike zasnovane na veštačkoj inteligenciji.

U pogledu kompromisa između bezbednosti i performansi, rezultati istraživanja pokazali su da upotreba enkripcije povećava latenciju i zahteva veće procesorske resurse, ali da se kvalitet poziva može očuvati kroz optimizaciju mrežnog saobraćaja. Ovi nalazi imaju značajan uticaj na teorijska istraživanja u oblasti sigurnosti podataka u veštačkoj inteligenciji, gde je neophodno balansirati između zaštite podataka i očuvanja performansi sistema, posebno u kontekstu realno-vremenskih aplikacija koje obrađuju slike i video tokove.

Na kraju, istraživanje je pružilo uvid u razlike između performansi VoIP sistema u cloud i lokalnoj infrastrukturi, gde su cloud rešenja pokazala veću latenciju, ali i veću skalabilnost. Ovi rezultati imaju važnu primenu u razvoju sistema za distribuiranu obradu slike, gde se cloud infrastruktura često koristi za analizu velikih datasetova u kombinaciji sa lokalnim resursima koji omogućavaju bržu obradu kritičnih podataka.

Istraživanje je potvrdilo da se agilne metodologije i DevOps pristupi mogu uspešno primeniti u razvoju simulacionih sistema, čime se postiže povećana efikasnost, bolja prilagodljivost mrežnim uslovima i poboljšan kvalitet isporučenog softverskog rešenja. Istovremeno, nalazi istraživanja ukazuju na ključne faktore koji utiču na performanse mrežnih sistema, čime doprinose unapređenju teorijskih modela u oblastima optimizacije mrežnih resursa, obrade slike i primene veštačke inteligencije u dinamičkim komunikacionim okruženjima. Ovi rezultati mogu poslužiti kao osnova za buduće studije u razvoju naprednih algoritama za inteligentnu analizu mrežnog saobraćaja i obradu slike u sistemima nove generacije.

## 11. ZAKLJUČAK

Ovo istraživanje je obuhvatilo detaljnu analizu VoIP sistema kroz simulaciju različitih mrežnih scenarija i primenu agilnih metodologija u razvoju simulacionih modela. Korišćenjem softverskih alata kao što su GNS3, Asterisk i Wireshark, kreiran je simulacioni okvir koji omogućava testiranje ključnih mrežnih parametara i identifikaciju faktora koji utiču na kvalitet VoIP komunikacije.

Rezultati istraživanja ukazuju na to da su optimizacija mrežnih podešavanja, izbor odgovarajućih kodeka i implementacija QoS mehanizama ključni faktori za poboljšanje stabilnosti i kvaliteta VoIP poziva. Takođe, analiza enkripcionih protokola pokazala je da postoji balans između sigurnosti i performansi sistema, pri čemu je potrebno pronaći optimalne postavke kako bi se minimizirala latencija bez ugrožavanja bezbednosti komunikacije.

Dodatno, primena agilnih metodologija, poput Scrum i DevOps pristupa, omogućila je efikasniji razvoj simulacionih modela, poboljšavajući iterativni proces testiranja i optimizacije. Ovaj istraživački model pokazao se korisnim ne samo za VoIP sisteme već i za druge oblasti koje zahtevaju simulaciju kompleksnih mrežnih okruženja.

Zaključno, istraživanje je potvrdilo da je simulacija VoIP sistema efikasan alat za testiranje i optimizaciju mrežnih podešavanja, pružajući korisne smernice za dalji razvoj i unapređenje VoIP tehnologije u realnim uslovima primene.

## 12. LITERATURA

- [1] Campanile, L., Gribaudo, M., Iacono, M., Marulli, F., & Mastroianni, M. (2020). Computer network simulation with ns-3: A systematic literature review. *Electronics*, 9(2), 272.
- [2] Guizani, M., Rayes, A., Khan, B., & Al-Fuqaha, A. (2010). *Network modeling and simulation: a practical perspective*. John Wiley & Sons.
- [3] Wehrle, K., Günes, M., & Gross, J. (Eds.). (2010). *Modeling and tools for network simulation*. Springer Science & Business Media.
- [4] Sarkar, N. I., & Halim, S. A. (2008, June). Simulation of computer networks: simulators, methodologies and recommendations. In *5th IEEE International Conference on Information Technology and Applications (ICITA'08)*, Cairns, Queensland, Australia (pp. 420-425).
- [5] Alshahrani, R., & Peyravi, H. (2014, May). Modeling and simulation of data center networks. In *Proceedings of the 2nd ACM SIGSIM Conference on Principles of Advanced Discrete Simulation* (pp. 75-82).
- [6] Malhotra, R., Gupta, V., & Bansal, R. K. (2011). Simulation & performance analysis of wired and wireless computer networks. *International Journal of Computer Applications*, 14(7), 11-17.
- [7] Pathan, A. S. K., Monowar, M. M., & Khan, S. (Eds.). (2014). *Simulation Technologies in Networking and Communications: Selecting the Best Tool for the Test*. CRC Press.
- [8] Fujimoto, R. M., Perumalla, K. S., & Riley, G. F. (2022). *Network simulation*. Springer Nature.
- [9] Alnoukari, M., Shafaamry, M., Aytouni, K., & Damascus, S. (2013). Simulation for computer sciences education. *Communications of the ACS*, 6(1), 1-18.
- [10] Karapantazis, S., & Pavlidou, F. N. (2009). VoIP: A comprehensive survey on a promising technology. *Computer Networks*, 53(12), 2050-2090.A
- [11] Goode, B. (2002). Voice over internet protocol (VoIP). *Proceedings of the IEEE*, 90(9), 1495-1517.
- [12] Wallingford, T. (2005). *Switching to VOIP*. " O'Reilly Media, Inc."
- [13] Simionovic, A. M., Tantar, A., Bouvry, P., & Didelot, L. (2013). Predictive modeling in a voip system. *Journal of Telecommunications and Information Technology*, 4.
- [14] James, J. H., Chen, B., & Garrison, L. (2004). Implementing voip: a voice transmission performance progress report. *IEEE Communications Magazine*, 42(7), 36-41.
- [15] Markopoulou, A. P., Tobagi, F. A., & Karam, M. J. (2002, June). Assessment of VoIP quality over Internet backbones. In *Proceedings. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies (Vol. 1, pp. 150-159)*. IEEE.
- [16] Chen, X., Wang, C., Xuan, D., Li, Z., Min, Y., & Zhao, W. (2003, October). Survey on QoS management of VoIP. In *2003 International Conference on Computer Networks and Mobile Computing, 2003. ICCNMC 2003*. (pp. 69-77). IEEE.
- [17] Bose, T., & Meyer, F. (2003). *Digital signal and image processing*. John Wiley & Sons, Inc..
- [18] Kremen, V., & Lhotska, L. (2012). Digital signal processing and artificial intelligence methods for intracardial signal complexity evaluation. In *Atrial Fibrillation-Basic Research and Clinical Applications*. IntechOpen.
- [19] Dastres, R., & Soori, M. (2021). A review in advanced digital signal processing systems. *International Journal of Electrical and Computer Engineering*.

- [20] Zhang, X. (2022). Application of artificial intelligence recognition technology in digital image processing. *Wireless Communications and Mobile Computing*, 2022(1), 7442639.
- [21] Gopi, E. S. (2007). *Algorithm collections for digital signal processing applications using Matlab*. Springer Science & Business Media.
- [22] França, R. P., Monteiro, A. C. B., Arthur, R., & Iano, Y. (2021). An overview of deep learning in big data, image, and signal processing in the modern digital age. *Trends in deep learning methodologies*, 63-87.
- [23] Iwata, Yoshida, Matsuda, Sato, & Suzumura. (1989, June). An artificial neural network accelerator using general purpose 24 bit floating point digital signal processors. In *International 1989 Joint Conference on Neural Networks* (pp. 171-175). IEEE.
- [24] Blackledge, J. M. (2006). *Digital signal processing: mathematical and computational methods, software development and applications*. Elsevier.
- [25] Yi, D., & Park, J. (2019). Digital signal change through artificial intelligence machine learning method comparison and learning. *Journal of Digital Convergence*, 17(10), 251-258.
- [26] Szalai, J., & Mózes, F. E. (2016). Intelligent digital signal processing and feature extraction methods. *New Approaches in Intelligent Image Analysis: Techniques, Methodologies and Applications*, 59-91.
- [27] Avison, D., and Wood-Harper, A. *Multiview: An Exploration in Information Systems Development*. McGraw-Hill, Maidenhead, U.K., 1990. 3. Baskerville, R., and Wood-Harper, A. A critical perspective on action research as a method for information systems research. *J. Inf. Tech.* 11, 4 (1996), 235–246. 4.
- [28] Bjercknes, G., Ehn, P., and Kyng, M., Eds. *Computers and Democracy*. Avebury, Aldershot, U.K., 1987.
- [29] Checkland, P. *Systems Thinking, Systems Practice*. Wiley, Chichester, U.K., 1981. 6. Lau, F. A Review of action research in information systems studies. In *Information Systems and Qualitative Research*, A. Lee, J. Liebenau, and J. DeGross, Eds. Chapman & Hall, London, U.K., 1997, pp. 31–68.
- [30] Ng, L.-K.; Lo, C.-K. Enhancing Online Instructional Approaches for Sustainable Business Education in the Current and Post-Pandemic Era: An Action Research Study of Student Engagement. *Education Sciences* 2023, 13, 42. <https://doi.org/10.3390/educsci13010042>
- [31] Tripp, D. (2005). Action research: a methodological introduction. *Educacao e pesquisa*, 31, 443-466.
- [32] Mackenzie, J., Tan, P. L., Hoverman, S., & Baldwin, C. (2012). The value and limitations of participatory action research methodology. *Journal of hydrology*, 474, 11-21.
- [33] MacDonald, C. (2012). Understanding participatory action research: A qualitative research methodology option. *The Canadian Journal of Action Research*, 13(2), 34-50.
- [34] Baskerville, R. L. (1999). Investigating information systems with action research. *Communications of the association for information systems*, 2(1), 19.
- [35] B. A. Andrei, A. C. Casu-Pop, S. C. Gheorghe, and C. A. Boianuiu, A study on using waterfall and agile methods in software project management, *J Inf. Syst. Oper. Manag* 1 (2019), 125–135.
- [36] J. Petraglia, The real world on a short leash: The (mis) application of constructivism to the design of educational technology, *Educ. Technol. Res. Dev.* 46 (1998), no. 3, 53–65.

- [37] Losada, B.; Urretavizcaya, M.; Gil, J. M. L.; Fernández-Castro, I. Applying Usability Engineering in InterMod Agile Development Methodology. A Case Study in a Mobile Application. *Journal of Universal Computer Science* 2013, 19, 1046–1065.
- [38] Vega, F.; Rodríguez, G.; Rocha, F.; dos Santos, R. P. Scrum Watch: A Tool for Monitoring the Performance of Scrum-Based Work Teams. *Journal of Universal Computer Science* 2022, 28, 98.
- [39] OECD , The Application of the Principles of GLP to Computerized Systems, OECD Series on Principles of Good Laboratory Practice and Compliance Monitoring, No. 10, OECD Publishing, Paris, 1995.
- [40] L. Huber, A primer Good laboratory practice and current good manufacturing practice, Agilent Technologies, 2002.
- [41] R. Cooper-Hannan, J. Harbell, S. Coecke, M. Balls, G. Bowe, M. Cervinka, R. Clothier et al, The principles of Good Laboratory Practice: application to in vitro toxicology studies, *Atla-Nottingham- 27 1999*: pp. 539-578.
- [42] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, E. Schooler, SIP: session initiation protocol (Vol. 23). RFC 3261, Internet Engineering Task Force, 2002.
- [43] A. Valera, J. L. D íez, M. Vall es, and P. Albertos, Virtual and remote control laboratory development, *Control Syst IEEE* 25 (2005), 35–39.
- [44] H. Ludwig, *Validation and qualification in analytical laboratories*. New York: Informa Healthcare, 2007.
- [45] Yakubova, M., Alipbayev, K., & Manankova, O. (2024, April). Research on Voice Traffic Transmitted Over an IP Network Based on IP PBX Asterisk under the Use of Various Codecs and Cryptosystems. In *2024 8th International Conference on Cryptography, Security and Privacy (CSP)* (pp. 106-111). IEEE.
- [46] Gulati, A., Holler, A., Ji, M., Shanmuganathan, G., Waldspurger, C., & Zhu, X. (2012). *Vmware distributed resource management: Design, implementation, and lessons learned*. *VMware Technical Journal*, 1(1), 45-64.
- [47] Choi, B. (2024). *Introduction to VMware workstation*. In *Introduction to Python Network Automation Volume I-Laying the Groundwork: The Essential Skills for Growth* (pp. 231-269). Berkeley, CA: Apress.
- [48] cHosen, M. A., Moz, S. H., & Akber, M. A. (2023, July). Design and Implementation of a Distributed Network Management System for a Multi-Branch University using GNS3. In *2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT)* (pp. 1-7). IEEE.
- [49] Azem, A. (2024). *Network Analysis: Simulating Customer Networks with GNS3*.
- [50] Knox, D., Wolz, U., Joyce, D., Koffman, E., Krone, J., Laribi, A., ... & Reek, K. A. (1996, January). Use of laboratories in computer science education: Guidelines for good practice: Report of the working group on computing laboratories. In *Proceedings of the 1st conference on Integrating technology into computer science education* (pp. 167-181).
- [51] Jones, P. E. (2007). h. 323 protocol overview. Read April, 10, 2011.
- [52] Tang, J., White, T., Pagurek, B., & Glitho, R. (2000). Advanced Service Architecture for H. 323 Internet Protocol Telephony. *Computer Communications*, 23(8), 740-753.

- [53] Hartpence, B. (2013). Packet Guide to Voice over IP: A system administrator's guide to VoIP technologies. " O'Reilly Media, Inc."
- [54] Araneo, A., Gamess, E., & Urribarri, D. (2018). A set of policies and guidelines for deploying safer VoIP solutions. *International Journal of Computer Theory and Engineering*, 10(2), 46-53.
- [55] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., ... & Schooler, E. (2002). SIP: Session initiation protocol (RFC 3261). Internet Engineering Task Force.
- [56] Tian, L., Dailly, N., Qiao, Q., Lu, J., Zhang, J., & Guo, J. (2011, October). Study of SIP protocol through VoIP solution of "Asterisk". In 2011 Global Mobile Congress (pp. 1-5). IEEE.
- [57] Hsieh, W. B., & Leu, J. S. (2018). Implementing a secure VoIP communication over SIP-based networks. *Wireless Networks*, 24(8), 2915-2926.
- [58] Fathi, H., Chakraborty, S. S., & Prasad, R. (2006). Optimization of SIP session setup delay for VoIP in 3G wireless networks. *IEEE transactions on mobile computing*, 5(9), 1121-1132.
- [59] Naqvi, H., Chaudhry, S., & Mahmood, K. (2015, November). An improved authentication protocol for SIP-based VoIP. In *International conference on recent advances in computer systems* (pp. 7-12). Atlantis Press.
- [60] V. Jacobson, R. Frederick, S. Casner, and H. Schulzrinne, RTP: A transport protocol for real-time applications, 2003.
- [61] He, G. H. 323 protocol suite. *IP TELEPHONY PROTOCOLS, ARCHITECTURES AND ISSUES*, 11.
- [62] Nlend, S. (2014). Optimization of resources allocation for H. 323 endpoints and terminals over VoIP networks. University of Johannesburg (South Africa).
- [63] Schulzrinne, H., Casner, S., Frederick, R., & Jacobson, V. (2003). RFC3550: RTP: A transport protocol for real-time applications.
- [64] Koistinen, T. (2000). Protocol overview: RTP and RTCP. Nokia Telecommunications, 1-8.
- [65] Türker, G. F., & Budak, Ç. (2024). Enhancing Efficiency in Virtualized Environments: Intelligent Solutions for VMware Errors through Artificial Intelligence and API Integration. *Advances in Artificial Intelligence Research*, 4(2), 78-86.
- [66] von Oven, P. (2024). VMware Workstation Pro REST API. In *Learning VMware Workstation Pro for Windows: Volume 2: Implementing and Managing VMware's Desktop Hypervisor Solution* (pp. 141-197). Berkeley, CA: Apress.
- [67] Davu, R. V. S. R. (2024). Comparative Analysis of Leading HCI Solutions: VMware vSAN vs. Azure Stack HCI vs. Nutanix. *Journal of Technological Innovations*, 5(2).
- [68] Tymoshchuk, D., & Yatskiv, V. (2025). Using hypervisors to create a cyber polygon. arXiv preprint arXiv:2501.10403.
- [69] Macron, T. (2024). Secure SIP protocol implementation in Asterisk-based VoIP networks.
- [70] Areo, G. (2024). Performance Evaluation of Asterisk VoIP in High-Traffic Environments.
- [71] Selin, D. O., Sferle, A. T., Blaga, T. M., Zinca, D., & Dobrota, V. (2024). WEBRTC AND PJSIP IN ASTERISK: A CLOUD-BASED APPROACH. *Acta Technica Napocensis*, 64(1), 26-35.
- [72] Timilehin, O. (2024). A Robust Security Architecture for Open Source VoIP: Safeguarding Next-Gen Communication.

- [73] Thangam, S., Rooshita, K., Vyshanvi, N. V., Chowdary, N. R. S., & Kumar, K. V. H. (2024, April). Enhancing VoIP Over Protocols Using Codec. In 2024 International Conference on Knowledge Engineering and Communication Systems (ICKECS) (Vol. 1, pp. 1-7). IEEE.
- [74] Nedyalkov, I., Georgiev, G., Gogushev, A., & Duev, V. (2024, November). Windows or Linux-the Most Suitable Platform to Run the GNS3. In 2024 32nd National Conference with International Participation (TELECOM) (pp. 1-5). IEEE.
- [75] Sharma, S. B., & Veena, S. (2024, July). Performance Study of Transport Layer Protocol QUIC against TCP in GNS3 Environment. In 2024 Third International Conference on Electrical, Electronics, Information and Communication Technologies (ICEEICT) (pp. 1-6). IEEE.
- [76] Sarrió Ferrández, M. (2024). GNS3-network simulator for evaluation of network projects using scripts (Doctoral dissertation, Universitat Politècnica de València).
- [77] González-Cortés, J. J., Cantero, D., & Ramírez, M. (2025). Project-Based Learning in Bioprocess Engineering: MATLAB Software as a Tool for Industrial-Scale Bioreactor Design. *Computer Applications in Engineering Education*, 33(1), e22811.
- [78] Osadchiy, A., Kamenev, A., Saharov, V., & Chernyi, S. (2021). Signal processing algorithm based on discrete wavelet transform. *Designs*, 5(3), 41.
- [79] Akhmetkhanov, R.S. Application of wavelet transforms for the analysis of one-, two- and three-dimensional data sets. *Probl. Mech. Eng. Mach. Reliab.* 2013, 5, 112–119. [Google Scholar]
- [80] Smolentsev, N.K. Fundamentals of wavelet theory. In *Wavelets in MATLAB*, 4th ed.; Add. and Reworked: Moscow, Russia, 2014; p. 628. [Google Scholar]
- [81] Ercelebi, E. Electrocardiogram Signals De-Noising Using Lifting-Based Discrete Wavelet Transform. *Comput. Biol. Med.* 2004, 34, 479–493. [Google Scholar] [CrossRef]
- [82] Kotze, B., & Jordaan, G. (2014). Investigation of MATLAB® as Platform in Navigation and Control of an Automatic Guided Vehicle Utilising an Omnivision Sensor. *Sensors*, 14(9), 15669-15686.
- [83] Santhosh, D. D., Balakrishnan, S., Dhivyasshri, C. K., & Prakash, N. (2018). Biometric authentication system using matlab. *International Journal of Engineering and Technology (UAE)*, 7(4.19), 101-103.
- [84] Avinash, P., Venkateswarlu, T., & Anand, D. (2018). A detail study on biometrics with Matlab. *Int J Eng Technol (UAE)*, 7(2.20), 243-249.
- [85] Abdullah, A., Palash, W., Rahman, A., Islam, K., & Alim, A. (2016). Digital image processing analysis using Matlab. *American Journal of Engineering Research (AJER)*, 5(12), 2320-0936.
- [86] Lessani, A., Yunus, A., & Bakar, K. (2017). Comparison of new mathematics teaching methods with traditional method. *People: International Journal of Social Sciences*, 3(2), 1285-1297.
- [87] Alsop, J. (2004). A Comparison of Constructivist and Traditional Instruction in Mathematics. *Educational Research Quarterly*, 28(4), 3-17.
- [88] Mekic, E., Djokic, I., Zejnelagic, S., & Matovic, A. (2016). Constructive approach in teaching of VoIP in line with good laboratory and manufacturing practice. *Computer Applications in Engineering Education*, 24(2), 277-287.
- [89] Pecanin, Emir, Petar Spalevic, Edis Mekic, Srdjan Jovic, and Ivan Milovanovic. "E-learning engineers based on constructive and multidisciplinary approach." *Computer applications in engineering education* 27, no. 6 (2019): 1544-1554.

- [90] Di, C., Zhou, Q., Shen, J., Li, L., Zhou, R., & Lin, J. (2021). Innovation event model for STEM education: A constructivism perspective. *STEM Education*, 1(1), 60-74.
- [91] Sidhu, M. S., Mousakhani, S., Lee, C. K., & Sidhu, K. K. (2024). Educational impact of Metaverse learning environment for engineering mechanics dynamics. *Computer Applications in Engineering Education*, 32(5), e22772.
- [92] Yavuzcan, H. G., & Gür, B. (2020). A Toolkit for Practice-Based Learning of Mechanisms in Industrial Design Education: An Application of a Method Combining Deductive and Inductive Learning. *Design and Technology Education*, 25(3), 47-68.
- [93] Al-Saqqa, S., Sawalha, S., & AbdelNabi, H. (2020). Agile software development: Methodologies and trends. *International Journal of Interactive Mobile Technologies*, 14(11).
- [94] Dybå, T., & Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. *Information and software technology*, 50(9-10), 833-859.
- [95] Williams, L. (2010). Agile software development methodologies and practices. In *Advances in computers* (Vol. 80, pp. 1-44). Elsevier.
- [96] Deshpande, S. A., Deshpande, N. A., Marathe, V. M., & Garje, G. V. (2010). Improving software quality with agile testing. *International Journal of Computer Applications*, 1(22), 68-73.
- [97] Innovative Approach to Teaching Distributed Systems in Education 4.0. Purkovic, Irfan Fetahovic, Edis Mekic, Gokmen Katipoglu, Semih Utku 2022 *Journal International Journal of Engineering Education* 40, 1-16
- [98] Shchetynina, O., Kravchenko, N., Horbatiuk, L., Aliksieieva, H., & Mezhuyev, V. (2022). Trello as a tool for the development of lifelong learning skills of senior students. *Postmodern Openings*, 13(2), 143-167.
- [99] Cosentino, V., Izquierdo, J. L. C., & Cabot, J. (2017). A systematic mapping study of software development with GitHub. *Ieee access*, 5, 7173-7192.
- [100] Abar, S., Theodoropoulos, G. K., Lemarinier, P., & O'Hare, G. M. P. (2017). Agent Based Modelling and Simulation tools: A review of the state-of-art software. *Computer Science Review*, 24, 13–33. <https://doi.org/10.1016/j.cosrev.2017.03.001>
- [101] Shirmard, H., Farahbakhsh, E., Heidari, E., Beiranvand Pour, A., Pradhan, B., Müller, D., & Chandra, R. (2022). A comparative study of convolutional neural networks and conventional machine learning models for lithological mapping using remote sensing data. *Remote Sensing*, 14(4), 819. <https://doi.org/10.3390/rs14040819>
- [102] Ciaburro, G. (2024). *MATLAB for Machine Learning: Unlock the power of deep learning for swift and enhanced results*. Packt Publishing Ltd.
- [103] Dabic, O., Aghajani, E., & Bavota, G. (2021, May). Sampling projects in github for MSR studies. In *2021 IEEE/ACM 18th International Conference on Mining Software Repositories (MSR)* (pp. 560-564). IEEE.
- [104] Tao, W., Zhou, Y., Wang, Y., Zhang, W., Zhang, H., & Cheng, Y. (2025). Magis: Llm-based multi-agent framework for github issue resolution. *Advances in Neural Information Processing Systems*, 37, 51963-51993.
- [105] Wessel, M., Vargovich, J., Gerosa, M. A., & Treude, C. (2023). Github actions: the impact on the pull request process. *Empirical Software Engineering*, 28(6), 131.

- [106] Saroar, S. G., Ahmed, W., Onagh, E., & Nayebi, M. (2024). GitHub marketplace for automation and innovation in software production. *Information and Software Technology*, 175, 107522.
- [107] Свінцицька, О. М., & Пулеко, І. В. (2023). Інтеграція Jira, Bitbucket та Sourcetree в системі управління ІТ-проєктами.
- [108] Kamila, J. S., & Marzuq, M. F. (2024). Asana and Trello: a comparative assessment of project management capabilities. *JOIV: International Journal on Informatics Visualization*, 8(1), 207-212.
- [109] Naik, N., & Jenkins, P. (2019, October). A web based method for managing PRINCE2® projects using trello®. In *2019 International Symposium on Systems Engineering (ISSE)* (pp. 1-3). IEEE.
- [110] Lema, A., & Enciso, L. (2022, September). Trello: a management strategy in project-based learning. In *2022 XII International Conference on Virtual Campus (JICV)* (pp. 1-4). IEEE.
- [111] Connie, C., & Risdianto, E. (2022). MOOCs and Trello based blended learning to increase student involvement. *AL-ISHLAH: Jurnal Pendidikan*, 14(1), 1001-1008.
- [112] Schwaber, K. (1997). Scrum development process. In *Business object design and implementation: OOPSLA '95 workshop proceedings 16 October 1995, Austin, Texas* (pp. 117-134). Springer London.
- [113] Sachdeva, S. (2016). Scrum methodology. *Int. J. Eng. Comput. Sci*, 5(16792), 16792-16800.
- [114] Hron, M., & Obwegeser, N. (2018). Scrum in practice: an overview of Scrum adaptations. In *Hawaii International Conference on System Sciences* (pp. 4496-4505). Curran Associates, Inc..
- [115] Srivastava, A., Bhardwaj, S., & Saraswat, S. (2017, May). SCRUM model for agile methodology. In *2017 International Conference on Computing, Communication and Automation (ICCCA)* (pp. 864-869). IEEE.
- [116] Cervone, H. F. (2011). Understanding agile project management methods using Scrum. *OCLC Systems & Services: International digital library perspectives*, 27(1), 18-22.
- [117] Paasivaara, M., Durasiewicz, S., & Lassenius, C. (2009, July). Using scrum in distributed agile development: A multiple case study. In *2009 Fourth IEEE International Conference on Global Software Engineering* (pp. 195-204). IEEE.

### 13. BIOGRAFIJA AUTORA

Mario Stojanović rođen je 29.04.1984. godine u Jagodini. Osnovne akademske studije završio je na Elektronskom fakultetu na Univerzitetu u Nišu sa prosečnom ocenom 8,11 i stekao zvanje Diplomirani inženjer elektrotehnike za telekomunikacije.

Kandidat 2022. godine upisuje doktorske studije na Fakultetu informacionih tehnologija, Alfa BK Univerziteta u Beogradu, gde je u skladu sa predviđenim planom i programom položio sve ispite sa prosečnom ocenom 9,58.

Na Alfa BK Univerziteta u Beogradu, kandidat je radio kao predavač van radnog odnosa od 2024. godine.

Oblasti njegovog interesovanja su i veštačka inteligencija. O obzirom da ima dugogodišnje radno iskustvo u mesnoj industriji, znanje koje je stekao kako na osnovim, a tako i na doktorskim studijama implementirao je u svoju sferu poslovanja.



Алфа БК Универзитет

## IZJAVA O AUTORSTVU

**1. Potpisani:** Mario Stojanović

**2. Broj upisa:** 2022/5903

**3. Izjavljujem da je doktorska disertacija pod naslovom:**

RAZVOJ SIMULACIONOG SISTEMA ZA VOIP I OBRADU SIGNALA ZASNOVANOG  
NA AI ALGORITMIMA PRIMENOM AGILNOG PRISTUPA RAZVOJU

- Rezultat sopstvenog istraživačkog rada.
- Da predložena disertacija u celini ni u delovima nije bila predložena za dobijanje bilo koje diplome prema studijskim programima drugih visokoškolskih ustanova.
- Da su rezultati korektno navedeni.
- Da nisam kršio autorska prava i koristio intelektualnu svojinu drugih lica.

U Beogradu, \_\_\_\_\_

Potpis doktoranda

\_\_\_\_\_

ALFA BK UNIVERZITET, NOVI BEOGRAD, Palmira Toljatija 3, + 381112674164,

[www.alfa.edu.rs](http://www.alfa.edu.rs), [info@alfa.edu.rs](mailto:info@alfa.edu.rs),

PIB:100421838



Алфа БК Универзитет

**IZJAVA O ISTOVETNOSTI ŠTAMPANE I ELEKTRONSKE VERZIJE  
DOKTORSKOG RADA**

**Ime i prezime:** Mario Stojanović

**Broj upisa:** 2022/5903

**Studijski program:** Informaciono-komunikacione tehnologije

**Naslov rada:** Predikcija uspeha studenata u sistemima za e-učenje primenom tehnika veštačke inteligencije: razvoj metodologije i praktične implikacije

**Mentor:** prof. dr Milan Gligorijević

Izjavljujem:

Da je štampana verzija mog doktorskog rada istovetna elektronskoj verziji koju sam predao za objavljivanje u repozitorijumu na sajtu Alfa BK Univerziteta.

Dozvoljavam da se objave moji lični podaci vezani za dobijanje naučnog zvanja doktora nauka kao što su ime i prezime, godina i mesto rođenja, podaci o stečenim stručnim i akademskim zvanjima, datum odbrane rada i drugi podaci u funkciji transparentnosti postupka sticanja naučnog zvanja.

Ovi lični podaci mogu se objaviti u publikacijama Alfa BK Univerziteta i dostaviti Ministarstvu prosvete, nauke i tehnološkog razvoja, kao i biti dostupni saglasno Zakonu o slobodnom pristupu i informacijama od javnog značaja.

U Beogradu, \_\_\_\_\_

Potpis doktoranda

\_\_\_\_\_

ALFA BK UNIVERZITET, NOVI BEOGRAD, Palmira Toljatija 3, + 381112674164,

[www.alfa.edu.rs](http://www.alfa.edu.rs), [info@alfa.edu.rs](mailto:info@alfa.edu.rs),

PIB:100421838

## IZJAVA O KORIŠĆENJU

Ovlašćujem Alfa BK Univerzitet da u Digitalni repozitorijum Univerziteta unese moju doktorsku disertaciju pod naslovom: „Razvoj simulacionog sistema za VoIP i obradu signala zasnovanog na AI algoritmima primenom agilnog pristupa razvoju”, koja je moje a utorsko delo.

Disertaciju sa svim priložima predao sam u elektronskom formatu pogodnom za trajno arhiviranje.

Moju doktorsku disertaciju pohranjenu u Digitalnom repozitorijumu Univerziteta, dostavljenu repozitorijumu Ministarstva prosvete, nauke i tehnološkog razvoja Republike Srbije i dostupnu u otvorenom pristupu mogu da koriste svi koji poštuju odredbe sadržane u odabranom tipu licence Kreativne zajednice (*Creative Commons*) za koju sam se odlučio.

1. Autorstvo (*CC BY*)
2. **Autorstvo - nekomercijalno (*CC BY-NC*)**
3. Autorstvo - nekomercijalno - bez prerada (*CC BY-NC-ND*)
4. Autorstvo - nekomercijalno - deliti pod istim uslovima (*CC BY-NC-SA*)
5. Autorstvo - bez prerada (*CC BY-ND*)
6. Autorstvo - deliti pod istim uslovima (*CC BY-SA*)

(Molimo da zaokružite samo jednu od šest ponuđenih licenci. Kratak opis licenci je sastavni deo ove izjave).

Potpis autora

U Beogradu, \_\_\_\_\_

\_\_\_\_\_

1. **Autorstvo.** Dozvoljavate umnožavanje, distribuciju i javno saopštavanje dela, i prerade, ako se navede ime autora na način određen od strane autora ili davaoca licence, čak i u komercijalne svrhe. Ovo je najslobodnija od svih licenci.

2. **Autorstvo - nekomercijalno.** Dozvoljav

3. ate umnožavanje, distribuciju i javno saopštavanje dela, i prerade, ako se navede ime autora na način određen od strane autora ili davaoca licence. Ova licenca ne dozvoljava komercijalnu upotrebu dela.

4. **Autorstvo - nekomercijalno - bez prerada.** Dozvoljavate umnožavanje, distribuciju i javno saopštavanje dela, bez promena, preoblikovanja ili upotrebe dela u svom delu, ako se navede ime autora na način određen od strane autora ili davaoca licence. Ova licenca ne dozvoljava komercijalnu upotrebu dela. U odnosu na sve ostale licence, ovom licencom se ograničava najveći obim prava korišćenja dela.

5. **Autorstvo - nekomercijalno - deliti pod istim uslovima.** Dozvoljavate umnožavanje, distribuciju i javno saopštavanje dela, i prerade, ako se navede ime autora na način određen od strane autora ili davaoca licence i ako se prerada distribuira pod istom ili sličnom licencom. Ova licenca ne dozvoljava komercijalnu upotrebu dela i prerada.

6. **Autorstvo - bez prerada.** Dozvoljavate umnožavanje, distribuciju i javno saopštavanje dela, bez promena, preoblikovanja ili upotrebe dela u svom delu, ako se navede ime autora na način određen od strane autora ili davaoca licence. Ova licenca dozvoljava komercijalnu upotrebu dela.

7. **Autorstvo - deliti pod istim uslovima.** Dozvoljavate umnožavanje, distribuciju i javno saopštavanje dela, i prerade, ako se navede ime autora na način određen od strane autora ili davaoca licence i ako se prerada distribuira pod istom ili sličnom licencom. Ova licenca dozvoljava komercijalnu upotrebu dela i prerada. Slična je softverskim licencama, odnosno licencama otvorenog koda.



Алфа БК Универзитет

## **IZJAVA MENTORA O PROCENI ORIGINALNOSTI I SAGLASNOSTI ZA PREDAJU URAĐENE DOKTORSKE DISERTACIJE**

Ovim izjavljujem da sam nakon pregledanog rukopisa doktorske disertacije saglasan da kandidat Mario Stojanović može da preda Službi za poslediplomske studije Univerziteta urađenu doktorsku disertaciju pod nazivom: „Razvoj simulacionog sistema za VoIP i obradu signala zasnovanog na AI algoritmima primenom agilnog pristupa razvoju”, radi organizacije njene ocene i odbrane i da ista sadrži originalan naučni doprinos koji se ogleda u: predloženoj metodologiji za predikciju uspeha studenata u sistemima za e-učenje i definisanim smernicama za pretprocesiranje podataka, koje donose bolje razumevanje važnosti tog procesa u svrhu poboljšanja tačnosti predikcije.

U Beogradu, \_\_\_\_\_

\_\_\_\_\_  
(potpis mentora)