



**Slovenská technická univerzita v Bratislave**  
**Fakulta informatiky a informačných technológií**  
**Ilkovičova 2, 842 16 Bratislava 4**

Predmet

**– Mobilné technológie a aplikácie –**

**- Dokumentácia -**

**Sip Proxy (Telefónna ústredňa)**

Ak. Rok : 2021/2022, letný semester

**Cvičiaci:**

**Ing. Miroslav Bahleda, PhD.**

**Študent:**

Martin Rudolf 97029



Bratislava, 2022.

## Obsah:

<b>1</b>	<b>Implementácia a použité knižnice .....</b>	<b>2</b>
<b>2</b>	<b>Implementované funkcionality a testovanie .....</b>	<b>2</b>
	a. Registrácia účastníka: .....	2
	b. Vytocenie hovoru, zvonenie na druhej strane, prijatie hovoru, fungujúci hlasový hovor a ukončenie hovoru (prijateho aj nepriateho) .....	2
	c. Realizovanie videohovoru .....	3
	d. Úprava SIP stavových kódov .....	3
	e. Testovanie.....	4
<b>3</b>	<b>User guide .....</b>	<b>4</b>
<b>4</b>	<b>Link na repozitár .....</b>	<b>4</b>

## 1 Implementácia a použité knižnice

Na implementáciu SIP Proxy, využívame programovací jazyk Python verziu 2.7.18. Pri implementácii nám poslúžila SIP knižnica dostupná na tomto [linku](#) knižnica využíva balíky ako SocketServer, string, time, re, logging

## 2 Implementované funkcionality a testovanie

### a. Registrácia účastníka:

Túto funkcionality poskytuje samotná knižnica ktorá implementuje triedu UDPHandler, ktorá disponuje metódou processRegister() ktorá ma na starosti túto funkcionality

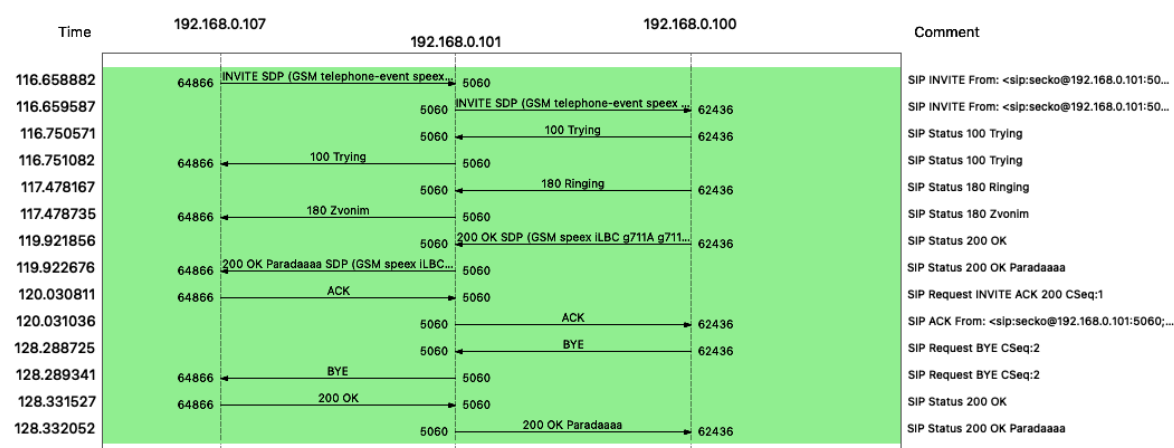
315	15.494525	192.168.0.107	192.168.0.101	SIP	637 Request: REGISTER sip:192.168.0.101:5060;transport=UDP (1 binding)
316	15.495499	192.168.0.101	192.168.0.107	SIP	638 Status: 200 OK (REGISTER) (1 binding)
324	16.120675	192.168.0.107	192.168.0.101	SIP	637 Request: REGISTER sip:192.168.0.101:5060;transport=UDP (1 binding)
325	16.121370	192.168.0.101	192.168.0.107	SIP	638 Status: 200 OK (REGISTER) (1 binding)
382	21.707523	192.168.0.100	192.168.0.101	SIP	640 Request: REGISTER sip:192.168.0.101:5060;transport=UDP (1 binding)
383	21.708238	192.168.0.101	192.168.0.100	SIP	641 Status: 200 OK (REGISTER) (1 binding)
410	25.807924	192.168.0.100	192.168.0.101	SIP	640 Request: REGISTER sip:192.168.0.101:5060;transport=UDP (1 binding)
411	25.808666	192.168.0.101	192.168.0.100	SIP	641 Status: 200 OK (REGISTER) (1 binding)

Obrázok 1 Registracia dvoch zariadení

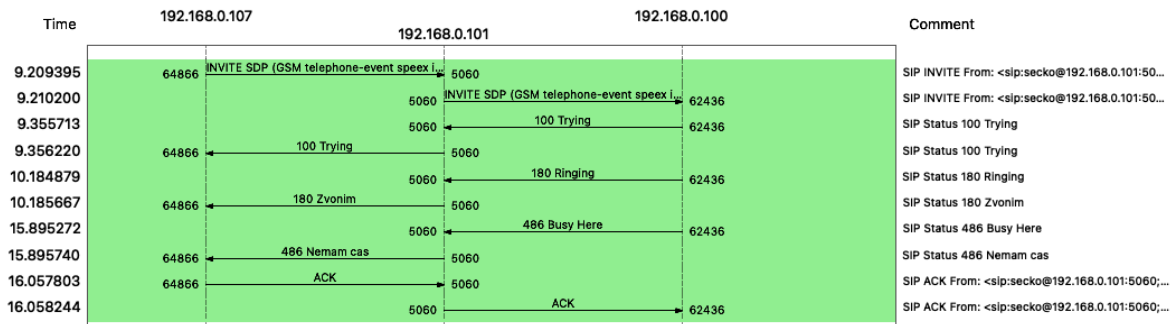
### b. Vytočenie hovoru, zvonenie na druhej strane, prijatie hovoru, fungujúci hlasový hovor a ukončenie hovoru (prijatého aj neprijatého)

Tak ako predchádzajúcu funkcionality, aj tieto poskytuje samotná knižnica.

Splnenie týchto funkcionalít majú v réžií metódy handle() a processRequest() ktorá vyhodnotí požiadavku a spustí patričnú metódu podľa požiadavky. V princípe knižnica hlavne pracuje s metódami ako handle() ktorá kontroluje prichádzajúce požiadavky na proxy a vyhodnocuje ich pomocou processRequest() ktorá následne napríklad pri vytočení hovoru zavolá metódu processInvite().



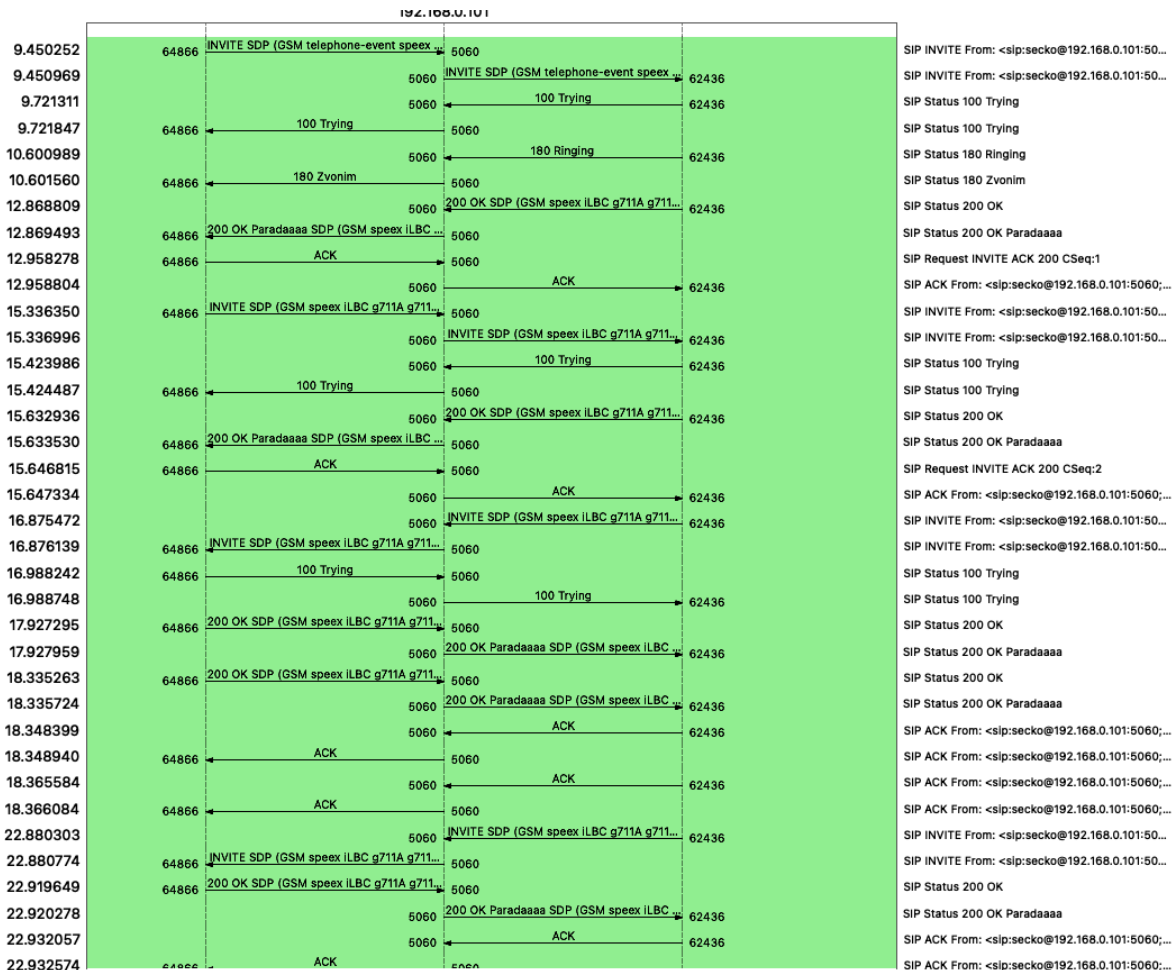
Obrázok 2 Flow sequence pre prijatý hovor a následne zrušený



Obrázok 3 Flow sequence pre ukončenie neprijatého hovoru

### c. Realizovanie video hovoru

Funkcionalitu video hovoru poskytuje priamo knižnica



Obrázok 4 Flow sequence pre video hovor

### d. Úprava SIP stavových kódov

Na implementáciu tejto funkcionality musíme prepísať metódu processCode() triedy UDPhandlet implementovanú v používanej knižnici. Táto metóda spracováva text ktorý sa odosiela ako stavový kód. Zmenili sme kód 180 „Ringing“ na „Zvoním“, kód 200 „OK“ na „OK Paradaaaa“ a kód 486 „Busy Here“ na „Nemam cas“. Vid'. Obrázky 2, 3, 4.

### e. Testovanie

Na testovanie potrebujeme 2 zariadenia medzi ktorými bude prebiehať hovor. Na tieto zariadenia je potrebné nainštalovať SIP klienta pomocou ktorého budeme komunikovať so SIP Proxy. Používame teda ako prvé zariadenie Iphone X, druhé zariadenie Iphone SE a Zoiper ako klienta. V ňom si vytvoríme nový SIP account a pri konfigurácii nastavíme domain ako IP\_zariadenia:port, v našom prípade to je 192.168.0.101:5060. Po vytvorení nového sip účtu sme testovali rôzne scenáre na testovanie vyššie spomenutých funkcionalít. Na overenie správnosti fungovania SIP Proxy sme použili aj nástroj wireshark na zachytávanie komunikácie na ktorej sa podieľal SIP protokol.

## 3 User guide

Na spustenie programu potrebujeme dve zdrojové súbory .py. A teda main.py súbor ktorý importuje vyššie spomínanú knižnicu na realizáciu SIP Proxy, a knižničný súbor sipfullproxy.py.

Na spustenie tejto implementácie si otvoríme terminál a pomocou príkazu cd sa dostaneme do adresára v ktorom máme náš projekt (súbory main.py a sipfullproxy.py), následne spustíme príkaz `python2 main.py IP_zariadenia`. Po odpálení tohto príkazu sa spustí SIP Proxy server na zariadení ktoré definuje IP adresa zadaná ako argument.

Pre overenie funkcionalít potrebujeme nainštalovať na zariadenie, ktoré budeme využívať na testovanie, sip klienta a nakonfigurovať nový sip account. Tento proces sme si opísali v predchádzajúcej kapitole Testovanie.

## 4 Link na repozitar

Na tomto [odkaze](#) je voľne dostupná realizácia tohto zadania aj s traceami teda pcap súbormi ktoré zaznamenávajú prebiehajúcu komunikáciu pomocou SIP protokolu. V súbore celyHovor.pcap mapuje registráciu zariadení a hovor (zvonenie, zdvihnutie a zloženie) medzi dvoma zariadeniami. Súbor busyHovor.pcap nesie informáciu o odmietnutom hovore (zvonenie a zloženie) a videoHovor.pcap traceuje video hovor medzi dvoma zariadeniami.