

FAKULTA INFORMATIKY  
A INFORMAČNÝCH TECHNOLOGIÍ  
SLOVENSKÁ TECHNICKÁ UNIVERZITA

Predmet: Mobilné technológie a aplikácie

Zadanie č. 1

**SIP Proxy**

Autor: Adrián Kabáč

Cvičiaci: Ing. Matej Janeba

Cvičenie: Streda 16:00

# Implementácia riešenia

## Použité knižnice

Pre vypracovania zadania som použil programovací jazyk python a pre proxy funkcionality knižnicu sipfullproxy <https://github.com/tirfil/PySipFullProxy>. Po stiahnutí knižnice z githubu som importoval súbor sipfullproxy.py do svojho programu a použil jeho funkcionality. Táto knižnica však bola napísaná v python2, preto bolo potrebné uskutočniť zopár zmien aby fungovala aj pre python3. Táto knižnica používa knižnicu re pre regulárne výrazy a logging na logovanie, ktoré som taktiež použil pri implementácii doplnkových funkcionalít. Ďalšia knižnica použitá v zadaní je socketserver, ktorú používam na spustenie UDPServera a zároveň v sipfullproxy je trieda UDPHandler ktorá dedí od BaseRequestHandler triedy zo socketserver knižnice.

## Vykonané zmeny pre fungovanie knižnice v python3

Knižnica obsahuje dictionary registrar na ukladanie registrovaných klientov. Na viacerých miestach v kóde sa nachádzala podmienka na zistenie, či sa určitý kľúč nachádza v slovníku zapísaná ako `if registrar.has_key(key)`. Aby rovnaká podmienka fungovala pre python3 zmenil som ju na `if key in registrar` na všetkých miestach kde sa v kóde používala.

Ďalšia zmena predstavovala encoding a decoding stringov posielaných cez socket do utf-8. Pôvodná implementácia obsahovala posielanie dát cez socket vo formáte stringu, avšak v python3 s tým bol problém. Preto bolo potrebné pred odoslaním dát cez socket encodovať string do utf-8 pomocou `encode()` funkcie, a po prijatí dát ich decodovať pomocou `decode()` funkcie.

Po vykonaní týchto zmien bola knižnica importovaná do môjho main.py súboru, kde po nastavení ip adresy na ktorej bude server bežať spustíme UDP server ktorý používa UDPHandler triedu z knižnice sipfullproxy.

## Doplnková funkcionality

### Videohovor, konferenčný hovor, presmerovanie hovoru

Implementácia v sipfullproxy podporuje vyššie spomenuté funkcionality, teda videohovor, konferenčný hovor aj presmerovanie hovoru. Na túto funkcionality je však potrebný klient ktorý tieto veci podporuje.

Na testovanie tejto funkcionality bol použitý klient linphone (<https://www.linphone.org>), keďže zoiper vo free verzii tieto funkcionality neumožňoval.

### Logovanie hovorov

Pôvodná implementácia obsahovala logovanie väčšiny prenášaných správ. Pre účely zadania bolo potrebné logovať iba záznam hovorov – kto kedy komu volal, kedy bol hovor prijatý a kedy ukončený. Na ukladanie denníka hovorov som taktiež použil logovanie, pričom logujem iba požadované veci a všetky pôvodné logovacie správy som zakomentoval. Ukladanie správ prebieha do súboru proxy.log

Každý záznam v logu obsahuje čas kedy bol log vytvorený, a Call-ID pre jednoznačné určenie ktoré záznamy patria ku ktorému hovoru. Metódu na inicializáciu logovania, ktorá obsahuje formát výpisov a výstupný súbor som pridal do sipfullproxy knižnice a v maine túto metódu volám pred spustením serveru. Nasledujúce riadky opisujú kedy vykonávam logovanie do denníka hovorov.

**Vytočenie hovoru** – Príchod INVITE requestu na proxy, ukladám kto komu kedy volal a Call-ID

**Zdvihnutie hovoru** – Príchod status správy 200 Ok na INVITE request

**Ukončenie hovoru** – Príchod BYE requestu na proxy ak hovor prebieha, ďalší spôsob ukončenia predstavuje správu 603 decline pri odmietnutí hovoru alebo CANCEL request pri jeho zrušení pred zdvihnutím.

```
2022-02-25 18:13:34,180:INFO:Call-ID:nf5nhH2w4c Hovor od proxy@192.168.43.99 pre kabadmobil@192.168.43.99
2022-02-25 18:13:36,871:INFO:Call-ID:nf5nhH2w4c Pouzivatel kabadmobil@192.168.43.99 prijal hovor od pouzivatela proxy@192.168.43.99
2022-02-25 18:13:40,181:INFO:Call-ID:nf5nhH2w4c Pouzivatel proxy@192.168.43.99 ukoncil hovor s pouzivatelom kabadmobil@192.168.43.99
2022-02-25 18:13:43,153:INFO:Call-ID:P-oEGSx~0~ Hovor od proxy@192.168.43.99 pre kabadmobil@192.168.43.99
2022-02-25 18:13:44,908:INFO:Call-ID:P-oEGSx~0~ Pouzivatel kabadmobil@192.168.43.99 prijal hovor od pouzivatela proxy@192.168.43.99
2022-02-25 18:13:46,759:INFO:Call-ID:P-oEGSx~0~ Pouzivatel kabadmobil@192.168.43.99 ukoncil hovor s pouzivatelom proxy@192.168.43.99
2022-02-25 18:13:54,624:INFO:Call-ID:~wpX2NVgqo Hovor od kabadmobil@192.168.43.99 pre proxy@192.168.43.99
2022-02-25 18:13:57,320:INFO:Call-ID:~wpX2NVgqo Pouzivatel proxy@192.168.43.99 prijal hovor od pouzivatela kabadmobil@192.168.43.99
2022-02-25 18:14:00,429:INFO:Call-ID:~wpX2NVgqo Pouzivatel proxy@192.168.43.99 ukoncil hovor s pouzivatelom kabadmobil@192.168.43.99
2022-02-25 18:14:07,950:INFO:Call-ID:jZz9PPrk0B Hovor od kabadmobil@192.168.43.99 pre proxy@192.168.43.99
2022-02-25 18:14:10,884:INFO:Call-ID:jZz9PPrk0B Pouzivatel proxy@192.168.43.99 prijal hovor od pouzivatela kabadmobil@192.168.43.99
2022-02-25 18:14:13,384:INFO:Call-ID:jZz9PPrk0B Pouzivatel kabadmobil@192.168.43.99 ukoncil hovor s pouzivatelom proxy@192.168.43.99
2022-02-25 18:15:10,935:INFO:Call-ID:EnNyfyDJSv Hovor od proxy@192.168.43.99 pre kabadmobil@192.168.43.99
2022-02-25 18:15:14,859:INFO:Call-ID:8gQA1-8m0o Hovor od proxy@192.168.43.99 pre kabadmobil@192.168.43.99
2022-02-25 18:15:16,769:INFO:Call-ID:8gQA1-8m0o Pouzivatel proxy@192.168.43.99 zrusil hovor s pouzivatelom kabadmobil@192.168.43.99
2022-02-25 18:17:39,807:INFO:Call-ID:0fNqblTznd Hovor od proxy@192.168.43.99 pre kabadmobil@192.168.43.99
2022-02-25 18:17:41,384:INFO:Call-ID:0fNqblTznd Pouzivatel kabadmobil@192.168.43.99 odmietol hovor od pouzivatela proxy@192.168.43.99
```

Obr.1: Ukážka výpisu z denníku hovorov.

## Zmena hlášok pri stavových kódach

Ak je na proxy server poslaná správa ktorá obsahuje status code, je spracovaná metódou processCode(). V tejto triede sa nachádza premenná self.data a na jej prvej pozícii sa nachádza stavový kód aj status hláška, ktorú správa obsahuje. Pre zmenenie hlášky som použil nájdenie stringu pomocou regex výrazu a jeho nahradenie novým výrazom. Napríklad pre zmenenie status kódu 486 Busy Here na 486 Obsadené som použil nasledovný kód:

```
# replace Busy Here status message
if rx_busy_here.search(self.data[0]):
    self.data[0] = rx_busy_here.sub("Obsadené", self.data[0])
```

Obr.2: Nahradenie stavového kódu v processCode metóde

## Popis pcap súborov

V priečinku pcap\_subory sa nachádzajú odchytené pcap súbory pre jednotlivé scenáre.

- Registrácia účastníka (bez nutnosti autentifikácie) – mtaa\_register.pcapng
- Vytočenie hovoru a zvonenie na druhej strane – mtaa\_ringing.pcapng
- Prijatie hovoru druhou stranou, fungujúci hlasový hovor – mtaa\_call.pcapng
- Ukončenie hlasového hovoru (prijatého aj neprijatého) – mtaa\_call\_cancel.pcapng
- Možnosť zrealizovať konferenčný hovor (aspoň 3 účastníci) – mtaa\_conference\_call.pcapng
- Možnosť presmerovať hovor – mtaa\_call\_redirect.pcapng
- Možnosť realizovať videohovor – mtaa\_video\_call.pcapng
- Logovanie “denníka hovorov” – kto kedy komu volal, kedy bol ktorý hovor prijatý, kedy bol ktorý hovor ukončený, do ľubovoľného textového súboru v ľubovoľnom formáte – denník hovorov sa nachádza v súbore proxy.log

- Úprava SIP stavových kódov v zdrojovom kóde proxy, napr. "486 Busy Here" zmeníte na "486 Obsadené" – mtaa\_custom\_status.pcapng

## **Používateľská príručka**

Pre spustenie ústredne je potrebné spustiť main.py program pričom ako argument použijeme ip adresu nášho zariadenia, kde bude server bežať. Pre spustenie je potrebný nainštalovaný jazyk python3. Ak sa nachádzame v rovnakom priečinku kde sa nachádza main.py skript, program spustíme ako: „`py main.py ipaddress`“, pričom parameter ipaddress je ip adresa nášho zariadenia. Túto adresu môžeme zistiť napríklad príkazom ipconfig, kde nájdeme ip adresu na ktorej sme pripojený. Po spustení programu sa spustí SIP proxy server a môžeme sa naň pripojiť z klienta. Na klientovi je potrebné zadať ip adresu na ktorej beží server do časti domain alebo proxy ip.

Zdrojový kód ku zadaniu ako aj pcap súbory sa nachádza na <https://github.com/kabadrian/MTAA-SIP>.