

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE

Fakulta informatiky a informačných technológií

Ilkovičova 2, 842 16 Bratislava 4

## **MTAA ZADANIE 1 – SIP Proxy**

Meno a priezvisko: Marko Stahovec

Dátum vypracovania: 24.2.2022

Cvičiaci: Ing. Marek Galinski, PhD.

<b>Zadanie</b>	<b>3</b>
<b>Teória</b>	<b>3</b>
<b>Knižnica a vlastné funkcionality</b>	<b>5</b>
<b>Záver</b>	<b>7</b>
<b>Dôležité linky</b>	<b>8</b>

# 1. Zadanie

Naším zadaním bola implementácia **SIP proxy serveru** na základe existujúcej knižnice. Na svoju implementáciu som použil knižnicu v programovacom jazyku Python, ktorej link je uvedený nižšie v kapitole 3.

## 2. Teória

SIP je skratka pre **Session Initiation Protocol**, čo je vo svojej podstate signalizačný protokol, ktorý definuje pravidlá pri nadväzovaní, udržiavaní a ukončovaní relácií. Pokrýva potreby pri funkčnosti komunikačných relácií, pod ktoré patrí prenos hlasového hovoru, videohovoru či bežných správ.

Na druhej strane je **proxy server**, ktorý predstavuje zariadenie, ktoré slúži ako **sprostredkovateľ** medzi dvoma klientmi alebo medzi klientom a serverom. Jeho fungovanie spočíva v preposielaní správ medzi dvoma koncovými bodmi. Pri komunikácii môže slúžiť aj ako ochranné zariadenie, keďže maskuje IP adresu na strane klienta.

SIP má v sebe zahrnutých 5 základných princípov pre sieťové komunikácie, a nimi sú [1]:

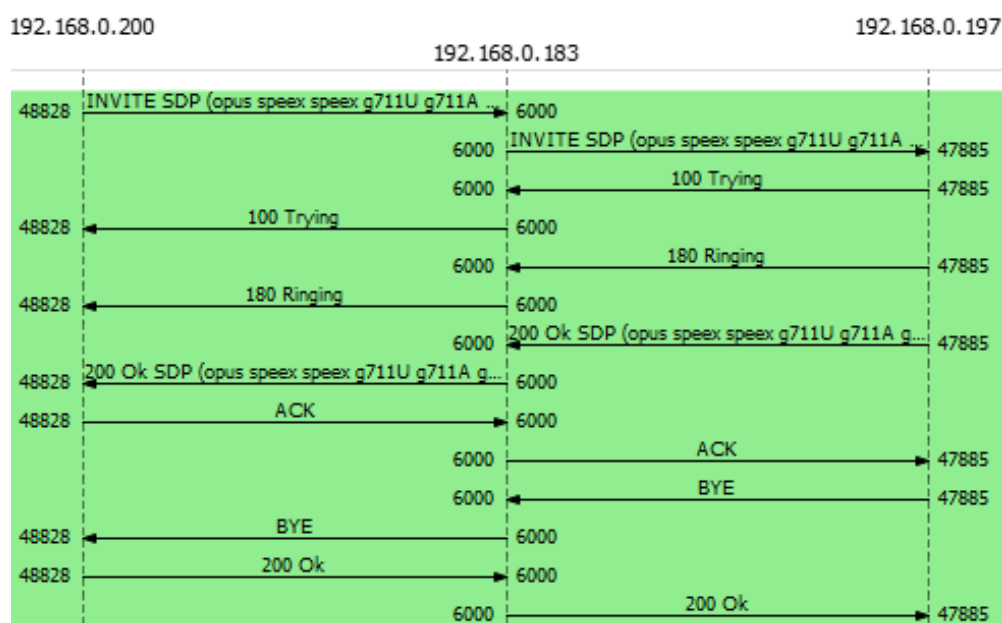
- poloha používateľa:
- dostupnosť používateľa
- schopnosti používateľského zariadenia:
- nastavenie relácie
- manažovanie relácie

Adresy SIP klientov majú rovnakú formu ako e-mailové adresy, napr:

- sip:[user@stuba.sk](mailto:user@stuba.sk)
- sip:user@177.77.7.0

SIP používa niekoľko **metód**, ktorými signalizuje zámer danej správy. Ide o napr. **INVITE, ACK, REGISTER, BYE, CANCEL...** INVITE slúži ako pozvánka do relácie pre iného klienta. ACK predstavuje potvrdenie požiadavky pre druhú stranu, napr. potvrdenie správy INVITE. REGISTER metóda je používaná na zaregistrovanie dostupnosti na SIP serveri. Metódy BYE a CANCEL oznamujú druhej strane stav o ukončení hovoru, v druhom prípade o predčasnom ukončení.

Na dodatkovú signalizáciu existujú v SIP protokole aj **kódy správ**, ktoré slúžia ako odpovede na prvotné požiadavky. Ide napr. o **180 Ringing, 100 Trying** či **200 Ok**, ktorých funkcia je samovysvetľujúca a je zobrazená aj v nasledujúcom obrázku, ktorý znázorňuje učebnicovú výmenu SIP správ medzi dvoma klientmi a proxy serverom.



Po stranách možno pozorovať dvoch klientov a výmenu SIP správ medzi nimi. Pred prvou správou INVITE prebehne ešte aj registrácia oboch účastníkov konverzácie. Ich REGISTER správy sú potvrdené 200 OK odpoveďou zo strany servera.

Ako možno pozorovať na obrázku, **klient s IP adresou 192.168.0.200 inicializoval reláciu**. Nasledovali informačné správy o pokuse nadviazaní spojenia a informáciou o zvonení cieľového zariadenia. Po prijatí hovoru odosiela klient sprava oznam o prijatí hovoru správou 200 Ok, ktorá je automaticky potvrdená klientom zľava SIP

metódou ACK. Následne prebieha výmeny hlasových správ, ktoré sú nasekané na 20ms úseky posielané protokolom RTP. Po ukončení hovoru je zaslaná správa BYE, ktorá je opäť potvrdená druhou stranou. Komunikácia skončila.

Videhovory a konferenčné hovory majú podobnú štruktúru, no ich inicializácia a priebeh je o niečo náročnejší. Pcap súbory z týchto hovorov sú priložené v git repozitári na linku:

<https://github.com/MarkoStahovec/MTAA-1>

### 3. Knižnica a vlastné funkcionality

Knižnicu, ktorú som využil na implementáciu SIP Proxy, som našiel na linku:

<https://github.com/tirfil/PySipFullProxy>

V tejto knižnici bolo vykonaných **niekoľko úprav**. Prvou z nich bolo prepísanie kódu do aktuálnejšej verzie Pythonu, keďže niektoré časti kódu boli zastaralé a prakticky nefunkčné. Išlo napr. o funkciu `has_key()`, ktorá v Python 3.9 nie je podporovaná. Ďalšími zmenami v kóde boli zmeny pri enkódovaní a dekodovaní stringov a bajtov, keďže v aktuálnej verzii Pythonu stringy nemožno posielat' po sieti.

Z tých “nepotrebných” zmien bolo uskutočnené odstránenie, resp. zakomentovanie častí kódu, ktoré obsahovali pôvodnú formu logovania. Tá bola nahradená mojou vlastnou implementáciou.

Záverečnou úpravou existujúcej knižnice bolo napísanie vlastnej main funkcie v separátnom súbore. Knižnica bola následne importovaná ako modul.

#### Ďalšie funkcionality

Všetky povinné funkcionality zo zadania boli implementované a ich korektnosť možno pozorovať na vzorovom obrázku komunikácie ako aj na priložených .pcap súboroch.

Zo zoznamu **doplňkových častí zadania** boli splnené taktiež **všetky body**, ku ktorým sú priložené aj .pcap súbory. Chcem poznamenať, že väčšina týchto .pcap súborov bola vytvorená nie na finálnom kóde, keďže som vedel, že tretie zariadenie na konferenčný hovor nebudem mať dostupné počas písania dokumentácie.

**Konferenčný hovor** možno nájsť pod .pcap súborom 3-call. Tento typ hovoru si nevyžadoval nijaké ďalšie špeciálne zmeny v kóde. Jediným rozdielom od hovoru medzi dvoma účastníkmi je ten, že priebeha aj výmena správ s metódou SUBSCRIBE, ktorou účastníci registrujú svoj záujem o prijímanie informácií o prístupe druhej strany.

**Ďalšie scenáre zo zadania** sa líšia od obvyčajného hovoru len minimálne, napr. videohovor zasiela pri zapnutí kamery novú INVITE správu, ktorá je potvrdená druhou stranou, alebo presmerovanie hovoru využíva metódu REFER na presmerovanie tretej strane apod.

**Logovanie denníka hovorov** bolo implementované bežným zápisom do .txt súboru. Prvým krokom k dosiahnutiu správneho zápisu je odchytenie unikátnych klientov pri posielaní metódy INVITE. Unikátnosť zabezpečí to, že pri napr. uskutočňovaní videohovoru nebudú pridaní tí istí používatelia do zoznamu, keďže aj v takej situácii sa posiela správa s metódou INVITE. Následne, po prijatí hovoru, kedy prebehnú správy 100 Trying, 180 Ringing a 200 Ok, sa zapíše prvá časť hovoru do súboru, ktorá hovorí o dátume, účastníkoch v danom hovore a čase, kedy bol hovor prijatý. Po zrušení hovoru bude do .txt súboru doplnený aj čas, kedy a kým bol hovor zrušený.

Vzor logu z jedného hovoru medzi dvoma používateľmi:

```
----- HOVOR -----  
DATUM: Mon, 21 Feb 2022  
OD: 999@192.168.43.212  
PRE: 444@192.168.43.212,  
  
PRIJATY 0: 18:45:03  
ZRUSENY 0: 18:45:06 , 999@192.168.43.212
```

**Úprava SIP kódov** ako doplnková funkcionálna bola splnená nasledujúcou vzorkou kódu:

```
if rx_ringing.search(data[0]):  
    data[0] = data[0].decode().replace("Ringing", "Zvoni").encode()  
if rx_trying.search(data[0]):  
    data[0] = data[0].decode().replace("Trying", "Skusa nadviazať spojenie").encode()
```

Tento kus kódu odchytkáva na strane proxy správy, ktoré v hlavičke obsahujú kód pre 180 Ringing či 100 Trying. Ich obsah je následne prepísaný slovenským ekvivalentom.

Vzor pre upravené SIP kódy sa nachádza v .pcap súbore s názvom custom-SIP-codes.pcap.

```
SIP      695 Request: REGISTER sip:192.168.0.183:6000 (1 binding) |  
SIP      709 Status: 200 OK (REGISTER) (1 binding) |  
SIP/SDP 1232 Request: INVITE sip:999@192.168.0.183:6000 |  
SIP      278 Status: 100 Skusa nadviazať spojenie |  
SIP      449 Status: 180 Zvoni |  
SIP/SDP 1235 Status: 200 OK (INVITE) |  
SIP/SDP 1235 Status: 200 OK (INVITE) |  
SIP      425 Request: ACK sip:999@192.168.0.197:46313;transport=udp |  
SIP      425 Request: ACK sip:999@192.168.0.197:46313;transport=udp |  
SIP      425 Request: BYE sip:999@192.168.0.197:46313;transport=udp |  
SIP      443 Status: 200 OK (BYE) |
```

## 4. Záver

Toto zadanie bolo primerane náročné aj keď miestami krkolomné. Tak či onak predpokladám, že splnilo účel, keďže som sa naučil veľa potrebných informácií na tému sieťové technológie a telefónne ústredne.

## 5. Dôležité linky

Knižnica:

- <https://github.com/tirfil/PySipFullProxy>

Github:

- <https://github.com/MarkoStahovec/MTAA-1>