

Martin Hric
ID: 111696

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA INFORMATIKY A INFORMAČNÝCH TECHNOLOGIÍ

MTAA

SIP Proxy

Cvičiaci: Ing. Miroslav Bahleda, PhD.

Ak. Rok: 2021/22

OBSAH

ZADANIE	3
IMPLEMENTÁCIA	4
SPLNENÉ SCENÁRE	5
TESTOVANIE.....	7

ZADANIE

Hlavná myšlienka zadania:

Na vašom počítači (alebo virtuálnom počítači) sprevádzajte SIP Proxy, ktorá umožní prepájanie a realizáciu hovorov medzi štandardnými SIP klientami.

Doplňujúce informácie k zadaniu:

Na implementáciu vašej SIP Proxy si môžete zvoliť akýkoľvek programovací jazyk a použiť akúkoľvek SIP knižnicu, ktorá pre daný programovací jazyk existuje. Vo výsledku však musíte spúšťať "váš kód", v ktorom sú zakomponované knižnice, ktoré poskytujú funkcionality SIP Proxy. To znamená, že nemôžete zobrať existujúcu SIP Proxy ako napr. Asterisk, kde len skompilujete alebo priamo spustíte cudziu binárku... Hovor musí byť realizovaný medzi dvoma fyzickými zariadeniami v rámci LAN siete.

Rozsah povinných funkcionalít:

- Registrácia účastníka (bez nutnosti autentifikácie)
- Vytocenie hovoru a zvonenie na druhej strane
- Prijatie hovoru druhou stranou, fungujúci hlasový hovor
- Ukončenie hlasového hovoru (prijatého aj neprijatého)

Ak sú splnené všetky tieto podmienky, študent získava 5 bodov, ktoré sú minimom na absolvovanie tohoto zadania.

Doplňkové funkcionality (ktoré môžete, ale nemusíte urobiť):

- Možnosť zrealizovať konferenčný hovor (aspoň 3 účastníci)
- Možnosť presmerovať hovor
- Možnosť realizovať videohovor
- Logovanie "denníka hovorov" – kto kedy komu volal, kedy bol ktorý hovor prijatý, kedy bol ktorý hovor ukončený, do ľubovoľného textového súboru v ľubovoľnom formáte
- Úprava SIP stavových kódov z zdrojovom kóde proxy, napr. "486 Busy Here" zmeníte na "486 Obsadené"

IMPLEMENTÁCIA

Zadanie som sa rozhodol implementovať v jazyku Python3, kvôli jednoduchosti a pohodlnosti pri výbere možných knižníc.

Zdroj môjho zadania: <https://github.com/tirfil/PySipFullProxy/blob/master/sipfullproxy.py> , je to implementované v jazyku Python2, čiže bolo tam dosť prepisovačiek a opravovaní bugov, ktoré ešte boli v starom Pythone.

Moja implementácia s kódmi a .pcap súbormi je: <https://github.com/martin-hric/MTAA-SIP-Proxy> , momentálne ešte to nechávam na Private, kvôli tomu aby nejakí študenti to nemohli okopírovať (snáď po termíne odovzdania nezapudnem to dať na Public :))

Ako klienta som sa rozhodol použiť Zoiper5 - <https://www.zoiper.com> , skúšal som prv Linphone ale ten mi bohužiaľ na mojom PC nefungoval.

Použité knižnice:

- sipfullproxy** – implementácia z GitHubu
- socket** – práca so socketmi, ip...
- socketserver** – na spustenie UDPServera
- re** - regexy
- time** – určovanie lokálneho času aj pre Denník hovorov
- logging** – v pôvodnej implementácii tam si autor kódu vypisoval logy

Testovanie a kontrolu SIP packetov som uskutočnil vo Wiresharku - <https://www.wireshark.org>.

USER GUIDE:

- Program spúšťam v main.py, ten mi spustí server, do konzole vypíše na akej IP adrese server beží
- Server slúži donekonečna, čiže nakonci ho len surovo cez Ctrl+C vypnem
- Cez Klienta ako doménu zvolím IP a port na akom server beží, port je :5060 default
- Zbehne mi registrácia a v klientovi uvidím, že som online a môžem volať
- Na konci program vygeneruje logs.txt , kde je výpis denníka hovorov ako aj proxy.log, to autor kódu si tam spravil vlastný výpis, ktorý som nechcel meniť.

Martin Hric
ID: 111696

SPLNENÉ SCENÁRE

Splnil som všetky povinné funkcionality, čím sú:

- Registrácia účastníka (bez nutnosti autentifikácie)
- Vytočenie hovoru a zvonenie na druhej strane
- Prijatie hovoru druhou stranou, fungujúci hlasový hovor
- Ukončenie hlasového hovoru (prijatého aj neprijatého)

Z doplnkových funkcionalít som implementoval **“Denník Hovorov”**, ktorý je uložený v súbore logs.txt. Takto cca vyzerá výstup:

```
01 Mar 2022 13:21:24 Registracia noveho uzivatela[martinhric@147.175.183.39:5060]
01 Mar 2022 13:21:38 Registracia noveho uzivatela[martin@147.175.183.39:5060]
01 Mar 2022 13:21:46 Registracia noveho uzivatela[martin@147.175.183.39:5060]
01 Mar 2022 13:21:57 Registracia noveho uzivatela[martinhric@147.175.183.39:5060]
01 Mar 2022 13:22:09 [Hovor prijaty] => Call-ID: 87U76tbJ-zaED4Pr11RPjg.. || From: <sip:martinhric@147.175.183.39:5060;transport=UDP>;tag=2b420c74 => To: <sip:martin@147.175.183.39:5060>;tag
01 Mar 2022 13:22:12 [Hovor ukonceny] => Call-ID: 87U76tbJ-zaED4Pr11RPjg.. || From: <sip:martin@147.175.183.39:5060>;tag=9a5bf905 => To: <sip:martinhric@147.175.183.39:5060;transport=UDP>;ta
01 Mar 2022 13:22:16 Registracia noveho uzivatela[martin@147.175.183.39:5060]
01 Mar 2022 13:22:18 Registracia noveho uzivatela[martinhric@147.175.183.39:5060]
01 Mar 2022 13:24:54 Registracia noveho uzivatela[martin@147.175.183.39:5060]
01 Mar 2022 13:24:55 Registracia noveho uzivatela[martinhric@147.175.183.39:5060]
01 Mar 2022 13:25:06 [Hovor prijaty] => Call-ID: EOudzmeJA-U_KHJawRbA7w.. || From: <sip:martinhric@147.175.183.39:5060;transport=UDP>;tag=10697651 => To: <sip:martin@147.175.183.39:5060>;ta
01 Mar 2022 13:25:06 [Ziadost zrusena] => User-Agent: Zoiper v2.10.17.3 || CSeq: 1 INVITE => Call-ID: EOudzmeJA-U_KHJawRbA7w..
01 Mar 2022 13:25:12 Registracia noveho uzivatela[martinhric@147.175.183.39:5060]
```

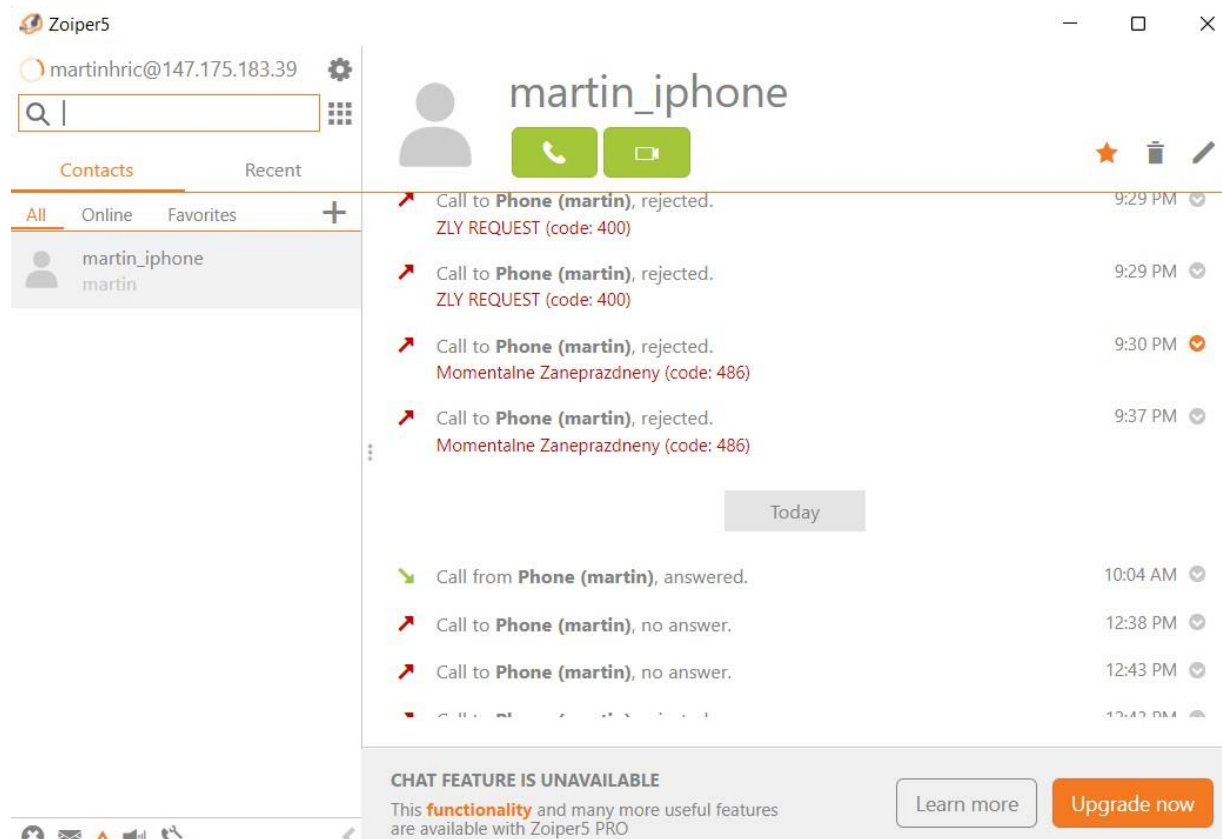
Zachytáva to registráciu nového užívateľa s vypísaním jeho adresy, vypísanie kto, kedy hovor prijal, kedy a kým bol hovor ukončený s ID daného hovoru. Žiadosť zrušená vypíše, keď ten ktorý začal volať zruší volanie, čiže hovor nebol ani dvihnutý. Ak daný hovor bol zrušený druhou stranou, čiže je User Busy vypíše že užívateľ zaneprázdnený.

```
if self.data[0] == 'SIP/2.0 100 Trying':
    data[0] = 'SIP/2.0 100 Skusam Vytocit'
elif self.data[0] == 'SIP/2.0 180 Ringing':
    data[0] = 'SIP/2.0 180 Zvoniiim'
elif self.data[0] == 'SIP/2.0 200 OK':
    data[0] = 'SIP/2.0 200 Poriesene OK'
    if data[7] == 'CSeq: 2 BYE':
        logs.write(time.strftime("%d %b %Y %H:%M:%S ", time.localtime()))
        logs.write("[Hovor ukonceny] => ")
        logs.write(f"{data[6]} || {data[5]} => {data[4]}\n")
    elif data[7] == 'CSeq: 1 INVITE':
        logs.write(time.strftime("%d %b %Y %H:%M:%S ", time.localtime()))
        logs.write("[Hovor prijaty] => ")
        logs.write(f"{data[6]} || {data[5]} => {data[4]}\n")
elif self.data[0] == 'SIP/2.0 487 Request Terminated':
    data[0] = 'SIP/2.0 487 Ziadost Zrusena'
    logs.write(time.strftime("%d %b %Y %H:%M:%S ", time.localtime()))
    logs.write("[Ziadost zrusena] => ")
    logs.write(f"{data[4]} || {data[3]} => {data[2]}\n")
elif self.data[0] == 'SIP/2.0 486 Busy Here':
    data[0] = 'SIP/2.0 486 Momentalne Zaneprazdneny'
    logs.write(time.strftime("%d %b %Y %H:%M:%S ", time.localtime()))
    logs.write("[Uzivatel zaneprazdneny] => ")
    logs.write(f"{data[4]} || {data[3]} => {data[2]}\n")
```

Martin Hric
ID: 111696

Ďalšou funkcionalitou ako sa už dá aj všimnúť v zdrojovom kóde vyššie je úprava SIP stavových kódov, ktoré sa zobrazujú aj vo wiresharku a Zoiper klientovi.

1252...	196.015647	147.175.190.161	147.175.183.39	SIP	663 Request: REGISTER sip:147.175.183.39:5060;transport=UDP (1 binding)
1252...	196.021287	147.175.183.39	147.175.190.161	SIP	673 Status: 200 FUNGUJE OK (1 binding)
1393...	219.126665	147.175.190.161	147.175.183.39	SIP/SDP	895 Request: INVITE sip:martinhric@147.175.183.39:5060;transport=UDP
1393...	219.275873	147.175.183.39	147.175.190.161	SIP	379 Status: 100 Skusam Vytocit
1406...	219.628044	147.175.183.39	147.175.190.161	SIP	644 Status: 180 Zvoniiim
1508...	236.688323	147.175.183.39	147.175.190.161	SIP	435 Status: 486 Momentálne Zaneprazdneny
1508...	236.712089	147.175.190.161	147.175.183.39	SIP	410 Request: ACK sip:martinhric@147.175.183.39:5060;transport=UDP
1598...	250.044869	147.175.190.161	147.175.183.39	SIP	663 Request: REGISTER sip:147.175.183.39:5060;transport=UDP (1 binding)
1598...	250.045446	147.175.183.39	147.175.190.161	SIP	673 Status: 200 FUNGUJE OK (1 binding)



Funkcionality ako videohovor, konferenčný hovor a presmerovanie hovoru sa mi nepodarilo otestovať, keďže tieto funkcionality sú v Zoiper5 dostupné len pre “PRO” účty, za ktoré sa platí 60eur jednorázová platba, čo som teda ochotný nebol za to zaplatiť ..

V kóde je to všetko zahrnuté a všetky tieto funkcionality sú implementované a podľa spolužiakov čo mi vraveli tak sú aj funkčné, je to len otázkou Klienta.

TESTOVANIE

V repozitári sú 4 .pcap súbory, ktoré zachytávajú 4 scenáre, na ktorých poukazujem na funkčnosť programu. Celé testovanie som vykonával na internáte v sieti Ynetu, kde z Windows PC volám na iOS mobilný telefón, pričom moj PC slúži aj ako Proxy server.

1. Testovací scenár:

- Zapnutie proxy následne zapnutie Zoiper na desktope, potom na mobile.
- Zavolanie z PC na mobil po dobu 10 sekúnd, následne zrušenie hovoru na PC.
- Vo Wiresharku sa nezobrazujú requesty REGISTER pre zaregistrovanie PC, ale do Denníka Hovorov sa zapisujú. Ak volám z PC na telefón, uvidím aj neprepísané stavové kódy, pretože tieto kódy idú z Klienta, čo samozrejme prepísať sa nedá, ak ale volám z telefónu na PC, vidím svoje prepísané kódy.
- Prichádza INVITE z proxy(rovnnaká IP ako PC) na mobil, po ktorom prichádzajú stavové správy 100 Trying, 180 Ringing, na ktoré odpovedám vlastnými(vôbec to nieje potrebné, do kódu som to len dal kvôli splneniu všetkých doplnkových funkcionalít). Ak sa hovor prijme, odvedá 200 OK. Následne prichádza veľa RTP packetov, ktoré sú posielané obojsmernie a tie zachytávajú zvukovú stopu po dobu 20ms.
- Na konci prichádza BYE, na ktorý sa odpovedá 200 OK

2. Testovací scenár:

- Volanie z mobilu na PC, čiže druhou stranou narozdiel od 1. Scenáru
- Zmena je len tá, že vo Wiresharku neuvidím packet zasielané od klienta

3. Testovací scenár:

- Zavolanie druhej strane ale druhá strana odmietne telefonovať, čiže mal by sa vyskytnúť kód 408 User Busy

4. Testovací scenár:

- Zavolanie druhej strane, kde ale ten ktorý začal telefonovať zruší volanie, mal by sa vyskytnúť kód 487 Request Terminated