Fakulta informatiky a informačných technológií
STU Bratislava

Mobilné technológie a aplikácie

Dokumentácia Zadanie1

Jakub Povinec
Čas cvičení: Streda 18:00

Zadanie

V tomto zadaní sme mali pomocou ľubovoľnej knižnice vytvoriť SIP proxy, ktoré umožní telefonovanie medzi dvomi (alebo viacerými) SIP klientami.

Link na github: https://github.com/kuko6/mtaa-sip-proxy

Riešenie

Zadanie som implementoval v jazyku Python 3.9.7. Použil som knižnicu PySipFullProxy[1] a voľne dostupný klient linphone[2].

Samotná implementácia nebola náročná. Zvolená knižnica a klient rovno podporovali všetky povinné a časť doplnkových funkcionalít. Pri knižnici bolo potrebné prepísať jej niektoré časti, pretože bola pôvodne napísaná v staršej verzií pythonu. Konkrétne sa jednalo o spôsobe zisťovania, či sa kľúč nachádza v slovníku (z *registrar.has_key(origin)* na *origin in registrar*) a pri spájaní reťazcov (z *string.join(data,"\r\n")* na *"\r\n".join(data)*). Taktiež bolo potrebné vymazať regex výrazy, ktoré blokovali určité IP adresy. Po týchto úpravách už bolo proxy funkčné a podporovalo:

- Registráciu účastníka
- Vytáčanie, zvonenie, prijatie a ukončenie hovoru
- Videohovor
- Presmerovanie
- Konferenčný hovor

V knižnici som ešte ďalej zmenil spôsob logovania komunikácie s jednotlivými klientami, ktorý sa uchováva do súboru *dennik_hovorov.log* a obsahuje informácie o registráciách a jednotlivých hovoroch medzi klientami. Ukážka tohoto súboru je na obrázku (Obr. 1).

```
15:28:04 > --------------------------------
15:28:04 > ==|Tue, 01 Mar 2022 15:28:04|==
15:28:04 > --------------------------------
15:28:04 > Proxy bezi na Kukov-Air.local - 192.168.1.194:5060
15:28:19 > Zaregistroval sa: ifu@192.168.1.194, adresa: 192.168.1.146:59066
15:28:21 > Zaregistroval sa: mac@192.168.1.194, adresa: 192.168.1.194:65045
15:28:25 > mac@192.168.1.194 zacal volanie ifu@192.168.1.194, [Call-ID: Xyk3eQXGAv]
15:28:25 > Zvoni
15:28:30 > Ok
15:28:36 > mac@192.168.1.194 polozil hovor
15:28:36 > Hovor [Call-ID: Xyk3eQXGAv] skoncil
15:28:36 > Ok
15:28:55 > ifu@192.168.1.194 zacal volanie mac@192.168.1.194, [Call-ID: 8oFq2En7BQ]
15:28:55 > Zvoni
15:28:58 > mac@192.168.1.194 odmietol hovor
15:29:02 > ifu@192.168.1.194 zacal volanie mac@192.168.1.194, [Call-ID: Rr9hUWXOzJ]
15:29:02 > Zvoni
15:29:33 > mac@192.168.1.194 neprijal hovor
15:29:57 > Zaregistroval sa: ifu@192.168.1.194, adresa: 192.168.1.146:59066
15:30:03 > --------------------------------
15:30:03 > ==|Tue, 01 Mar 2022 15:30:03|==
15:30:03 > --------------------------------
15:30:03 > Proxy bezi na Kukov-Air.local - 192.168.1.194:5060
15:30:04 > Zaregistroval sa: ifu@192.168.1.194, adresa: 192.168.1.146:59066
15:30:06 > Zaregistroval sa: mac@192.168.1.194, adresa: 192.168.1.194:65045
15:56:01 > --------------------------------
15:56:01 > ==|Tue, 01 Mar 2022 15:56:01|==
15:56:01 > --------------------------------
15:56:01 > Proxy bezi na Kukov-Air.local - 192.168.1.194:5060
15:56:26 > Zaregistroval sa: mac@192.168.1.194, adresa: 192.168.1.194:62470
```

Obr. 1 Ukážka logovania hovorov

[1] https://github.com/tirfil/PySipFullProxy
[2] https://www.linphone.org

Ďalej som ešte v knižnici zmenil SIP stavové kódy, ktoré proxy posiela:

- 200 Ok -> 200 Oki
- 180 Ringing -> 180 Zvoni
- 100 Trying -> 100 Skusam
- 603 Decline -> 603 Odmietnutie
- 486 Busy here -> 486 Neotravuj

Časť kódu, ktorá mení tieto stavové kódy sa primárne nachádza v metóde *processCode()* a funguje tak, že zmení text packetu predtým ako sa odošle. Ukážka packetov s upravenými kódmi sa nachádza na obrázkoch (Obr. 2 a Obr. 3).

```
SIP              1051 Status: 200 Oki (REGISTER)  (1 binding) |
SIP/SDP          1718 Request: INVITE sip:mac@192.168.1.194 |
SIP               339 Status: 100 Trying |
SIP/SDP           113 Request: INVITE sip:mac@192.168.1.194 |
SIP               286 Status: 100 Skusam |
SIP               513 Status: 180 Ringing |
SIP               458 Status: 180 Zvoni |
SIP/SDP          1276 Status: 200 Ok (INVITE) |
SIP/SDP          1224 Status: 200 Oki (INVITE) |
```

Obr. 2 Ukážka zmenených stavových kódov

```
1.194 SIP               341 Status: 100 Trying |
1.194 SIP               515 Status: 180 Ringing |
1.194 SIP               474 Status: 486 Busy here |
1.146 SIP               552 Request: ACK sip:ifu@192.168.1.194 |
1.194 SIP               401 Status: 486 Neotravuj |
1.194 SIP               407 Request: ACK sip:ifu@192.168.1.194 |
```

Obr. 3 Ukážka zmenených stavových kódov 2

Testovanie

Proxy som testoval na troch rôznych zariadeniach so zvoleným klientom. Hovory medzi dvomi účastníkmi som kvôli prehľadnosti radšej realizoval na zariadeniach, na ktorých zároveň nebolo spustené samotné proxy, ktoré som ale použil pri testovaní presmerovania hovorov alebo pri konferenčnom hovore. Správnosť implementovaného proxy som si overoval vo wiresharku, pomocou jednotlivých packetov a vygenerovaných diagramov.

Registrácia

Registrácia prebieha tak, že klient pošle proxy požiadavku **REGISTER**, na ktorú, ak je v poriadku, dostane odpoveď **200 Ok**. Ukážka registrácie 2 klientov je na obrázku (Obr. 4).

```
1 0.000000 192.168.1.128 192.168.1.194 SIP              671 Request: REGISTER sip:192.168.1.194  (1 binding) |
2 0.001719 192.168.1.194 192.168.1.128 SIP              692 Status: 200 Oki (REGISTER)  (1 binding) |
3 3.592020 192.168.1.156 192.168.1.194 SIP             1030 Request: REGISTER sip:192.168.1.194  (1 binding) |
4 3.593991 192.168.1.194 192.168.1.156 SIP             1051 Status: 200 Oki (REGISTER)  (1 binding) |
```

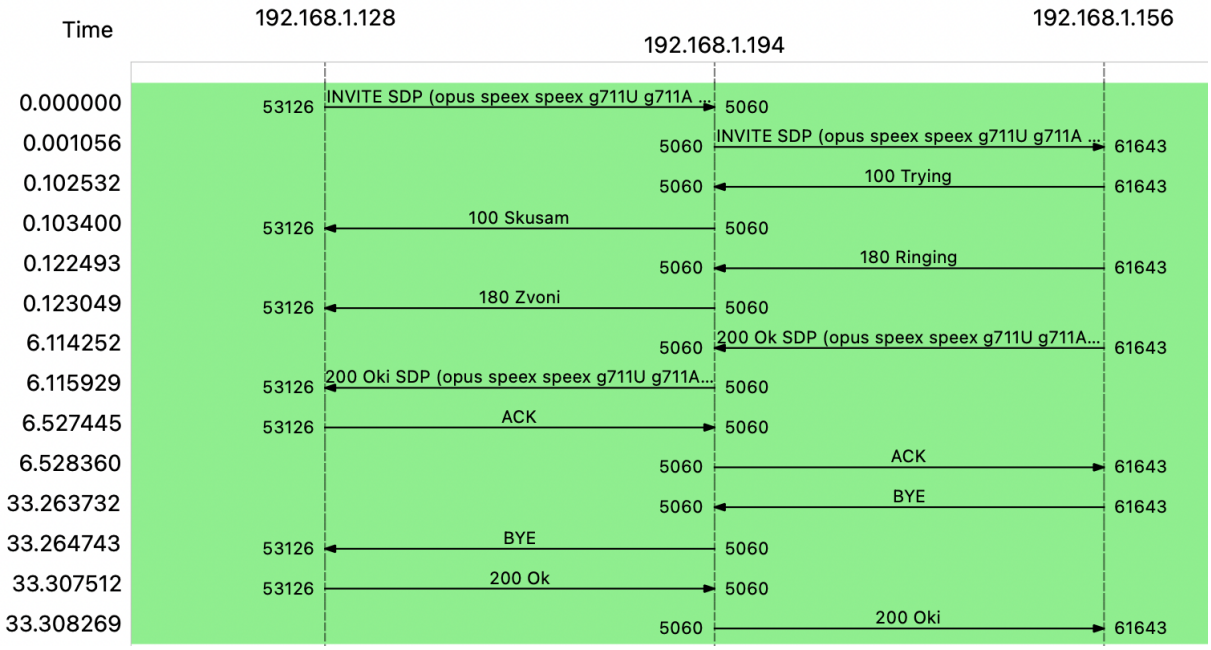
Obr. 4 Registrácia

Hovor

Samotný hovor sa skladá z viacerých častí. Prvá časť je vytočenie hovoru, kedy klient, ktorý začal hovor pošle druhému klientovi cez proxy požiadavku **INVITE**. Na túto požiadavku druhý klient odpovedá packetom **100 Trying** a **180 Ringing**, kedy začal používateľa druhého klienta upozorňovať na prichádzajúci hovor.

Následne môže používateľ hovor prijať, kedy druhý klient odpovie packetom **200 Ok**, na ktorý prvý klient odpovie **ACK** a hovor sa začne. Ak používateľ hovor odmietne, druhý klient odošle packet **603 Decline**, na ktorý prvý klient odpovie **ACK**. Ak používateľ do nejakeho určeného času hovor nezdvihne, druhý klient zašle packet **486 Busy here**, na ktorý prvý klient odpovie **ACK** a hovor tým pádom taktiež skončí.

Pri zrušení hovoru, odošle strana, ktorá ho zrušila packet **BYE**, na ktorý dostane odpoveď **200 Ok**. Ukážka štandardného hovoru je na obrázku (Obr. 5).



Obr. 5 Ukážka hovoru

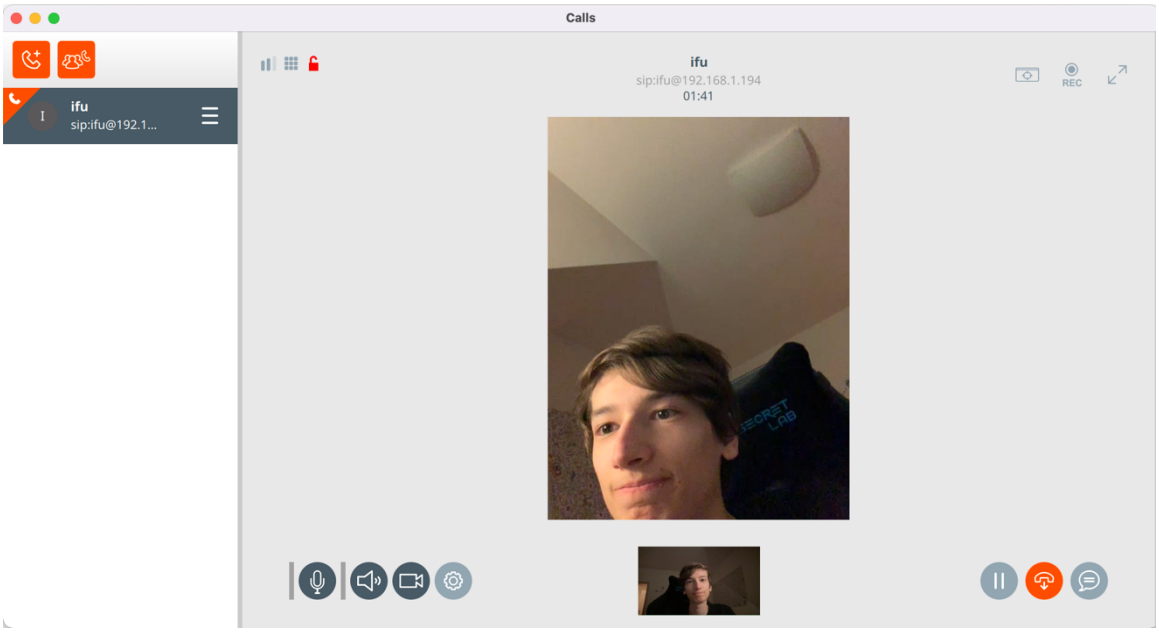
Na diagrame chýba prenos dát (v tomto prípade zvuku) pomocou protokolu RTP. Neprišiel som na to prečo, ale wireshark mi pri týchto dvoch zariadeniach nezachytával RTP. Preto som dal na github aj pcap súbory, kde tieto protokoly sú, je to ale hovor medzi zariadením, na ktorom beží proxy a teda to je trochu neprehľadné.

Videohovor

Videohovor je veľmi podobný štandardnému hovoru. Pri zapnutí kamery si klienti znovu vymenia packety **INVITE**, **Trying**, **Ok** a **ACK** a taktiež sa posielajú dva typy dát (zvuk a video). Na obrázku (Obr. 6) je ukážka videohovoru.

Presmerovanie

Pri presmerovaní hovorov, strana, ktorá presmerovanie vyvolala odošle druhej strane požiadavku **REFER**, ktorá oznamuje kam sa hovor presmeruje. Druhá strana na túto požiadavku odpovie packetom **202 Accepted**. Následne prebieha akoby nový hovor medzi klientom, na ktorý sa presmeroval a druhým klientom. Súčasne ako hovor prebieha, posiela druhý klient prvému packety **NOTIFY**.



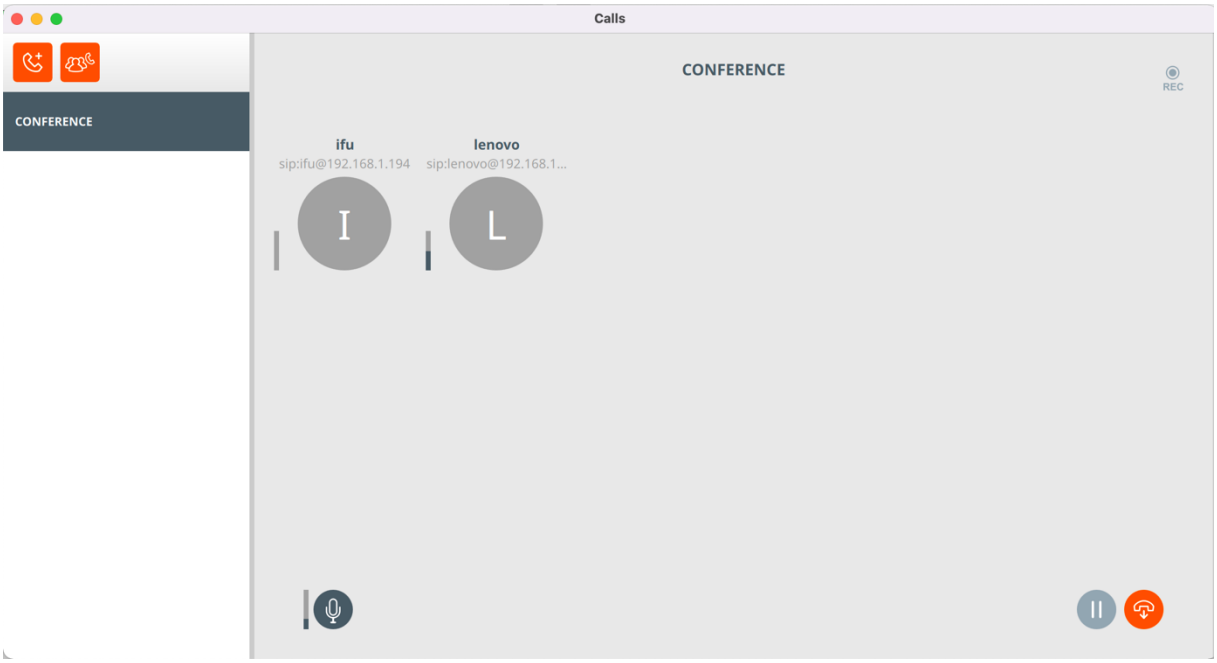
Obr. 6 Ukážka videohovoru

Konferenčný hovor

Konferenčný hovor prebieha ako viac štandardných hovorov medzi účastníkmi a organizátorom s tým, že účastníci takéhoto hovoru posielajú požiadavky **SUBSCRIBE**, na ktoré dostanú odpoveď **200 Ok**. Ukážka týchto požiadaviek je na obrázku (Obr. 7) a ukážka konferenčného hovoru je na obrázku (Obr. 8).

```
77… 43.4529… 192.168.1.156 192.168.1.194 SIP      1372 Request: SUBSCRIBE sip:ifu@192.168.1.156:61643;pn-
77… 43.4533… 192.168.1.194 192.168.1.156 SIP      1003 Status: 200 Oki (SUBSCRIBE) |
77… 43.4850… 192.168.1.128 192.168.1.194 SIP       653 Request: SUBSCRIBE sip:lenovo@192.168.1.128:53126;
78… 43.4854… 192.168.1.194 192.168.1.128 SIP       635 Status: 200 Oki (SUBSCRIBE) |
```

Obr. 7 Ukážka požiadaviek SUBSCRIBE



Obr. 8 Ukážka konferenčného hovoru

Spúšťanie

Proxy sa spúšťa zo súboru *proxy.py*. Používateľ môže zadať IP adresu servera ako argument, napríklad:

```
python3 src/proxy.py 192.168.1.194
```

Alebo sa dá program spustiť aj bez argumentov, kedy sa na IP adresu spýta po spustení. Číslo portu pre proxy sa automaticky nastaví na `5060`.

Program som neskúšal na inej verzií pythonu ako 3.9.7 a na inom os ako macOS, ale nemyslím si, že by mal byť nejaký problém.