

SIP Proxy - ústredňa*

Matej Pakán

ID: 110867

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií
`xpakan@stuba.sk`

1. marca 2022

*1. zadanie - SIP proxy – v predmete mobilné technológie a aplikácie, ak. rok 2021/22,
cvičiaci: Ing. Matej Janeba

Obsah

1	Znenie zadania	3
1.1	Hlavná myšlienka	3
1.2	Doplňujúce informácie k zadaniu	3
1.3	Rozsah povinných funkcionalít	3
1.4	Doplňkové funkcionality	3
2	Úvod	4
3	Spúšťanie knižnice	4
4	Scenáre - povinné funkcionality	6
4.1	Registrácia účastníka	6
4.2	Vytočenie hovoru a zvonenie na druhej strane	6
4.3	Prijatie hovoru druhou stranou, fungujúci hlasový hovor	7
4.4	Ukončenie hlasového hovoru (prijatého aj neprijatého)	8
5	Scenáre - doplnkové funkcionality	9
5.1	Realizácia konferenčného hovoru (aspoň 3 účastníci)	9
5.2	Presmerovanie hovoru	10
5.3	Realizácia videohovoru	11
6	Záver	11

1 Znenie zadania

1.1 Hlavná myšlienka

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

1.2 Doplnujúce 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 dvomi **fyzickými** zariadeniami v rámci LAN siete.

1.3 Rozsah povinných funkcionalít

- 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)

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

1.4 Doplnkové funkcionality

- 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”

Každá doplnková funkcionalita predstavuje plus 1 bod.

Počas prezentácie zadania musíte byť schopní na zariadení, kde beží ústredňa urobiť SIP trace a otvoriť ho pomocou tcpdump alebo Wireshark, a v primeranom rozsahu vysvetliť cvičiacemu, ako daná signalizácia prebieha.

2 Úvod

Hneď od začiatku som vedel, že projekt budem chcieť realizovať v Pythone, kvôli najlepšiemu porozumeniu tohto jazyka a jednoduchým troubleshooting. Ako knižnicu som použil PySipFullProxy na odporúčanie vedúceho cvičení. Ako SIP klienta som použil Linphone, kvôli jednoduchému nastaveniu klientov.

Verejný github repozitár k zdrojovému kódu a WireShark scenárom:
<https://github.com/Matt1s/MTAA-SIP-PROXY>

3 Spúšťanie knižnice

Knižnicu PySipFullProxy bolo nutné upraviť, kvôli nekompatibilite s mojou verziou Pythonu (3.10). Väčšinou išlo o chyby pri funkcii `join()` alebo pri **zakódovaní/dekódovaní z/do UTF-8**

Niekoľko riadkov som teda musel upraviť, napríklad:

```
|| text = string.join(data, "\r\n")
na
|| text = "\r\n".join(data)
a
|| self.socket.sendto(text, self.client_address)
na
|| self.socket.sendto(text.encode(), self.client_address)
a
|| if registrar.has_key(origin):
na
|| if origin in registrar:
```

Takisto som odstránil aj celú vetvu obsahujúcu `rx_invalid` a `rx_invalid2`, kvôli vypnutiu reštrikcií na správny beh servera.

Následne po odstránení všetkých chýb som spúšťal svoj `main.py` s použitím knižnice tak, ako som uznal za vhodné.

Môj `main.py`:

```
import sipfullproxy as SIP
import socket
def main():
    # moved these 2 variables as they are no longer needed in
    # sipfullproxy.py file
    HOST = '0.0.0.0'
    PORT = 5060

    # edited filename
    SIP.logging.basicConfig(format='%(asctime)s: %(levelname)s: %(
message)s', filename='zaznam.log', level=SIP.logging.INFO,
datefmt='%H:%M:%S')
    SIP.logging.info(SIP.time.strftime("%a, %d %b %Y %H:%M:%S ",
SIP.time.localtime()))

    #selecting specific IP address
```

```
    ipaddress = socket.gethostbyname_ex(socket.gethostname())[2][0]
    if ipaddress == "127.0.0.1":
        ipaddress = SIP.sys.argv[1]
    SIP.logging.info(ipaddress)
    SIP.recordroute = "Record-Route: <sip:%s:%d;lr>" % (
        ipaddress, PORT)
    SIP.topvia = "Via: SIP/2.0/UDP %s:%d" % (ipaddress, PORT)

    # setting up server
    server = SIP.socketserver.UDPServer((HOST, PORT), SIP.
        UDPHandler)
    try:
        print('Server starting at ' + ipaddress)
        server.serve_forever()
    except:
        server.server_close()
        print('Server stopped, closing program...')
main()
```

V tomto riadku:

```
|| ipaddress = socket.gethostbyname_ex(socket.gethostname())[2][0]
```

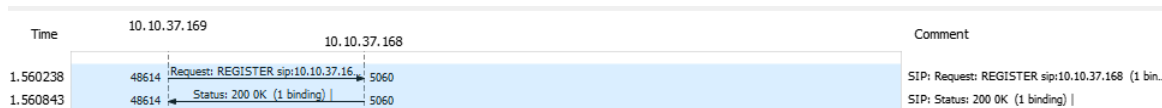
Vyberám zo zoznamu dostupným rozhraní, na ktorom konkrétnom rozhraní a s akou IP sa spustí môj SIP proxy server.

Následne pomocou **try/except** spúšťam server s informačnou správou, že kde je spustený. Pri násilnom vypnutí (CTRL+C) sa server vypne.

4 Scenáre - povinné funkcionality

4.1 Registrácia účastníka

Analýza súboru '200 OK - registracia ucastnika.pcapng'- filter 'sip'

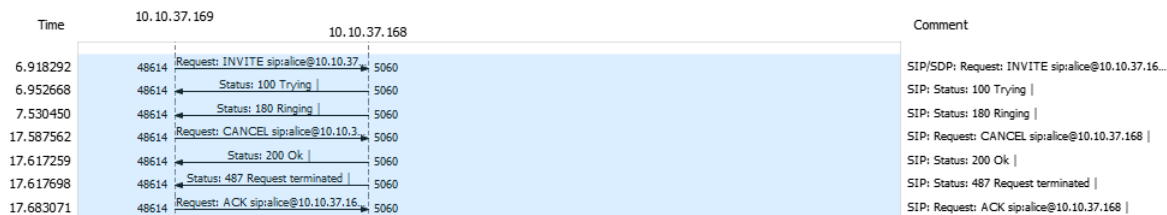


Obr. 1: Flow chart registrácia klienta

Žiadosť o registráciu sa snaží prejsť z '10.10.37.139' (môj mobil), ktorý už je na lokálnej sieti. Proxy server tohto používateľa zapíše a odpovie klientovi odpoveď STATUS: 200 OK, čo znamená, že klient bol úspešne zaregistrovaný. Keďže nerobíme autorizáciu klientov, tak tam nič nechýba. Ak by sme autorizáciu robili, potrebovali by sme ešte jeden REGISTER a odpoveď 401 Unauthorized s registračnými údajmi. Ústredňa teda používateľa registruje napriamo bez autorizácie. V message header správy REGISTER je 'to' a 'from' rovnaké.

4.2 Vytocenie hovoru a zvonenie na druhej strane

Analýza súboru '100 trying 180 ringing - vytocenie a zvonenie.pcapng' - filter 'sip'

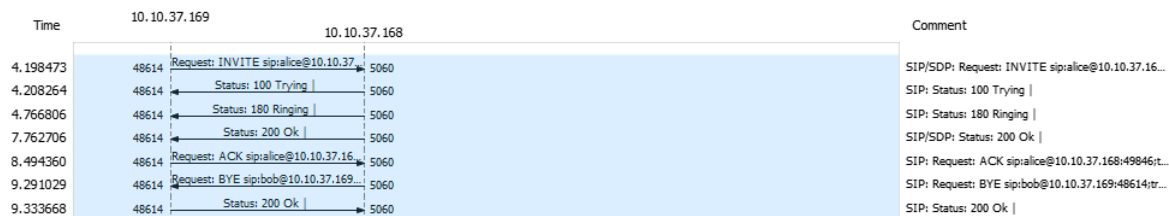


Obr. 2: Flow chart vytocenie a zvonenie klienta

Po úspešnej registrácii zo strany mobilu s IP '10.10.37.139' odchádza INVITE pre 'alice@10.10.37.168'. Z ústredne následne prichádza odpoveď 100 Trying a 180 Ringing. Teda na druhej strane (Alice) začína zvoniť. Po pár sekundách Bob zruší hovor kliknutím na tlačidlo skončiť, teda prechádza Request CANCEL, odpoveď 200 OK a teda hovor bol zrušený zo strany Boba (volajúci).

4.3 Prijatie hovoru druhou stranou, fungujúci hlasový hovor

Analýza súboru '200 OK - hlasovy hovor.pcapng' - filter 'sip' Hlasový hovor



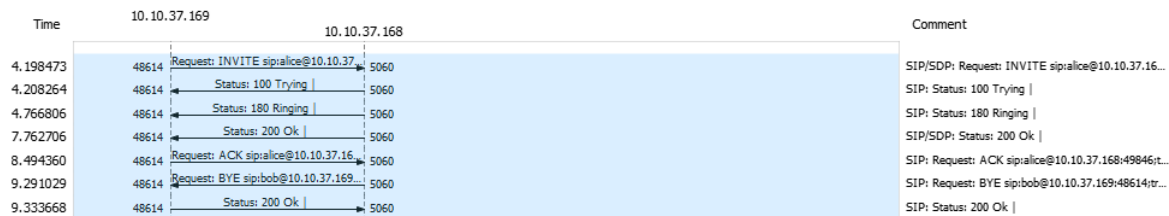
Obr. 3: Flow chart hlasový hovor

prebieha po prijatí druhou stranou (Bob volá Alice). Následne 2 sekundy prebieha hovor cez SIP/SDP protokol. V invite správe si môžeme pozrieť ponúknuté kodeky, ktoré ak druhá strana spustí, budú medzi sebou správne komunikovať. V INVITE takisto oznamuje, na akom porte bude daný klient počúvať.

Pri aplikovaní filtra 'sip || rtp' je možné si zobrazit' aj tok hlasových dát a informácie o tom, aké kodeky boli použité.

4.4 Ukončenie hlasového hovoru (prijatého aj neprijatého)

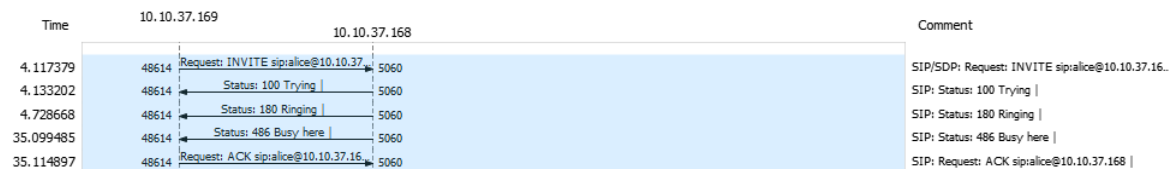
Analýza súboru '200 OK - hlasový hovor.pcapng' - filter 'sip'



Obr. 4: Flow chart hlasový hovor (rovnaký obr. ako vyššie)

Na konci flowu vidíme BYE a ACK, teda signalizačné správy o tom, že nadviazaný hovor je ukončený a druhá strana je s tým oboznámená.

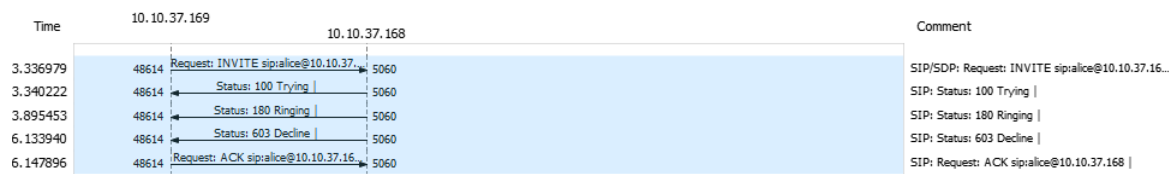
Analýza súboru '486 Busy here - neprijaty hovor.pcapng' - filter 'sip'



Obr. 5: Flow chart 486 Busy here

Podobný scenár ako v sekcii 'Vytočenie hovoru a zvonenie na druhej strane', akurát ústredňa vracia po 30 sekundách 486 Busy here, teda Bob vytočil Alice, ale tá nestihla zdvihnúť do doby 30 sekúnd. Hovoríme teda o neprijatom hovore.

Analýza súboru '603 Decline - odmietnutie hovoru.pcapng' - filter 'sip'



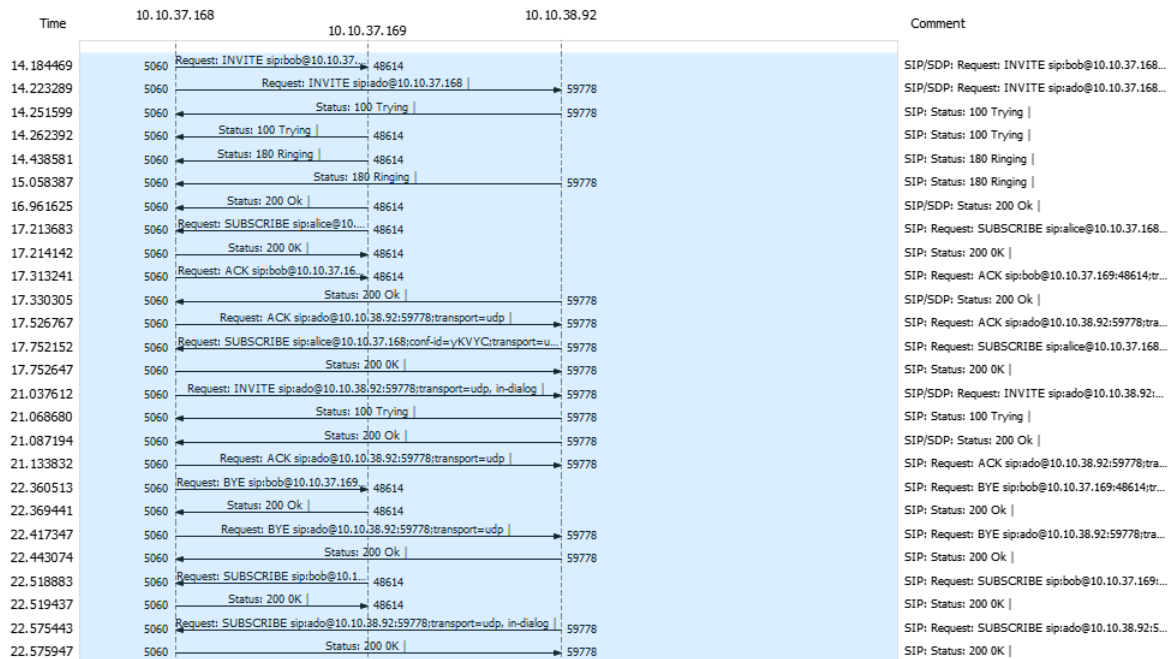
Obr. 6: Flow chart 603 Decline

Rovnaký scenár ako vyššie, miesto nezdvihnutia (ignorovania) ale Alice zrušila (odmietla) hovor, preto Bob dostal informáciu z ústredne 603 Decline.

5 Scenáre - doplnkové funkcionality

5.1 Realizácia konferenčného hovoru (aspoň 3 účastníci)

Analýza súboru 'KONFERENCNY HOVOR.pcapng' - filter 'sip'

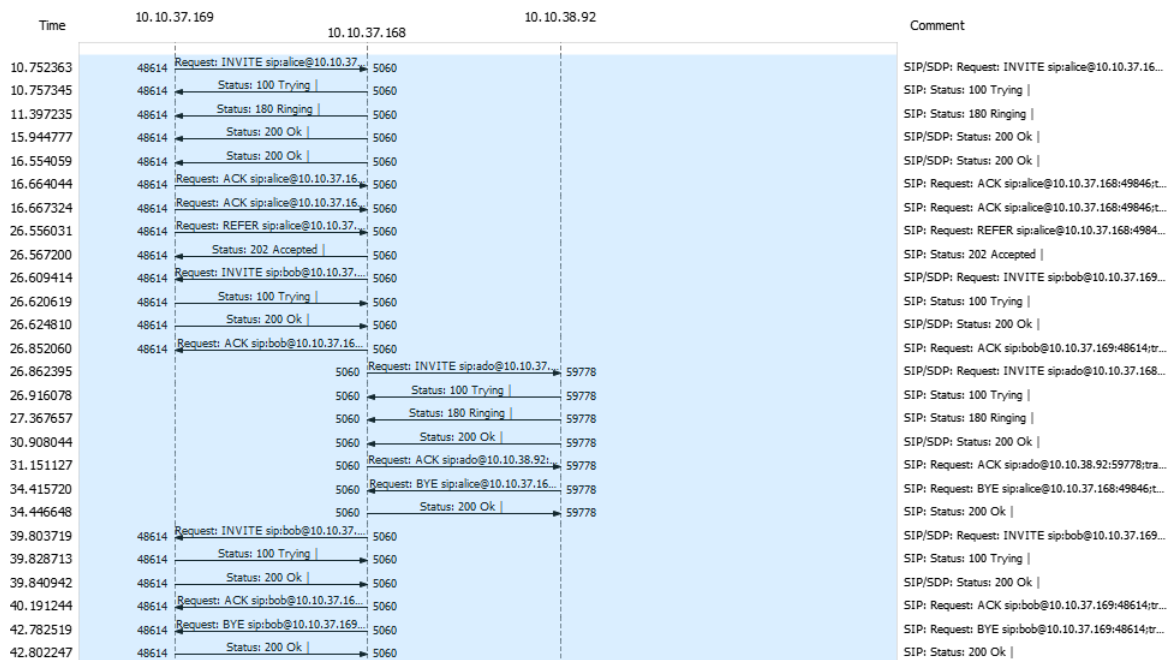


Obr. 7: Flow chart konferenčný hovor (3 ľudia)

Pozvánka na konferenčný išla od užívateľa Alice užívateľom Bob a Ado. Na obrázku preto vidíme 2x 100 Trying a 180 Ringing. Odpovede 200 OK nasvedčujú tomu, že všetko prebieha v poriadku a hovor sa spustil medzi 3 ľuďmi. Následne vidíme novú správu SUBSCRIBE o tom, že volajúci chce byť informovaný ak sa ďalší účastník pripojí do hovoru. Neskôr zase vidíme správu BYE od používateľa Ado. To znamená, že hovor pokračuje už len medzi 2 ľuďmi.

5.2 Presmerovanie hovoru

Analýza súboru 'PRESMEROVANIE.pcapng' - filter 'sip'

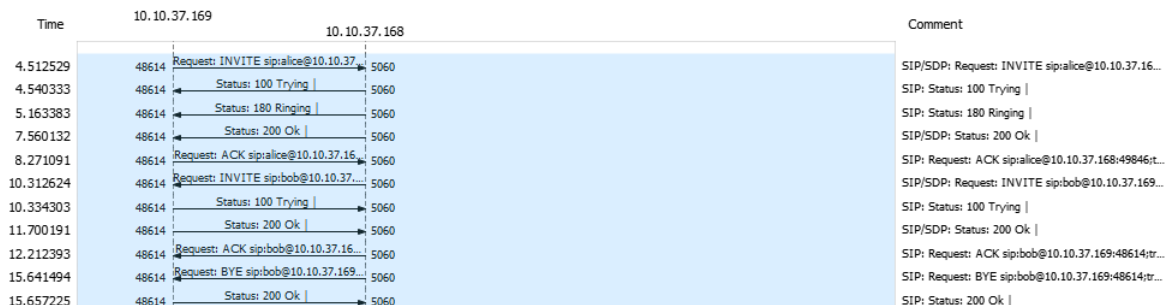


Obr. 8: Flow chart presmerovanie

Na obrázku vidíme, ako bola nadviazaná komunikácia medzi Bobom (10.10.37.169) a Alice (10.10.37.168). Následne Bob presmeroval hovor na Ado (10.10.37.92), tým pádom hovor prebiehal iba medzi Alice a Ado (u Boba hrá hudba). Potom Ado hovor zrušil (BYE a ACK) a hovor sa vrátil do normálu medzi Bob a Alice. Nakonci prebieha štandardné ukončenie hovoru pomocou BYE a ACK.

5.3 Realizácia videohovoru

Analýza súboru 'VIDEOHOVOR.pcapng' - filter 'sip'



Obr. 9: Flow chart videohovor

Rozdiel oproti štandardnému hlasovému hovoru je v tom, že po nadviazaní hlasového hovoru ide ešte jeden INVITE, 100 Trying a 200 OK (žiadosť o zapnutie kamery). Hovor sa inak líši iba v použitých kodekoch. Nakonci záznamu je štandardné ukončenie hovoru pomocou BYE a ACK.

6 Záver

Čakal som že bude zložitejšie rozbehať vlastný proxy server na realizáciu hovoru, za použitia knižnice to bolo omnoho jednoduchšie. Pomocou WireSharku bolo pekne vidieť ako sa všetky packety presúvajú. Bol pekný pocit keď sa podarilo rozbehnúť videohovor :)

Verejný github repozitár k zdrojovému kódu a WireShark scenárom:
<https://github.com/Matt1s/MTAA-SIP-PROXY>