

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií

Mobilné technológie a aplikácie

Zadanie č. 1 – SIP Proxy

Lívia Kostová

110825

2021/2022

Obsah

Zadanie 1 – SIP Proxy (telefónna ústredňa)	3
Úvod.....	5
Opis riešenia.....	5
Testovanie	6
Link na repozitár	7
Používateľské rozhranie.....	7
Voľba implementačného prostredia	7

Zadanie 1 – SIP Proxy (telefónna ústredňa)

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.

Doplnkové 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”

Každá doplnková funkcionality 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.

Forma odovzdania:

- Dokumentácia, kde opíšete, ako ste vaše riešenie implementovali, aké knižnice ste použili, a ako ste ústredňu sprevádzkovali, vo formáte PDF do miesta odovzdania v AIS. V PDF dokumente musí byť odkaz na váš repozitár v Gite (verejný repozitár, ľubovoľná služba)
- Vo vašom repozitári sa musí nachádzať PCAP trace z každého scenára, ktorý vaša implementácia poskytuje.

Termín odovzdania:

- 1.3.2022 23:59 do miesta odovzdania v AIS

Užitočné odkazy:

- Zoznam SIP knižníc pre rôzne programovacie jazyky (v sekcii SIP Protocol Stacks and Libraries):

o <https://www.voip-info.org/open-source-voip-software/>

- Odporúčané SIP klienty na testovanie hovorov:

o Linphone: <https://www.linphone.org/>

o Zoiper: <https://www.zoiper.com/>

o Jitsi: <https://desktop.jitsi.org/Main/Download.html>

o Ďalšie sú tu, v sekcii SIP Clients: <https://www.voip-info.org/open-source-voip-software/>

Úvod

Našou úlohou pre prvé zadanie je sprevádzkovať SIP Proxy, ktorá umožní prepájanie a realizáciu hovorov medzi štandardnými SIP klientami. Hovor je realizovaný medzi dvomi fyzickými zariadeniami v rámci LAN siete. Na implementáciu sme si zvolili programovací jazyk Python a ako SIP klientov pre testovanie hovorov sme volili Linphone a Zoiper. Podarilo sa nám splniť všetky povinné funkcionality, možnosť zrealizovať konferenčný hovor, video hovor, presmerovanie hovoru a upraviť SIP stavový kód v zdrojovom kóde.

Opis riešenia

Link na prevzatú použitú knižnicu:

<https://github.com/tirfil/PySipFullProxy/blob/master/sipfullproxy.py>

Pri riešení zadania sme zistili, že niektoré časti kódu v knižnici je potrebné upraviť, aby program správne fungoval. Odstránili sme regexy (rx_invalid, rx_invalid2), prerobili sme podmienky, kde sa nachádzalo `.has_key()` na `"<key> in <dictionary>"` a `string.join(data, oddeľovač)` na `oddeľovač.join(data)`, konkrétne `"\r\n".join(data)`, a tiež sme pridali `encode(„utf-8“)` (Obr.1). Vo funkcii `processCode` sme upravili stavové kódy na slovenskú verziu (Obr.2).

```
if registrar.has_key(destination) and self.checkValidity(destination):
    socket, claddr = self.getSocketInfo(destination)
    #self.changeRequestUri()
    self.data = self.addTopVia()
    data = self.removeRouteHeader()
    #insert Record-Route
    data.insert(1, recordroute)
    text = string.join(data, "\r\n")

if destination in registrar and self.checkValidity(destination):
    socket, claddr = self.getSocketInfo(destination)
    # self.changeRequestUri()
    self.data = self.addTopVia()
    data = self.removeRouteHeader()
    # insert Record-Route
    data.insert(1, recordroute)
    text = "\r\n".join(data).encode("utf-8")
```

(Obr.1)

```
def processCode(self):
    origin = self.getOrigin()
    if len(origin) > 0:
        logging.debug("origin %s" % origin)
        if origin in registrar:
            socket, claddr = self.getSocketInfo(origin)
            self.data = self.removeRouteHeader()
            data = self.removeTopVia()
            print(data)
            x = data[0].split()
            if (x[1] == "100"):
                data[0] = "SIP/2.0 100 Skusa"
            elif (x[1] == "180"):
                data[0] = "SIP/2.0 180 Zvoni"
            elif (x[1] == "200"):
                data[0] = "SIP/2.0 200 Dobre"
```

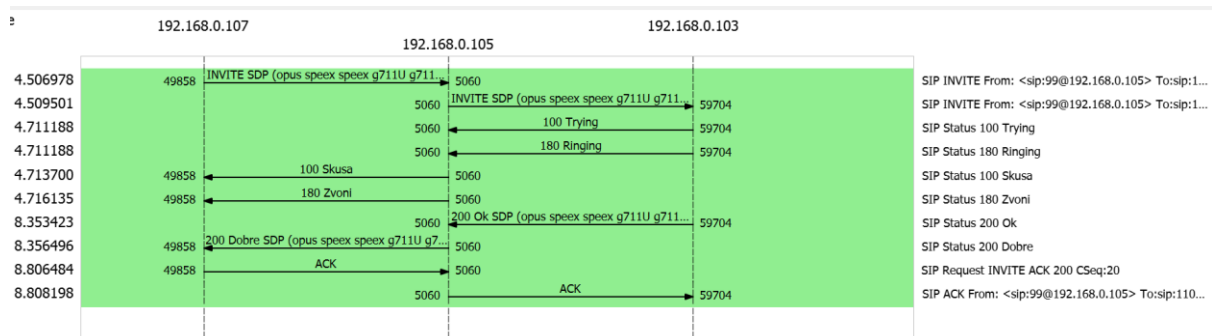
(Obr.2)

Nami vytvorený main pozostáva z niekoľkých častí. Najprv voláme pomocnú funkciu create_file(). V tejto funkcii vytvoríme pomocný file typu .log, do ktorého sa počas behu programu budú zapisovať všetky potrebné informácie, taktiež v tejto funkcii zistíme host name. V druhej funkcii ip_address() sa overí správnosť zadanej IP adresy, ktorá je zadávaná ako prvý argument príkazového riadka. Návrátová hodnota tejto funkcie je práve IP adresa v podobe stringu. Následne už v maine voláme knižnicu proxy, konkrétne UDPServer.

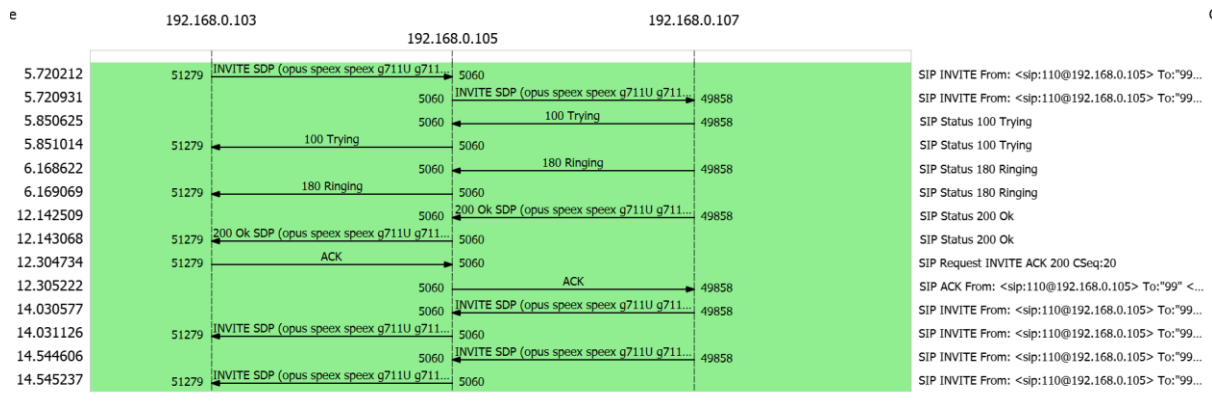
Testovanie

Testovanie riešenia sme realizovali pomocou aplikácií Linphone aj Zoiper. V oboch prípadoch prebehlo testovanie úspešne.

Príklady niektorých Flow Sequence:



(Obr. 3 – prijatie hovoru)



(Obr.4 – video hovor)

Link na repozitár

<https://github.com/Livia103/MTA>

Používateľské rozhranie

Program spúšťame cez terminál zadáním príkazu `python main.py „IP adresa“`. Pre ukončenie programu/spojenia používame kombináciu `Ctrl+c`.

Voľba implementačného prostredia

Pre realizáciu Zadania č.1 sme si zvolili ako implementačné prostredie programovací jazyk Python 3.9. Využívali sme IDE - PyCharm Community Edition 2021.2.1.