

Mobilné technológie a aplikácie 2021/2022
FIIT STU, Bratislava

Zadanie 1 – SIP PROXY (Telefónna ústredňa)

Dokumentácia

Patrik Harmaňoš

Hlavná myšlienka zadania:

Na vašom PC sprevádzkovať SIP PROXY, ktorá umožní prepájanie a realizáciu hovorov medzi štandardnými SIP klientami.

Implementácia zadania:

Zadanie je implementované v programovacom jazyku Python s použitím SIP PROXY knižnice PySipFullProxy. Spomínaná knižnica bola pôvodne implementovaná v staršej verzii Pythonu, takže pred jej použitím som musel urobiť drobné zmeny, napríklad pri funkciách na prácu s reťazcami, ktoré sú v novej verzii vyradené.

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

Pred odosielaním socketov som pridal kódovanie posiadaného obsahu pomocou funkcie `encode()` a taktiež po prijatí socketov som obsah dekodoval pomocou funkcie `decode()`.

```
text = text.encode('utf-8')
socket.sendto(text, claddr)
```

```
data = self.request[0].decode('utf-8')
```

Následne som si vytvoril môj súbor `main.py`, do ktorého som danú knižnicu importoval. Ihneď po spustení si program vypýta od používateľa zadať IP adresu, na ktorej bude SIP PROXY server bežať. Následne sa daná IP adresa pošle do funkcií importovanej knižnice a spustí sa server.

```
import socketserver
import time
import logging
import sipproxy

if __name__ == "__main__":
    sipproxy.HOST = input("Zadajte IP adresu: ")
    logging.basicConfig(format='%(asctime)s: %(levelname)s: %(message)s', filename='proxy.log', level=logging.INFO,
                        datefmt='%H:%M:%S')
    logging.info(time.strftime("%a, %d %b %Y %H:%M:%S ", time.localtime()))
    sipproxy.recordroute = "Record-Route: <sip:%s:%d;\r>" % (sipproxy.HOST, sipproxy.PORT)
    sipproxy.topvia = "Via: SIP/2.0/UDP %s:%d" % (sipproxy.HOST, sipproxy.PORT)
    server = socketserver.UDPServer((sipproxy.HOST, sipproxy.PORT), sipproxy.UDPHandler)
    server.serve_forever()
```

Použitý klient:

Ako klienta na uskutočňovanie hovorov som použil aplikáciu Linphone (<https://www.linphone.org/>), ktorú som nastavil na mobilnom zariadení aj na laptope, aby som mohol navzájom telefonovať.

Implementované funkcionality:

- Povinné
 - Registrácia účastníka
 - Vytočenie hovoru a zvonenie na druhej strane
 - Prijatie hovoru druhou stranou, fungujúci hlasový hovor
 - Ukončenie hlasového hovoru
- Doplnkové
 - Konferenčný hovor
 - Presmerovanie hovoru
 - Video-hovor
 - Logovanie denníka hovorov
 - Úprava SIP stavových kódov

Logovanie denníka hovorov:

```
11:19:17:INFO:Sun, 27 Feb 2022 11:19:17
11:20:52:INFO:Hovor od pouzivatela: mobil@192.168.0.164, komu: mackbook@192.168.0.164 || Call ID: XozEHnt8qk
11:20:52:INFO:Pouzivatel mackbook@192.168.0.164 zodvihol hovor || Call ID: XozEHnt8qk
11:20:53:INFO:Pouzivatel mackbook@192.168.0.164 zodvihol hovor || Call ID: XozEHnt8qk
11:20:55:INFO:Pouzivatel mackbook@192.168.0.164 zodvihol hovor || Call ID: XozEHnt8qk
11:20:56:INFO:Pouzivatel mackbook@192.168.0.164 zodvihol hovor || Call ID: XozEHnt8qk
11:20:58:INFO:Hovor od pouzivatela: mobil@192.168.0.164, komu: mackbook@192.168.0.164 || Call ID: XozEHnt8qk
11:20:58:INFO:Pouzivatel mackbook@192.168.0.164 zodvihol hovor || Call ID: XozEHnt8qk
11:20:58:INFO:Pouzivatel mackbook@192.168.0.164 zodvihol hovor || Call ID: XozEHnt8qk
11:20:58:INFO:Hovor od pouzivatela: mackbook@192.168.0.164, komu: mobil@192.168.0.164 || Call ID: XozEHnt8qk
11:20:58:INFO:Pouzivatel mobil@192.168.0.164 zodvihol hovor || Call ID: XozEHnt8qk
11:20:59:INFO:Pouzivatel mobil@192.168.0.164 zodvihol hovor || Call ID: XozEHnt8qk
11:21:02:INFO:Hovor od pouzivatela: mobil@192.168.0.164, komu: mackbook@192.168.0.164 || Call ID: XozEHnt8qk
11:21:02:INFO:Pouzivatel mackbook@192.168.0.164 zodvihol hovor || Call ID: XozEHnt8qk
11:21:02:INFO:Pouzivatel mackbook@192.168.0.164 zodvihol hovor || Call ID: XozEHnt8qk
11:21:03:INFO:Pouzivatel mackbook@192.168.0.164 zodvihol hovor || Call ID: XozEHnt8qk
11:21:11:INFO:Pouzivatel mackbook@192.168.0.164 ukoncil hovor. || Call ID: XozEHnt8qk
```

Vytvoril som funkcie na zistenie Call ID a na zistenie typu odpovede. Taktiež som pridal regex reťazec, podľa ktorého sa to kontroluje.

```
rx_type = re.compile("^CSeq:")
rx_id = re.compile("^Call-ID:")
```

```
def getType(self):
    for line in self.data:
        if rx_type.search(line):
            return line
    return ""

def getId(self):
    for line in self.data:
        if rx_id.search(line):
            return line[9:]
    return ""
```

Ak jedno zariadenie pošle druhému INVITE, tak následne chceme zistiť, či to druhé zariadenie zodvihlo hovor, tak musíme nájsť odpoveď 200 OK, ale v poli CSeq musí byť INVITE, to znamená, že odpoveď prišla na daný INVITE.

2063 117.838357 192.168.0.164 192.168.0.213 SIP/SDP 530 Status: 200 OK (INVITE) |

CSeq: 22 INVITE

Úprava SIP stavových kódov:

```
self.sendResponse("200 V poriadku")
```

Status: 200 V poriadku (REGISTER)

Odkaz na GitHub repozitár:

https://github.com/PatrikHarmanos/mtaa_sip_proxy