

MTAA - ZADANIE 1

SIP Proxy (telefónna ústredňa)

Timotej Smoleň

2021/2022

Slovenská technická univerzita v Bratislave

Fakulta informatiky a informačných technológií

28.02.2022

1. Zadanie

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

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)

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”

2. Implementačné prostredie a knižnice

Programovací jazyk, v ktorom bol program vyvíjaný je Python verzie 3.9.7. Použité integrované vývojové prostredie bolo PyCharm 2021.2.2 (Professional Edition). Nosnou knižnicou na funkcionalitu SIP proxy bola knižnica PySipFullProxy¹, ktorú bolo potrebné následne upraviť. V rámci importovanej knižnice sú používané ďalšie knižnice ako socket, SocketServer, re, či sys.

3. SIP proxy knižnica

Hlavná využívaná funkcionalita knižnice spočíva v triede UDPHandler, ktorá spracováva signalizáciu medzi zariadeniami ako SIP proxy. Pri zachytení signálu je spustená metóda handle, ktorá skontroluje či sú dáta validné a ak áno spustí metódu processRequest. Táto metóda určí z prijatých dát s pomocou regulárnych výrazov o aký typ signálu ide a spustí zodpovedajúcu metódu na spracovanie(najmä processInvite, processRegister, processAck alebo processNonInvite). Pri registrácii dochádza k dôležitému prvku a to uloženiu kontaktu

¹ <https://github.com/tirfil/PySipFullProxy/blob/master/sipfullproxy.py>

v registri programu. Keďže volať môže len zaregistrované zariadenie, pred volaním je stále potrebná registrácia. Po úspešnej registrácii je poslaná odpoveď vo forme statusu 200. Po registrácii dochádza väčšinou k signálu Invite, ktorý značí pokus o začatie hovoru. Signál je poslaný zariadením číslo 1 na SIP proxy a zo SIP proxy na zariadenie číslo 2. Podobne fungujú aj odpovede, či ostatné signály ako bye, trying, ringing atď. - prvé zariadenie pošle signál na SIP proxy, ktorý následne prepošle signál na druhé zariadenie. ProcessNonInvite slúži najmä na preposielanie ostatných typov správ. Pri pokuse o hovor či preposielaní je kontrolovaná aj správnosť(najmä prítomnosť v registri) odosielateľa či prijímateľa.

4. Zmeny v knižnici a implementácia

V knižnici bolo treba urobiť zmeny, keďže bola implementovaná v skorších verziách Pythonu. Zmeny boli nasledovné:

- Dáta ktoré prichádzali na server bolo potrebné v rámci handleru dekodovať na štandard utf-8
- Keďže sme dáta dekoďovali, pri odosielaní ich bolo potrebné znova zakódovať, pri každom použití metódy sendto, boli teda odosielané dáta zakódované pomocou metódy encode.
- Knižnica používala už vyradenú funkciu pre slovníky dictionary.has_key(key), ktorá zisťovala či sa kľúč nachádza v slovníku. Táto funkcia bola v kóde nahradená s key in dictionary, ktorý dosiahne rovnaký cieľ
- Tvar funkcie string.join(separator, iterable) bol upravený na korektný tvar separator.join(iterable)
- V knižnici bol takisto zakomentovaný blok kódu, ktorý vylučoval použitie privátnych IP adries
- V knižnici boli zmenené texty stavových kódov, ktoré sú posielané z proxy

Z knižnice bolo takisto odstránené(zakomentované) logovanie pomocou knižnice logging. Namiesto toho bol implementovaný vlastný denník hovorov. Táto funkcionálna je implementovaná pomocou funkcií v súbore logs.py, ktoré sú volané v rámci SIP knižnice. Pri pokuse o začatí hovoru je inicializovaná funkcia log_invite, ktorá zapíše poradové číslo hovoru a takisto informácie o pokuse na volanie. Funkcia log_bye naopak zapisuje informácie o ukončení hovoru. Ak má hovor len pokus o začatie, znamená to, že hovor nebol prijatý druhým zariadením. V denníku sú dostupné informácie ako poradové číslo hovoru, čas zavolania, mená a IP adresy účastníkov hovoru, čas zloženie a informáciu o tom, ktorý účastník hovoru hovor ukončil.

SIP proxy je spúšťaná zo súboru main.py, v ktorom sa manuálne nastavuje IP adresa SIP proxy, port, 2 globálne premenné z PySipFullProxy potrebné na jej správne fungovanie a následne je spustený UDPserver, kde argument na handler je práve veľmi dôležitý UDPHandler z knižnice PySipFullProxy, ktorý spracováva signalizáciu.

5. Testovanie

V priečinku sú dostupné pcap súbory na rôzne scenáre, ktoré boli testované(na zobrazenie SIP je potrebné vyfiltrovanie komunikácie):

- Registrácia zariadenia
- Volanie zariadenia so zrušením
- Volanie zariadenia so zdvihnutím a následným zložením z každej strany(dva scenáre)
- Videohovor
- Transfer hovoru(hovor je posunutý na rovnaké zariadenie, kde beží proxy)
- Konferenčný hovor(jedným z účastníkov je zariadenie, na ktorom beží proxy)

Pri scenároch kde boli potrebné aspoň tri zariadenia bolo jedným zo zariadení počítač, na ktorom bežal program so SIP proxy. Z tohto dôvodu môže vyzerat' komunikácia trochu zvlášťne-> komunikácia vyzerá akoby priamo so zariadením, keďže rovnaké zariadenie(teda rovnaká IP) je aj účastníkom hovoru aj SIP proxy.

6. Github

Link na repozitár: