

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE

Fakulta informatiky a informačných technológií

Ilkovičova 2, 842 16 Bratislava 4

Zadanie č. 1 - SIP Proxy (telefónna ústredňa)

MOBILNÉ TECHNOLOGIE A APLIKÁCIE

Róbert Junas

FIIT STU

Cvičenie: streda 12:00

1.3.2022

1. Implementácia

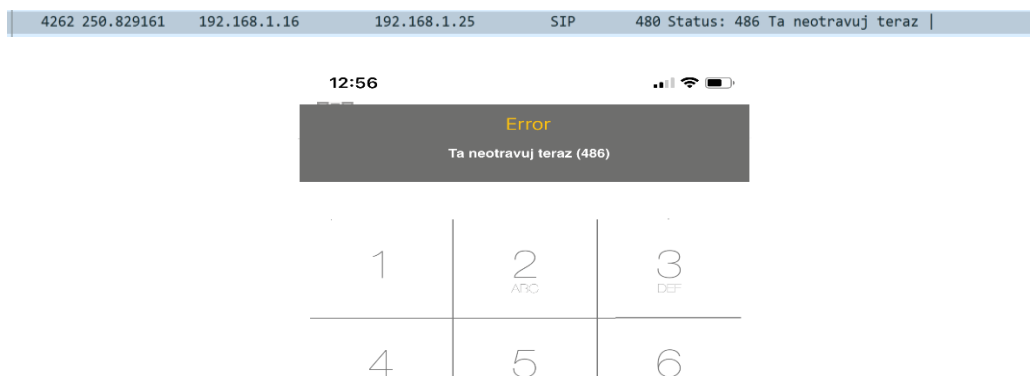
1.1. Použité knižnice

Použili sme UDP implementáciu SIP proxy servera, ktorá bola vytvorená pre python 2.X od THIRION Philippe – presnejšie repozitár PySipFullProxy a súbor sipfullproxy.py (<https://github.com/tirfil/PySipFullProxy>)

1.2. Naša doimplementácia

Na našu implementáciu sme sa rozhodli v jazyku python verzie 3.10.0 64-bit. Kvôli tomu, že knižnica bola implementovaná pre starú verziu python-u, tak sme nad ňou vykonali potrebné úpravy. Ako napr. funkcia dict.has_key(key) nie je podporovaná v 3.X, náhradou je key in dict alebo funkcia string.join(list,str) nie je podporovaná. Namiesto nej sa používa str.join(list). Ďalej socketserver (pôvodne SocketServer → bol nahradený socketserver) nepodporuje posielanie obyčajných reťazcov, ale musia byť správne zakódované a odkódované. Ďalej pôvodná implementácia nepovoľuje adresy zo súkromných sietí 10.0.0.0 a 192.168.0.0, táto podmienka bola odstránená.

Implantovali sme SIPCodeInjector.py, ktorý mení text kódov, ktoré si nastavujeme pri konfigurácii servera. Linphone je implementovaný tak, že spracováva iba kódy a vypíše text, ktorý má sám implementovaný pre 486 to je „<ciel> is busy“, teda nezobrazí upravený text kódu. Za to Zoiper ukáže upravený text.



Implementovali sme vlastný logger – myLogger.py, ktorý vypisuje registrácie, kedy bol, ktorý INVITE poslaný a odmietnutý alebo prijatý. Ak bol prijatý, tak kedy bol hovor položený. Príklad nášho výstupu je v example.log vyzerá nasledovne:

```
>> Logging
02/27/2022, 11:13:47 >> [ REGISTER: 231234@192.168.1.16 ]
02/27/2022, 11:14:04 >> [ REGISTER: 1234@192.168.1.16 ]
02/27/2022, 11:14:05 >> [ REGISTER: 1234@192.168.1.16 ]
02/27/2022, 11:14:05 >> [ REGISTER: 1234@192.168.1.16 ]
02/27/2022, 11:14:20 >> [ REGISTER: 1234@192.168.1.16 ]
02/27/2022, 11:14:26 >> wJSLSnFTWm >> [ INVITE: 231234@192.168.1.16 >>> 1234@192.168.1.16 ]
02/27/2022, 11:14:31 >> wJSLSnFTWm >> [ 200: hovor bol zodvihnuty: 231234@192.168.1.16 <<> 1234@192.168.1.16 ]
02/27/2022, 11:15:03 >> wJSLSnFTWm >> [ BYE: Hovor bol ukonceny: 1234@192.168.1.16 >> 231234@192.168.1.16 ]
```

Posledné sme implementovali Wrapper.py, ktorý obaľuje všetky potrebné veci na spustenie servera do jedného súboru – sipfullproxy.py, SIPCodeInjector.py a myLogger.py. Obsahuje triedu s dvomi funkciami __init__(...) - na inicializáciu potrebných dát a funkcia start() na zapnutie servera.

Tento wrapper sa potom spúšťa cez main.py, kde sú umiestnené aj niektoré konfiguračné premenné ako port, adresa hosta a texty kódov. Program sa spustí pomocou príkazu py main.py v priečinku kde je súbor main.py.

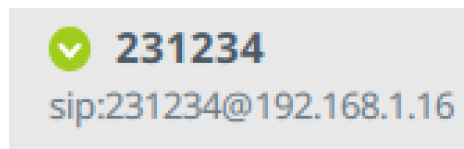
Odkaz na git repozitár: <https://github.com/RobJun/Ustredna>

2. Komunikácie

2.1. Registrácia

V pcap-e registracia.pcapng sa pripojili 3 účty na server, pričom jeden poslal požiadavku na registráciu 2x za sebou.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.1.2	192.168.1.16	SIP	699	Request: REGISTER sip:192.168.1.16 (1 binding)
2	0.002438	192.168.1.16	192.168.1.2	SIP	718	Status: 200 OK (1 binding)
3	16.391829	192.168.1.25	192.168.1.16	SIP	649	Request: REGISTER sip:192.168.1.16 (1 binding)
4	16.392284	192.168.1.25	192.168.1.16	SIP	649	Request: REGISTER sip:192.168.1.16 (1 binding)
5	16.394182	192.168.1.16	192.168.1.25	SIP	669	Status: 200 OK (1 binding)
6	16.396606	192.168.1.16	192.168.1.25	SIP	669	Status: 200 OK (1 binding)
7	30.094925	192.168.1.16	192.168.1.16	SIP	700	Request: REGISTER sip:192.168.1.16 (1 binding)
8	30.096206	192.168.1.16	192.168.1.16	SIP	720	Status: 200 OK (1 binding)

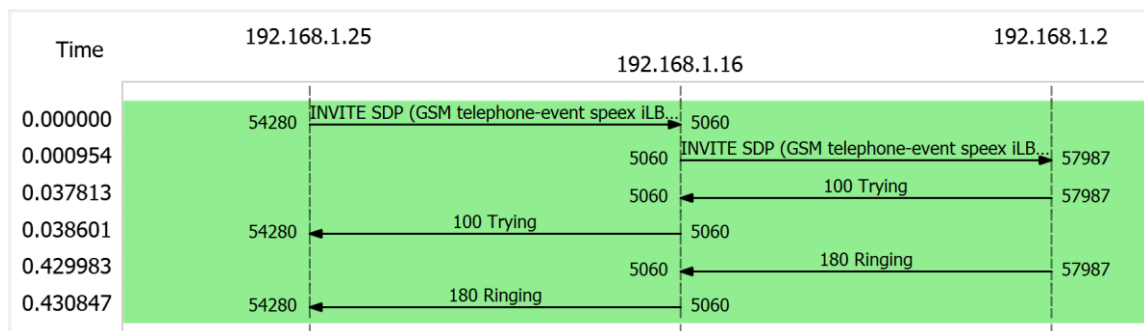


Obrázok 1 Úspešná registrácia v Linphone

2.2. Vytocenie hovoru a zvonenie

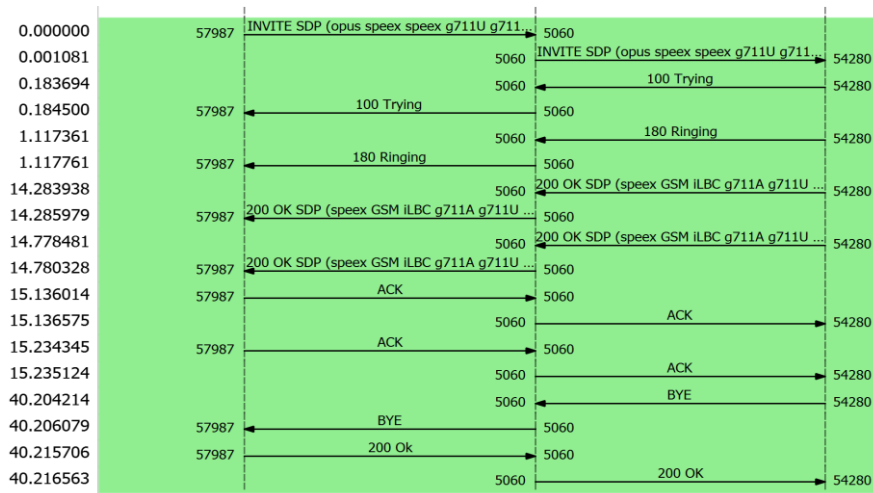
Vytocenie a zvonenie je v pcap-e vytocenie.pcap, kde je vidieť poslanie INVITE requestu serveru, ktorý prepošle INVITE cieľovému klientovi, ktorý pošle Trying a Ringing rámce, ktoré server pošle zdrojovému klientovi.

1	0.000000	192.168.1.25	192.168.1.16	SIP/SDP	851	Request: INVITE sip:robo@192.168.1.16;transport=UDP
2	0.000954	192.168.1.16	192.168.1.2	SIP/SDP	1001	Request: INVITE sip:robo@192.168.1.16;transport=UDP
3	0.037813	192.168.1.2	192.168.1.16	SIP	402	Status: 100 Trying
4	0.038601	192.168.1.16	192.168.1.25	SIP	322	Status: 100 Trying
5	0.429983	192.168.1.2	192.168.1.16	SIP	589	Status: 180 Ringing
6	0.430847	192.168.1.16	192.168.1.25	SIP	509	Status: 180 Ringing

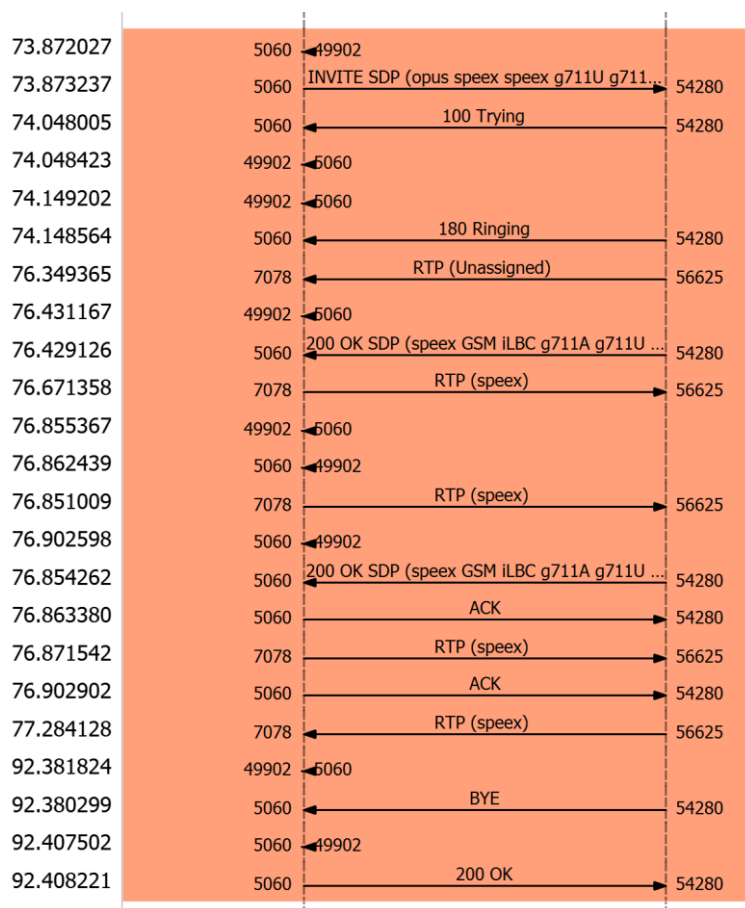


2.3. Prijatie hovoru

Prijatie hovoru je v pcap súbore prijatieHovoru.pcapng. Súbor obsahuje 2 hovory – jeden medzi dvoma zariadeniami a serverom (obrázok 2.) a medzi serverom, na ktorom beží aj klient a iným klientom (obrázok 3.). Dôvod prečo používame dva hovory je, že keď server aj klient beží na rovnakom zariadení, tak ich server zlúči do jedného. Pri dvoch zariadeniach a servery zase nie je možné vidieť RTP komunikáciu.



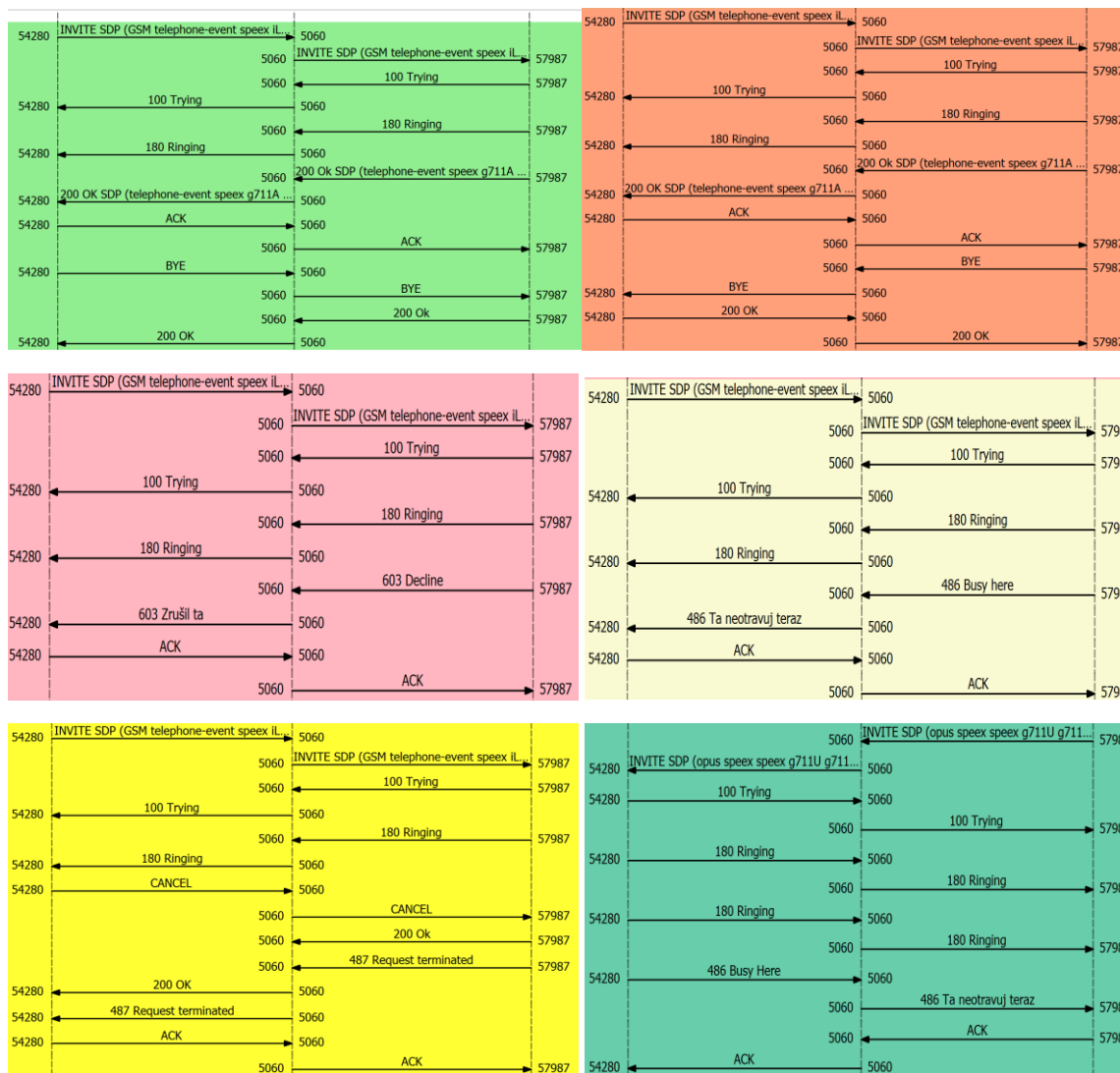
Obrázok 2 prijatie hovoru 1.



Obrázok 3 Prijatie hovoru 2.

2.4. Ukončenie hovoru

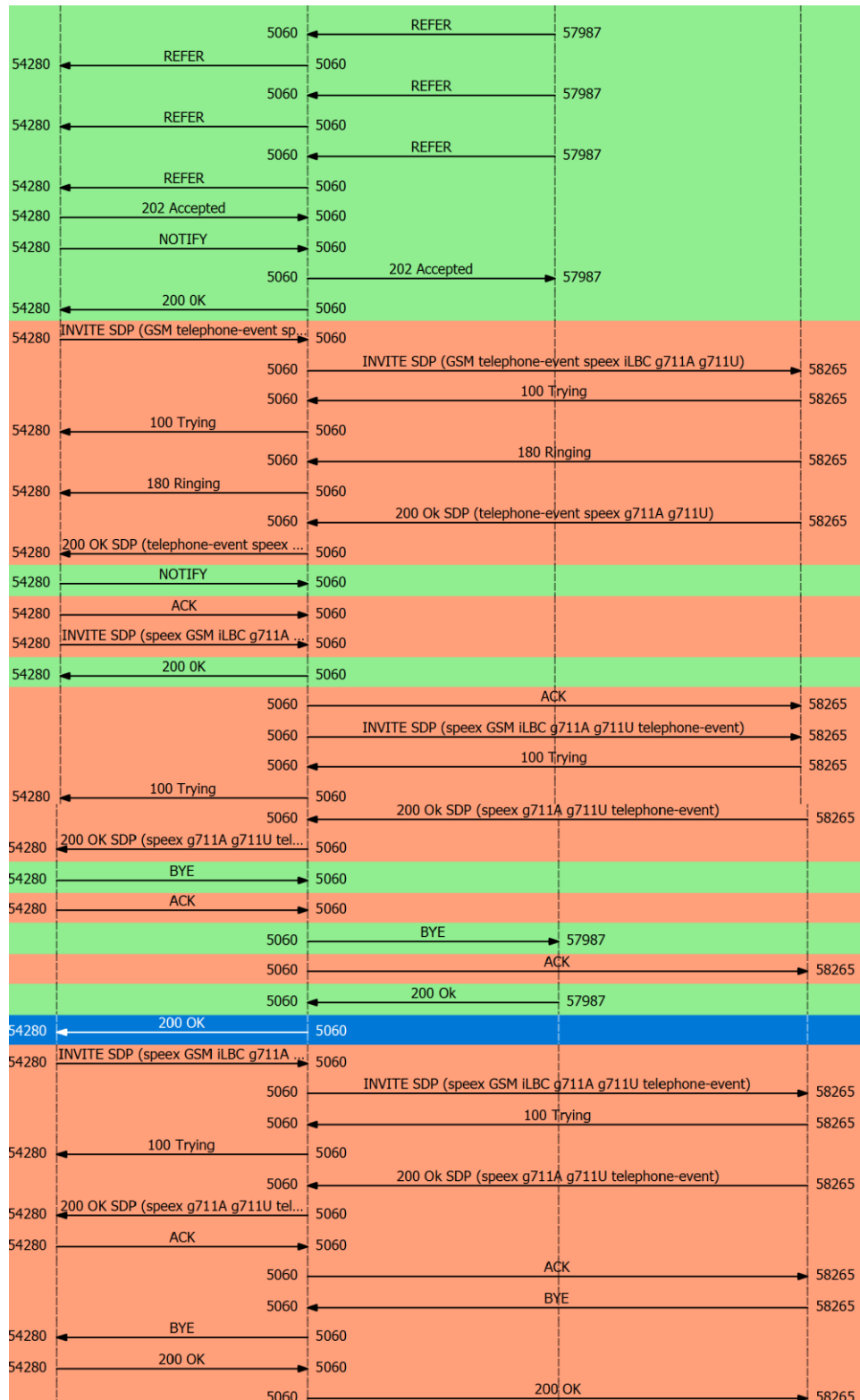
V pcap-e neprijateHovory.pcapng sa nachádzajú ukončenia hovorov, kde zdroj po zodvihnutí zruší hovor, cieľ po zodvihnutí zruší hovor, cieľ ukončí hovor pred zodvihnutím, cieľ nezodvihne a zdroj zruší pred zodvihnutím.



Poznámka: Každá aplikácia má svoje vlastné špecifikácie a čo robia, napr. Zoiper pri zrušení hovoru pred zodvihnutím pošle kód 486, ale Linphone pošle 486, len vtedy, ak sa telefonát nezodvihne, ale ak sa zruší pošle kód 603.

2.5. Presmerovanie

Presmerovanie je v presmerovanie.pcapng, kde jedno zariadenie presmeruje hovor na iné zariadenie pomocou REFER požiadavky.



2.6. Videohovor

Videohovor.pcapng obsahuje rámce sip komunikácie pri zapnutí videohovoru medzi klientami. Ako je vidieť, posielajú sa dva INVITE požiadavky s tým, že na začiatku klienti volajú pomocou audia a ďalší INVITE obsahuje dáta o audio a aj o videu.



2.7. Konferenčný telefón

Vykonané medzi 3 zariadeniami a serverom – pcap konferencia.pcapng. Zariadenia najprv volajú vo dvojici a keď tretie zariadenie zavolá druhému, tak druhé pozastaví telefonát s prvým a začne komunikovať s 3.. Potom druhé stlačí tlačidlo konferencie a iba odzastaví komunikáciu s prvým a zariadenia už komunikujú v konferenčnom režime.

