

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií

Zadanie 1 – SIP Proxy
Jerguš Kapusta

Rok: 2021

Cvičiaci: M. Bahleda

ID: 102980

Predmet: MTAA

Obsah

Zadanie	3
Použité knižnice	3
Spracovanie zdrojového kódu.....	3
Funkcionalita	3
Registrácia účastníka (bez nutnosti autentifikácie)	3
Vytočenie hovoru a zvonenie na druhej strane	4
Priятие hovoru druhou stranou, fungujúci hlasový hovor	4
Ukončenie hlasového hovoru	4
Priятý hovor	4
Odmietnutý hovor.....	4
Konferenčný hovor.....	5
Presmerovanie hovoru.....	5
Video hovor.....	5
Úprava SIP stavových kódov	6
Denník hovorov.....	6
Repozitár a návod k programu.....	7

Zadanie

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

Použité knižnice

Základ SIP Proxy tvorí zdrojový kód sipfullproxy.py vytvorený github používateľom *tirfil*. Zdrojový kód je prevzatý z github repozitára <https://github.com/tirfil/PySipFullProxy>. V danom zdrojovom kóde sa pôvodne využívali viaceré knižnice. Nakoniec boli potrebné len 4: *socketserver* – vďaka nej je realizovaný a spúšťaný samotný UDP server, *re* – pomoc na prácu so stringami, *time* – pomoc pri kontrole validity a *logging* – logovanie hovorov a *socket* – na komunikáciu a posielanie správ medzi zariadeniami.

Spracovanie zdrojového kódu

Pôvodný program sipfullproxy.py bol napísaný v pythone2. Ja som na implementáciu zadania použil python3, čiže bolo treba prerobiť / opraviť niektoré časti kódu. Niektoré boli menej závažné: Knižnicu *SocketServer* som premenoval na *socketserver*. V niektorých podmienkach som zmenil syntax z pôvodného `x.has_key(y)` na `y in x`.

Pôvodný program tiež pracoval s niektorými dátami typu `byte` ako so `stringom`, čo Py3 nepodporuje. Bolo nutné robiť kódovanie a dekódovanie dát. Pretože dáta, ktoré boli zachytené boli typu `byte`. Tiež som zakomentoval niektoré nepotrebné podmienky, ako napríklad tie, ktoré brali adresy 192.168... ako invalidné. Po týchto ošetreniach bolo možné spustiť SIP Proxy z pôvodného súboru a zavolať si medzi zariadeniami.

Pre spúšťanie servera z vlastného programu som vytvoril súbor `mySipProxy.py`, kde je vytvorená jednoduchá `main` funkcia a importované potrebné knižnice.

Okrem nevyhnutného som v pôvodnom zdrojovom kóde zakomentoval nepotrebné výpisy a vytvoril som si vlastné, na logovanie denníku hovorov s pomocou vlastných funkcií. Tiež som zmenil niektoré SIP stavové kódy.

Funkcionalita

Po vyššie popísanom spracovaní zdrojového kódu bola funkčná všetka základná požadovaná funkcionalita zadanie (registrácia, vytočenie, prijatie, priebeh a ukončenie hovoru, konferenčný hovor, video hovor, presmerovanie hovorov) Jednotlivé body nebolo nutné riešiť samostatne. Kód funguje pomocou triedy *UDPHandler*. Tá zachytáva prijaté správy cez metódu *handle()* a pošle ich na roztriedenie do funkcie *processRequest()*. Podľa toho o akú SIP metódu ide sa správa pošle na ďalšie spracovanie. Odpovede má na starosť metóda *sendResponse()*, ktorá spracuje dáta a pošle ich smer klient.

Registrácia účastníka (bez nutnosti autentifikácie)

Záznam registrácie účastníka sa nachádza v súbore `Z1_register.pcapng`. Registrácia prebieha bez autentifikácie. Proces je jednoduchý – klient pošle SIP segment s metódou `register` smerom na server. Ten mu odpovie stavovým kódom 200. V samotnom kóde to vyzerá tak, že z *processRequest()* ide správa do

metódy *processRegister()*. Tam sa po spracovaní uloží kontakt s adresou do globálnej premennej *registrar* a následne sa pošle odpoveď s kódom 200 cez *sendResponse()*.

Time	Source	Destination	Port	Protocol	Length	Info
14 5.404673	192.168.1.15	192.168.1.14	5060	SIP	676	Request: REGISTER sip:192.168.1.14:5060 (1 binding)
15 5.405222	192.168.1.14	192.168.1.15	39858	SIP	700	Status: 200 OK - pohoda (REGISTER) (1 binding)

Vytočenie hovoru a zvonenie na druhej strane

Záznam vytočenia hovoru sa nachádza v súbore *Z1_ringing.pcapng*. Volajúci klient pošle požiadavku INVITE smerom na server a s adresou volaného. Server ju prepošle volanému. Volaný odpovedá cez server správou s kódom 100, ktorý indikuje, že požiadavka sa spracováva. Následne pošle ešte jednu správu cez server. Teraz s kódom 180, ktorá oznamuje, že volaný dostal INVITE a hovor už "zvoní". Čo sa týka programu INVITE sa spracuje v metóde *processInvite()*. Skontroluje sa, či je pôvodca a cieľ zaregistrovaný (ich adresa je v *registrar*), upraví sa správa a pošle sa ďalej. Správa s kódom sa spracujú v metóde *processCode()*, kde sa skontroluje pôvod a prepošle sa ďalej.

8 0.187573	192.168.1.15	192.168.1.14	5060	SIP/SDP	146	Request: INVITE sip:tablet@192.168.1.14:5060
10 0.187966	192.168.1.14	192.168.1.19	53373	SIP/SDP	278	Request: INVITE sip:tablet@192.168.1.14:5060
11 0.342599	192.168.1.19	192.168.1.14	5060	SIP	338	Status: 100 Trying
12 0.342871	192.168.1.14	192.168.1.15	39858	SIP	276	Status: 100 Trying
13 0.692835	192.168.1.19	192.168.1.14	5060	SIP	529	Status: 180 Ringing
14 0.693172	192.168.1.14	192.168.1.15	39858	SIP	467	Status: 180 Ringing

Prijatie hovoru druhou stranou, fungujúci hlasový hovor

Záznam prijatia hovoru sa nachádza v súbore *Z1_pickup-call.pcapng*. Ak volaný po spracovaní INVITE hovor zdvihne, pošle správu serveru a ten zase volajúcemu s kódom 200, ktorá potvrdzuje požiadavku INVITE. Ešte pošle volajúci volanému oboznámenie ACK cez server a hovor začne. Počas hovoru si posielajú RTP segmenty, kde sú zaznamenané dáta z hovoru. ACK správy sa spracujú v *processACK()* a pošlú sa ďalej.

1 0.000000	192.168.1.14	192.168.1.14	5060	SIP/SDP	1204	Status: 200 Ok (INVITE)
2 0.000624	192.168.1.14	192.168.1.15	39858	SIP/SDP	1152	Status: 200 Ok (INVITE)
8 0.415556	192.168.1.14	192.168.1.14	63033	SIP	528	Request: ACK sip:linPC@192.168.1.14:63033;transport=udp
16 0.415127	192.168.1.15	192.168.1.14	5060	SIP	441	Request: ACK sip:linPC@192.168.1.14:63033;transport=udp

Ukončenie hlasového hovoru

Prijatý hovor

Záznam ukončenia prijatého hovoru sa nachádza v súbore *Z1_hangup.pcapng*. Ukončenie prebiehajúceho hovoru začne vyslaním požiadavky BYE cez server, ktorý správu prepošle. Prijímajúca správa odpovie potvrdením s kódom 200. Tým sa hovor ukončí.

234 2.003729	192.168.1.14	192.168.1.14	63033	SIP	528	Request: BYE sip:linPC@192.168.1.14:63033;transport=udp
236 2.025484	192.168.1.14	192.168.1.14	5060	SIP	506	Status: 200 Ok (BYE)
237 2.003412	192.168.1.15	192.168.1.14	5060	SIP	441	Request: BYE sip:linPC@192.168.1.14:63033;transport=udp
239 2.025918	192.168.1.14	192.168.1.15	39858	SIP	454	Status: 200 Ok (BYE)

Odmietnutý hovor

Záznam odmietnutia hovoru sa nachádza v súbore *Z1_decline.pcapng*. Odmietnutie hovoru prebieha podobne ako prijatie, ale miesto správy s kódom 200 ako odpoveď na INVITE príde správa 603, ktorý oznámi, že hovor bol odmietnutý. Následne ešte prebehne potvrdenie odmietnutia cez ACK správu.

9 0.466054	192.168.1.14	192.168.1.15	39858	SIP	467 Status: 180 Ringing
12 2.631483	192.168.1.19	192.168.1.14	5060	SIP	487 Status: 603 Decline
13 2.631954	192.168.1.14	192.168.1.15	39858	SIP	425 Status: 603 Decline
14 2.658097	192.168.1.15	192.168.1.14	5060	SIP	448 Request: ACK sip:tablet@192.168.1.14:5060
15 2.658358	192.168.1.14	192.168.1.19	53373	SIP	580 Request: ACK sip:tablet@192.168.1.14:5060

Konferenčný hovor

Záznam konferenčného hovoru sa nachádza v súbore Z1_conferencecall.pcapng. Konferenčný hovor začína podobne ako hovor medzi dvoma účastníkmi. Rozdiel je v tom, že ten, čo vytvára hovor pošle INVITE viacerým účastníkom konferenčného hovoru. Po tom čo s nimi začne hovor potvrdením s kódom 200 a ACK, pripoja sa do konferencie pomocou požiadavky SUBSCRIBE, ktorá ja tiež potvrdená cez kód 200. Ukončenie konferenčného hovoru začína rovnako ako u bežného hovoru cez správy BYE a kód 200. Okrem toho pošlú účastníci hovoru, ktorých zavola tvorca konferencie ďalšiu požiadavku SUBSCRIBE na odpojenie z hovoru. Tá sa tiež potvrdzuje cez správu s kódom 200.

250 12.224758	192.168.1.15	192.168.1.14	5060	SIP/SDP	1215 Status: 200 Ok (INVITE)
259 12.457491	192.168.1.14	192.168.1.14	5060	SIP	433 Request: ACK sip:mobilin@192.168.1.15:39858;transport=udp
292 12.457765	192.168.1.14	192.168.1.15	39858	SIP	540 Request: ACK sip:mobilin@192.168.1.15:39858;transport=udp
364 12.881163	192.168.1.15	192.168.1.14	5060	SIP	671 Request: SUBSCRIBE sip:linPC@192.168.1.14;conf-id=r5rVr;transport=udp
365 12.881488	192.168.1.14	192.168.1.15	39858	SIP	665 Status: 200 OK - pohoda (SUBSCRIBE)
4515 33.457020	192.168.1.14	192.168.1.14	5060	SIP	433 Request: BYE sip:mobilin@192.168.1.15:39858;transport=udp
4516 33.482107	192.168.1.14	192.168.1.14	5060	SIP	429 Request: BYE sip:tablet@192.168.1.19:53373;transport=udp
4517 33.501926	192.168.1.14	192.168.1.14	63033	SIP	445 Status: 200 Ok (BYE)
4518 33.515472	192.168.1.14	192.168.1.14	63033	SIP	446 Status: 200 Ok (BYE)
4529 33.457279	192.168.1.14	192.168.1.15	39858	SIP	540 Request: BYE sip:mobilin@192.168.1.15:39858;transport=udp
4532 33.482760	192.168.1.14	192.168.1.19	53373	SIP	536 Request: BYE sip:tablet@192.168.1.19:53373;transport=udp
4537 33.501382	192.168.1.15	192.168.1.14	5060	SIP	517 Status: 200 Ok (BYE)
4538 33.515056	192.168.1.19	192.168.1.14	5060	SIP	518 Status: 200 Ok (BYE)
4539 33.879308	192.168.1.19	192.168.1.14	5060	SIP	672 Request: SUBSCRIBE sip:tablet@192.168.1.19:53373;transport=udp, in-dialog
4540 33.879602	192.168.1.14	192.168.1.19	53373	SIP	662 Status: 200 OK - pohoda (SUBSCRIBE)
4541 34.048446	192.168.1.15	192.168.1.14	5060	SIP	675 Request: SUBSCRIBE sip:mobilin@192.168.1.15:39858;transport=udp, in-dialog
4542 34.048796	192.168.1.14	192.168.1.15	39858	SIP	663 Status: 200 OK - pohoda (SUBSCRIBE)

Presmerovanie hovoru

Záznam presmerovania hovoru sa nachádza v súbore Z1_transfercall.pcapng. Po spustení hovoru môže volajúci A presmerovať hovor na niekoho iného (volajúci C), tak že pošle požiadavku REFER smerom na server s údajmi účastníka C. Server požiadavku akceptuje správou s kódom 202, ktorú pošle späť volávúcemu A. Server potom pošle INVITE volávúcemu C, ten pošle naspäť kód 100. Server potom pošle NOTIFY volávúcemu A, čím potvrdí, že sa požiadavka spracúva. Ten pošle naspäť na server kód 200 OK. Tento postup s NOTIFY sa opakuje, ale namiesto kódu 100 Trying ho nahradí správa s kódom 180 Ringing. Začiatok presmerovanej komunikácie sa potvrdí ACK správami.

Video hovor

Záznam video-hovoru sa nachádza v súbore Z1_videocall.pcapng. Video hovor sa môže uskutočniť počas obyčajného hovoru, tak že jeden z účastníkov o to požiada. Požiadať a začatie video hovoru sa robí cez požiadavku INVITE s atribútom Subject s hodnotou „Media change“ a v Media Description je Media Type: Video. Po tom čo príde na server INVITE, pošle sa správa s kódom 100 Trying ako odpoveď, že sa požiadavka spracúva. Prijatie požiadavky sa potvrdí správou 200 OK z požadovaného na server a odtiaľ na iniciátora video hovoru. Zrušenie video hovoru je rovnaký. Rozdiel je v tom, že INVITE má Media Attribute s hodnotou „inactive“.

> Media Description, name and address (m): video 0 RTP/AVP 0
Media Attribute (a): inactive
[Generated Call-ID: F7CIR0IXjd]

162 4.456561	192.168.1.14	192.168.1.14	5060 SIP	341 Status: 100 Trying
164 4.449865	192.168.1.15	192.168.1.14	5060 SIP/SDP	215 Request: INVITE sip:linPC@192.168.1.14:63033;transport=udp, in-dialog
167 4.456986	192.168.1.14	192.168.1.15	39858 SIP	289 Status: 100 Trying
345 6.003842	192.168.1.14	192.168.1.14	5060 SIP/SDP	1392 Status: 200 Ok (INVITE)
346 6.076851	192.168.1.14	192.168.1.14	63033 SIP	528 Request: ACK sip:linPC@192.168.1.14:63033;transport=udp
352 6.004331	192.168.1.14	192.168.1.15	39858 SIP/SDP	1340 Status: 200 Ok (INVITE)
365 6.076538	192.168.1.15	192.168.1.14	5060 SIP	441 Request: ACK sip:linPC@192.168.1.14:63033;transport=udp
1309 11.396080	192.168.1.14	192.168.1.15	39858 SIP/SDP	1298 Request: INVITE sip:mobilin@192.168.1.15:39858;transport=udp, in-dialog
1330 11.447252	192.168.1.15	192.168.1.14	5060 SIP	352 Status: 100 Trying
1340 11.454232	192.168.1.15	192.168.1.14	5060 SIP/SDP	1233 Status: 200 Ok (INVITE)

Úprava SIP stavových kódov

Správy som zmenil tak, že som zmenil vstupný parameter funkcie `sendResponse()` pri jednotlivých prípadoch. Napríklad zmena správy 200 OK je vidieť aj vo Wiresharku. Alebo kód 400 je vidieť na zariadení, keď bol server reštartovaný, klient si to neuvedomil a chce niekomu zavolať.

```
self.sendResponse("200 OK - pohoda")
```

```
self.sendResponse("400 Ajaj, zla poziadavka")
```

Denník hovorov

Záznamy sa zapisujú do súboru dennik-hovorov.log vo formáte: „{dátum} {čas}: {správa}“ Pri každej správe je zaznamenané tiež ID hovoru. Do denníka sa zapisujú logy úrovne WARNING. Logované sú začiatky hovorov, kto komu volá, či a kto hovor zdvihol, požiadavky na video hovor, začatie alebo ukončenie video hovoru a ukončenie hovoru. Na pomoc pri logovaní som vytvoril nové funkcie: `cleanName(msg)` – vráti adresu v prehľadnejšom stave, `getCallID(data)` – vráti ID hovoru, `checkVideo(data)` - kontrola, či sa správa týka videa a či je to požiadavka o vypnutie alebo zapnutie videa, `checkNewCall(callID)` – ošetruje logovanie duplikovaných správ. Analýza prijatých správ sa deje v metóde `processInvite()` a `processCode()`. Tam kontrolujem o aký typ kódu respektíve SIP metódy ide. Popríade, či bolo požiadané a zmenu média a podobne. Podľa toho sa zapíšu príslušné správy do denníka hovorov.

```
2022-02-26 21:28:30,985: Incoming Call From: sip:mobilin@192.168.1.14 >> To: sip:tablet@192.168.1.14 (Call-ID: ruR~1yA-g9)
2022-02-26 21:28:33,597: Call Declined sip:tablet@192.168.1.14 (Call-ID: ruR~1yA-g9)
2022-02-26 21:28:45,209: Incoming Call From: sip:tablet@192.168.1.14 >> To: sip:mobilin@192.168.1.14 (Call-ID: x70HabYg8p)
2022-02-26 21:28:45,318: Call Answered by sip:mobilin@192.168.1.14 (Call-ID: x70HabYg8p)
2022-02-26 21:28:47,097: Requested Video Call ON From: sip:mobilin@192.168.1.14 >> To: sip:tablet@192.168.1.14 (Call-ID: x70HabYg8p)
2022-02-26 21:28:48,875: Video Call ON (Call-ID: x70HabYg8p)
2022-02-26 21:28:52,923: Requested Video Call OFF From: sip:mobilin@192.168.1.14 >> To: sip:tablet@192.168.1.14 (Call-ID: x70HabYg8p)
2022-02-26 21:28:52,977: Video Call OFF (Call-ID: x70HabYg8p)
2022-02-26 21:28:59,136: Requested Video Call ON From: sip:mobilin@192.168.1.14 >> To: sip:tablet@192.168.1.14 (Call-ID: x70HabYg8p)
2022-02-26 21:29:00,892: Video Call OFF (Call-ID: x70HabYg8p)
2022-02-26 21:29:03,762: Call Has Ended (Call-ID: x70HabYg8p)
```

```
fromm = data[4].split(';')[0].replace('<','').replace('>','')
to = data[5].split(';')[0].replace('<','').replace('>','')
callID = getCallID(data)
if data[9].find("Media change") > -1:
    if data[-2].find("inactive") > -1:
        logging.warning(f"\tRequested Video Call OFF {fromm} >> {to} ({callID})")
    else:
        logging.warning(f"\tRequested Video Call ON {fromm} >> {to} ({callID})")
elif checkNewCall(getCallID(data)) == 1:
    calls[callID] = 1
    logging.warning(f"\tIncoming Call {fromm} >> {to} ({callID})")
```

Repozitár a návod k programu

Zdrojový kód a ostatné súbory sú v repozitári na adrese https://github.com/jkloris/MTAA_SIP_proxy.git. V priečinku pcaps sú všetky záznamy komunikácie vrátane ešte nespomenutého Z1_fullcall.pcapng, ktorý obsahuje jeden celý hovor.

Na spustenie programu sú dôležité 2 súbory: *sipfullproxy.py* a *mySipProxy.py*. Tieto 2 súbory musia byť v jednom adresári pri spúšťaní. Pre zapnutie serveru treba zavolať príkaz *py mySipProxy.py*. Mala by sa vypísať správa o zapnutí servera a na akom zariadení a IP adrese beží. Od tohto momentu môže začať komunikácia.

```
C:\Users\Lenovo T470\PycharmProjects\learning\MTAA_SIPProxy>py mySipProxy.py
SIP PROXY server DESKTOP-VPB1560 is running on 192.168.1.14
```