Slovenská technická univerzita v Bratislave

Fakulta informatiky a informačných technológií

Zadanie 1 – SIP Proxy

Mobilné technológie a aplikácie 2021/2022

Marek Chrappa

Hlavná myšlienka zadania:

Na vašom počítači (alebo virtuálnom počítači) sprevádzkujte 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 funkcionalitu 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 dvomi fyzickými zariadeniami v rámci LAN siete.

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)
- 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
- Možnosť realizovať
- Logovanie "denníka hovorov" – kto kedy komu volal, kedy bol ktorý hovor prijatý, kedy bol ktorý hovor ukončený, do ľubovoľného textového súboru v ľubovoľnom
- Úprava SIP stavových kódov z zdrojovom kóde proxy, napr. "486 Busy Here" zmeníte na "486 Obsadené"

Implementácia

Ako knižnicu som využil knižnicu PySipFullProxy. Ako prvé treba vytvoriť a naconfigurovať logovací súbor proxy.log. Potom program získa lokálnú IP adresu na ktorej bude prevádzkovať server ktorý bude mať na starosti počúvanie na porte. Treba taktiež nastaviť port. A na koniec sa spustí serve.forever() vdaka ktorému bude server počúvať dookola.

https://github.com/tirfil/PySipFullProxy

Hovor, video hovor, presmerovanie hovoru a konferenčný hovor boli všetky dobre implementované v danej knižnici.

Denník hovorov

```
Call:
telefon@192.168.0.106 calling PC@192.168.0.106
Zvoni
Hovor prijaty 03-01-2022 17:18:42
Hovor ukonceny 03-01-2022 17:18:47

Call:
PC@192.168.0.106 calling telefon@192.168.0.106
Zvoni
Hovor prijaty 03-01-2022 17:18:52
Hovor ukonceny 03-01-2022 17:19:03

Call:
PC@192.168.0.106 calling telefon@192.168.0.106
Zvoni
Hovor odmietnuty 03-01-2022 17:19:09

Call:
PC@192.168.0.106 calling telefon@192.168.0.106
Zvoni
Hovor prijaty 03-01-2022 17:19:58
Hovor ukonceny 03-01-2022 17:20:04

Call:
telefon@192.168.0.106 calling PC@192.168.0.106
Zvoni
Hovor prijaty 03-01-2022 17:20:08
Hovor ukonceny 03-01-2022 17:20:20

Call:
telefon@192.168.0.106 calling PC@192.168.0.106
Zvoni
Hovor odmietnuty 03-01-2022 17:20:33


```

Jednoduchý .txt ktorý je výsledkom počúvania SIP správ podla ktorých zapisuje informácie do zoznamu. V zozname hovorov sa nachádza vždy informácia že kto chce volať komu a následne či sa informácia dobre posunula. Dalej sa zapíše informácia či druhá strana prijala hovor alebo ho odmietla aj s časom. A v prípade že telefonat bol prijatý tak v momente kedy niekto z účastníkov hovoru hovor položí tak sa zapíše čas do denníka hovorov.

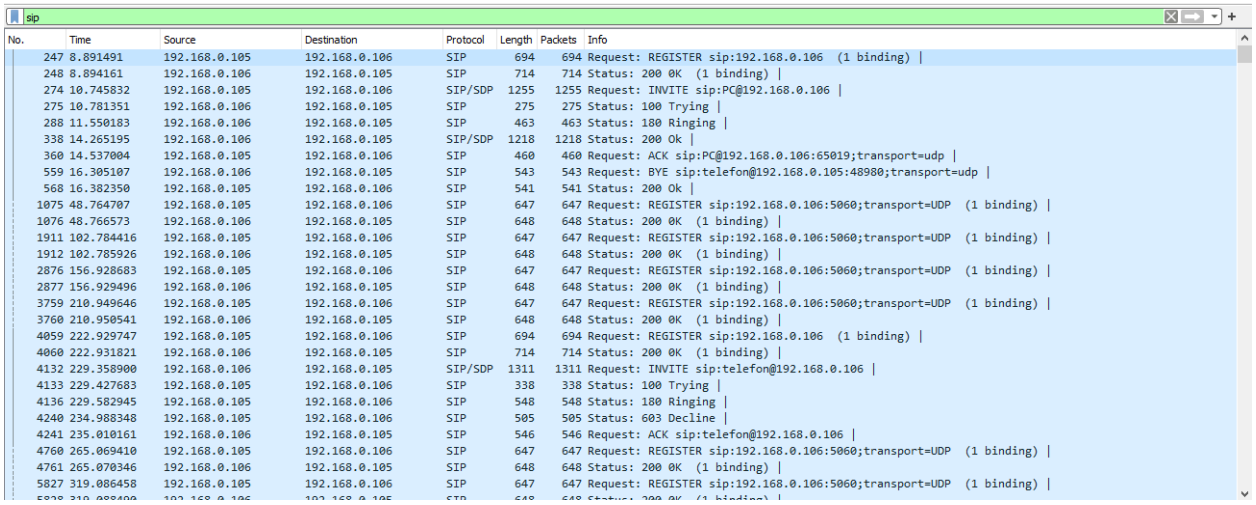
Napr. v kode:

```
elif rx_bye.search(request_uri):
    self.dennik("Hovor ukonceny " + currTime + "\n")
    self.processNonInvite()
```

Sip stavové správy

Program odchytáva SIP správy ktoré potom program prepíše a zapíšeme do proxy.log

```
currTime = time.strftime("%m-%d-%Y %H:%M:%S", time.localtime())
if data[0] == 'SIP/2.0 100 Trying':
    logging.info("<<< %s" % 'SIP/2.0 100 Pokus o nadviazanie')
elif data[0] == 'SIP/2.0 603 Decline':
    self.dennik("Hovor odmietnuty " + currTime + "\n")
elif data[0] == 'SIP/2.0 200 Ok':
    if 'INVITE' in data[5]:
        self.dennik("Hovor prijaty " + currTime)
elif data[0] == 'SIP/2.0 180 Ringing':
    self.dennik("Zvoni")
    logging.info("<<< %s" % 'SIP/2.0 180 Zvoni')
```



sip

No.	Time	Source	Destination	Protocol	Length	Packets	Info
247	8.891491	192.168.0.105	192.168.0.106	SIP	694	694	Request: REGISTER sip:192.168.0.106 (1 binding) |
248	8.894161	192.168.0.106	192.168.0.105	SIP	714	714	Status: 200 OK (1 binding) |
274	10.745832	192.168.0.105	192.168.0.106	SIP/SDP	1255	1255	Request: INVITE sip:PC@192.168.0.106 |
275	10.781351	192.168.0.106	192.168.0.105	SIP	275	275	Status: 100 Trying |
288	11.550183	192.168.0.106	192.168.0.105	SIP	463	463	Status: 180 Ringing |
338	14.265195	192.168.0.106	192.168.0.105	SIP/SDP	1218	1218	Status: 200 Ok |
360	14.537004	192.168.0.105	192.168.0.106	SIP	460	460	Request: ACK sip:PC@192.168.0.106:65019;transport=udp |
559	16.305107	192.168.0.106	192.168.0.105	SIP	543	543	Request: BYE sip:telefon@192.168.0.105:48980;transport=udp |
568	16.382350	192.168.0.105	192.168.0.106	SIP	541	541	Status: 200 Ok |
1075	48.764707	192.168.0.105	192.168.0.106	SIP	647	647	Request: REGISTER sip:192.168.0.106:5060;transport=UDP (1 binding) |
1076	48.766573	192.168.0.106	192.168.0.105	SIP	648	648	Status: 200 OK (1 binding) |
1911	102.784416	192.168.0.105	192.168.0.106	SIP	647	647	Request: REGISTER sip:192.168.0.106:5060;transport=UDP (1 binding) |
1912	102.785926	192.168.0.106	192.168.0.105	SIP	648	648	Status: 200 OK (1 binding) |
2876	156.928683	192.168.0.105	192.168.0.106	SIP	647	647	Request: REGISTER sip:192.168.0.106:5060;transport=UDP (1 binding) |
2877	156.929496	192.168.0.106	192.168.0.105	SIP	648	648	Status: 200 OK (1 binding) |
3759	210.949646	192.168.0.105	192.168.0.106	SIP	647	647	Request: REGISTER sip:192.168.0.106:5060;transport=UDP (1 binding) |
3760	210.950541	192.168.0.106	192.168.0.105	SIP	648	648	Status: 200 OK (1 binding) |
4059	222.929747	192.168.0.105	192.168.0.106	SIP	694	694	Request: REGISTER sip:192.168.0.106 (1 binding) |
4060	222.931821	192.168.0.106	192.168.0.105	SIP	714	714	Status: 200 OK (1 binding) |
4132	229.358900	192.168.0.106	192.168.0.105	SIP/SDP	1311	1311	Request: INVITE sip:telefon@192.168.0.106 |
4133	229.427683	192.168.0.105	192.168.0.106	SIP	338	338	Status: 100 Trying |
4136	229.582945	192.168.0.105	192.168.0.106	SIP	548	548	Status: 180 Ringing |
4240	234.988348	192.168.0.105	192.168.0.106	SIP	505	505	Status: 603 Decline |
4241	235.010161	192.168.0.106	192.168.0.105	SIP	546	546	Request: ACK sip:telefon@192.168.0.106 |
4760	265.069410	192.168.0.105	192.168.0.106	SIP	647	647	Request: REGISTER sip:192.168.0.106:5060;transport=UDP (1 binding) |
4761	265.070346	192.168.0.106	192.168.0.105	SIP	648	648	Status: 200 OK (1 binding) |
5827	319.086458	192.168.0.105	192.168.0.106	SIP	647	647	Request: REGISTER sip:192.168.0.106:5060;transport=UDP (1 binding) |

Github

GitHub kodu - https://github.com/MarekChrappa/MTAA