

Slovenská Technická Univerzita

Fakulta informatiky a informačných technológií

Ilkovičova 3, 842 19 Bratislava 4

Richard Szarka

SIP PROXY

Prednášajúci: prof. Ing. Ivan Kotuliak, PhD., Ing. Marek Galinský

Cvičiaci: Ing. Marek Galinský

Cvičenie: Streda 12:00

Obsah

| | |
|-----------------------------------------------------------------|----|
| 1 Zadanie | 3 |
| 2 Prevzatá knižnica | 3 |
| 3 Implementácia s knižnicou..... | 3 |
| 3.1 Základné funkcionality..... | 4 |
| 3.2 Nepovinné funkcionality, ktoré knižnica implementuje..... | 4 |
| 3.3 Nepovinné funkcionality, ktoré knižnica neimplementuje..... | 4 |
| 3.3.1 Vytvorenie logovania hovor | 4 |
| 3.3.2 Úprava niekoľkých SIP stavových kódov | 5 |
| 4 Wireshark..... | 6 |
| 4.1 Základné funkcionality a video hovor | 6 |
| 4.2 Presmerovaný hovor | 12 |
| 4.3 Konferenčný hovor | 14 |
| 5 Záver | 15 |

1 Zadanie

Na vašom počítači (alebo virtuálnom počítači) sprevádzajte SIP Proxy, ktorá umožní prepájanie a realizáciu hovorov medzi štandardnými SIP klientami. 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 funkcionality 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 dvoma fyzickými zariadeniami v rámci LAN siete.

2 Prevzatá knižnica

Prevzatú knižnicu použitú v mojej implementácii mám z git repozitára <https://github.com/tirfil/PySipFullProxy>. Kód je napísaný ešte v python 2, čiže kód potreboval veľa úprav hlavne čo sa týkalo práce so string-ami, slovníkmi ale aj objektov typu bytes. Knižnica používa knižnicu socketserver, ktorá slúži s implementáciou triedy `class UDPHandler` na to, aby SIP proxy server spracovával požiadavky donekonečna. Princíp fungovania je taký, že `class UDPHandler` obsahuje metódu `handle(self)`, ktorá spracuje danú požiadavku (ktorá prišla na server) a pošle ju funkcii metóde `processRequest(self)`. Tá už podľa druhu požiadavky na server prepošle danú požiadavku (request) do správnej metódy spracovania (`processNonInvite(self)`, `processAck(self)`, `processInvite(self)`, `def processRegister(self)` alebo len obyčajný `self.sendResponse("200 OK")`).

Link na môj repozitár: https://github.com/RichardSzarka/SIP_PROXY

3 Implementácia s knižnicou

Kód v mojej main funkcii je nasledovný:

```
if __name__ == "__main__":  
    try:  
        HOST = input("Zadajte IP proxy servera: ")  
        PORT = int(input("Zadajte PORT na server: "))  
  
        siplib.recordroute = "Record-Route: <sip:%s:%d;lr>" % (HOST, PORT)  
        siplib.topvia = "Via: SIP/2.0/UDP %s:%d" % (HOST, PORT)  
        siplib.topvia = bytes(siplib.topvia, "utf8")  
  
        server = siplib.socketserver.UDPServer((HOST, PORT),  
        siplib.UDPHandler)  
    except:  
        print("Socket binding sa nepodaril (asi sa port používa alebo ste  
        zadali zlu IP)")  
        exit(1)  
  
    try:  
        print("Server beží")  
        server.serve_forever()  
    except:  
        print("Server sa vypoľ")
```

V podstate vypýtam od užívateľa na akej IP adrese a porte chce aby daný server pracoval a následne sa nabinď-ujú všetky potrebné veci pre daný server a server sa spustí. Ak užívateľ zadá zlú IP adresu alebo port, ktorý sa už používa, tak sa mu vráti chybová hláška. Zadávanie IP adresy je manuálne z dôvodu, že na mojom počítači (ak sa išlo cez WLAN pripojenie), zvyklo zobrať nesprávnu IP adresu čo zaručovalo nesprávne fungovanie servera (napríklad zobralo IP adresu VMware). Taktiež úprava knižnice spočívala aj v implementácii jednotlivých nepovinných požiadaviek.

3.1 Základné funkcionality

Prevzatá knižnica implementuje všetky základné funkcionality

3.2 Nepovinné funkcionality, ktoré knižnica implementuje

Z nepovinných funkcionalít knižnica implementuje:

- Videohovor
- Možnosť presmerovania hovoru
- Konferenčný hovor.

3.3 Nepovinné funkcionality, ktoré knižnica neimplementuje

Z nepovinných funkcionalít som do knižnice dorábal:

- Vytvorenie logovania hovoru
- Úpravu niekoľkých SIP stavových správ

3.3.1 Vytvorenie logovania hovoru

Logovanie hovorov realizuje pomocou funkcie `make_log(data)`, a pomocou triedy `Call`. Princíp je taký, že je v programe udržiavané pole bežiacich a zvoniacich hovorov. Ak sa identifikuje, že jeden z hovorov v danom poli skončil, alebo, že zvoniaci hovor bol nezodvihnutý, zapíše sa správa o priebehu daného hovoru do súboru `log.txt` a daný hovor sa z poľa odstráni. Párovanie nastáva pomocou ID jednotlivých hovorov. Trieda `Call` vyzerá nasledovne:

```
class Call:
    def __init__(self, data):
        for info in data:
            if rx_from.search(info):
                self.From = str(rx_uri.search(info).group(), "utf8")
            elif rx_to.search(info):
                self.to = str(rx_uri.search(info).group(), "utf8")
            elif rx_call_id.search(info):
                self.call_id = info

        self.ringing_time = ""
        self.start_time = ""
        self.end_time = ""
```

Premenná `data` obsahuje pole jednotlivých parametrov SIP hlavičky. `From`, `to` a `Call ID` avšak nie sú v protokole pevne na konkrétnych miestach, preto for cyklom program prechádza jednotlivé prvky v poli a regexmi hľadá zhodu. Väčšina regexov v doimplementácii je napísaná mnou tiež, avšak niektoré som vedel využiť z knižnice.

Funkcia `make_log(data)` je zavolaná vo funkcii knižnice `processRequest(self)`. Tým pádom zaručím, že bude kontrolovaná hocijaká požiadavka na server. V jednotlivých požiadavkách hľadám požiadavky typu: „Ringing“, „ok“, „bye“, „decline“, „request terminated“ (všetky hľadám regexmi). Funkcia na jednotlivé požiadavky zareaguje nasledovne:

- Ak požiadavka obsahuje „Ringing“, vytvorí nový objekt triedy `Call`, nastaví v nej premennú `ringing_time` a `call_id` na požadovanú hodnotu, a vloží objekt `Call` do poľa hovorov
- Ak požiadavka obsahuje „ok“, spáruje ho s konkrétnou triedou `Call` v poli hovorov a následne nastaví v nej premennú `start_time` na požadovanú hodnotu.
- Ak požiadavka obsahuje „bye“, spáruje ho s konkrétnou triedou `Call` a poli hovorov a následne nastaví v nej premennú `end_time` na požadovanú hodnotu. Zapiše hovor do `log.txt` a pop-ne objekt `Call` z poľa hovorov.
- Ak požiadavka obsahuje „decline“ alebo „request terminated“, tak sa do `log.txt` súboru zapiše z hovoru iba parameter `ringing_time` a potom čas „decline“ (odmietnutia) alebo čas „request terminated“ (zrušenie požiadavky – zvonenia) alebo čas „Busy here“ (čas nedzvihnutia). Následne sa odstráni konkrétny objekt `Call` z poľa hovorov.

3.3.2 Úprava niekoľkých SIP stavových kódov

Úpravu stavových kódov realizuje funkcia `change_texts(data)`. Funkcia je volaná vo vnútri funkcii `processRequest(self)` po tom, ako sa vytvára logovanie hovorov (aby zmeny stavových kódov náhodou nepriamo neinterferovali s vytváraním logu). Funkcia vyzerá nasledovne:

```
def change_texts(data):
    if data[0] == b'SIP/2.0 100 Trying':
        data[0] = b'SIP/2.0 100 Skusam'

    elif data[0] == b'SIP/2.0 180 Ringing':
        data[0] = b'SIP/2.0 180 Zvonim'

    elif data[0] == b'SIP/2.0 486 Busy Here' or data[0] == b'SIP/2.0 486
Busy here':
        data[0] = b'SIP/2.0 486 Nechaj ma na pokoji'

    return data
```

Ako je možné vidieť, funkcia mení konkrétne stavové kódy „100 Trying“, „180 Ringing“ a „486 Busy here“. V knižnici som zmenil ešte stavový kód na jednom mieste a to je v metóde `processNonInvite(self)`. Tam ak server nevie spracovať požiadavku vráti „400 Zlá požiadavka“ miesto „400 Bad Request“.

4 Wireshark

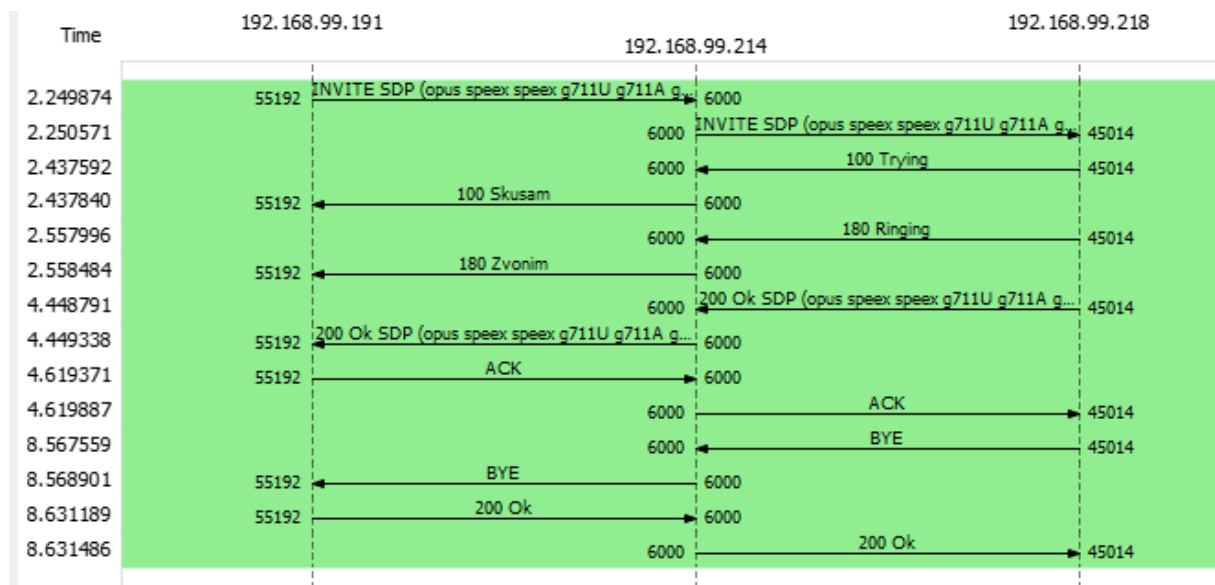
V testovaní wiresharku rozoberiem všetky základné a všetky potrebné funkcionality, ktoré sa týkajú SIP proxy. Z jednotlivých pcap trace-ov budem v dokumentácii ukazovať aj log.txt súbory. Testy budú realizované na 3 zariadeniach s IP adresami 192.168.99.214 (SIP proxy – môj počítač, číslo 100), 192.168.99.218 (môj mobil, číslo 101) a 192.168.99.191 (druhý mobil, číslo 444)

4.1 Základné funkcionality a video hovor

Rozoberaný pcap súbor normal_call.pcap. Odfotené rámce z pcap súboru s filtrom „sip“:

| | | | | | | |
|------|----------|----------------|----------------|---------|------|---------------------------------------------------------|
| 156 | 2.249874 | 192.168.99.191 | 192.168.99.214 | SIP/SDP | 1236 | Request: INVITE sip:101@192.168.99.214:6000 |
| 157 | 2.250571 | 192.168.99.214 | 192.168.99.218 | SIP/SDP | 1374 | Request: INVITE sip:101@192.168.99.214:6000 |
| 183 | 2.437592 | 192.168.99.218 | 192.168.99.214 | SIP | 340 | Status: 100 Trying |
| 184 | 2.437840 | 192.168.99.214 | 192.168.99.191 | SIP | 276 | Status: 100 Skusam |
| 205 | 2.557996 | 192.168.99.218 | 192.168.99.214 | SIP | 547 | Status: 180 Ringing |
| 206 | 2.558484 | 192.168.99.214 | 192.168.99.191 | SIP | 482 | Status: 180 Zvonim |
| 706 | 4.448791 | 192.168.99.218 | 192.168.99.214 | SIP/SDP | 1358 | Status: 200 Ok (INVITE) |
| 707 | 4.449338 | 192.168.99.214 | 192.168.99.191 | SIP/SDP | 1294 | Status: 200 Ok (INVITE) |
| 716 | 4.619371 | 192.168.99.191 | 192.168.99.214 | SIP | 443 | Request: ACK sip:101@192.168.99.218:45014;transport=udp |
| 718 | 4.619887 | 192.168.99.214 | 192.168.99.218 | SIP | 544 | Request: ACK sip:101@192.168.99.218:45014;transport=udp |
| 993 | 8.567559 | 192.168.99.218 | 192.168.99.214 | SIP | 460 | Request: BYE sip:444@192.168.99.191:55192;transport=udp |
| 994 | 8.568901 | 192.168.99.214 | 192.168.99.191 | SIP | 561 | Request: BYE sip:444@192.168.99.191:55192;transport=udp |
| 1004 | 8.631189 | 192.168.99.191 | 192.168.99.214 | SIP | 524 | Status: 200 Ok (BYE) |
| 1005 | 8.631486 | 192.168.99.214 | 192.168.99.218 | SIP | 460 | Status: 200 Ok (BYE) |

VoIP call flow:

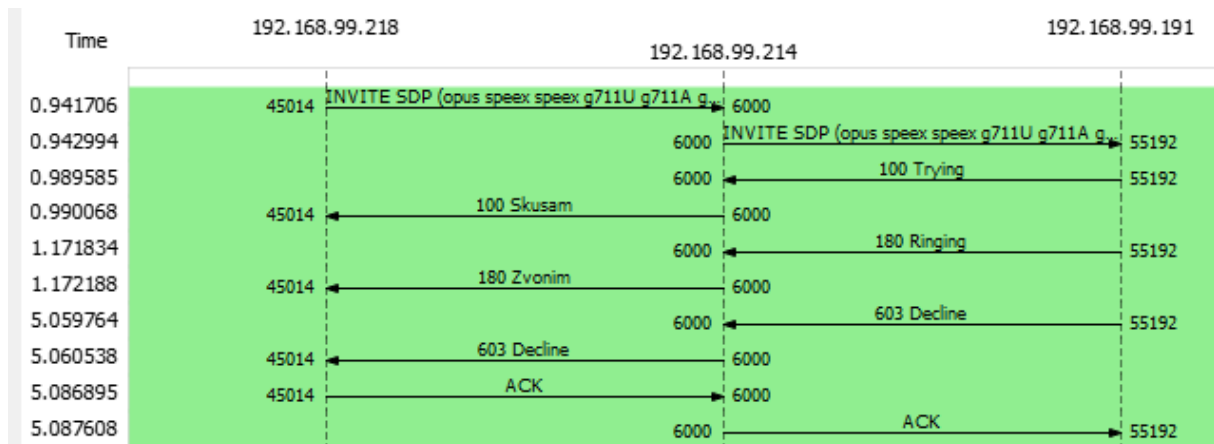


V rámcach vidíme zdvojenie z dôvodu, že každú požiadavku prijme server od jedného používateľa, následne ju spracuje a poskytne odozvu druhej strane (druhému používateľovi). V call flow je vidieť ako postupnosť INVITE -> Trying -> Ringing -> Ok -> ACK -> BYE -> OK je dodržaná, čo je vlastne ukážkové sa správanie jedného hovora. RTP (opus) cez ktorý je prenášaný samotný hovor nie je vidno z dôvodu, že to už prebieha medzi end-point-ami. Taktiež si môžeme povšimnúť ako PROXY server mení jednotlivé stavové správy.

Ďalší pcap súbor, ktorý si rozoberieme bude declined_call.

| | | | | | | |
|-----|----------|----------------|----------------|---------|------|---------------------------------------------|
| 74 | 0.941706 | 192.168.99.218 | 192.168.99.214 | SIP/SDP | 1253 | Request: INVITE sip:444@192.168.99.214:6000 |
| 75 | 0.942994 | 192.168.99.214 | 192.168.99.191 | SIP/SDP | 1391 | Request: INVITE sip:444@192.168.99.214:6000 |
| 82 | 0.989585 | 192.168.99.191 | 192.168.99.214 | SIP | 340 | Status: 100 Trying |
| 83 | 0.990068 | 192.168.99.214 | 192.168.99.218 | SIP | 276 | Status: 100 Skusam |
| 104 | 1.171834 | 192.168.99.191 | 192.168.99.214 | SIP | 531 | Status: 180 Ringing |
| 105 | 1.172188 | 192.168.99.214 | 192.168.99.218 | SIP | 466 | Status: 180 Zvonim |
| 624 | 5.059764 | 192.168.99.191 | 192.168.99.214 | SIP | 487 | Status: 603 Decline |
| 626 | 5.060538 | 192.168.99.214 | 192.168.99.218 | SIP | 423 | Status: 603 Decline |
| 628 | 5.086895 | 192.168.99.218 | 192.168.99.214 | SIP | 442 | Request: ACK sip:444@192.168.99.214:6000 |
| 629 | 5.087608 | 192.168.99.214 | 192.168.99.191 | SIP | 580 | Request: ACK sip:444@192.168.99.214:6000 |

VoIP call flow:

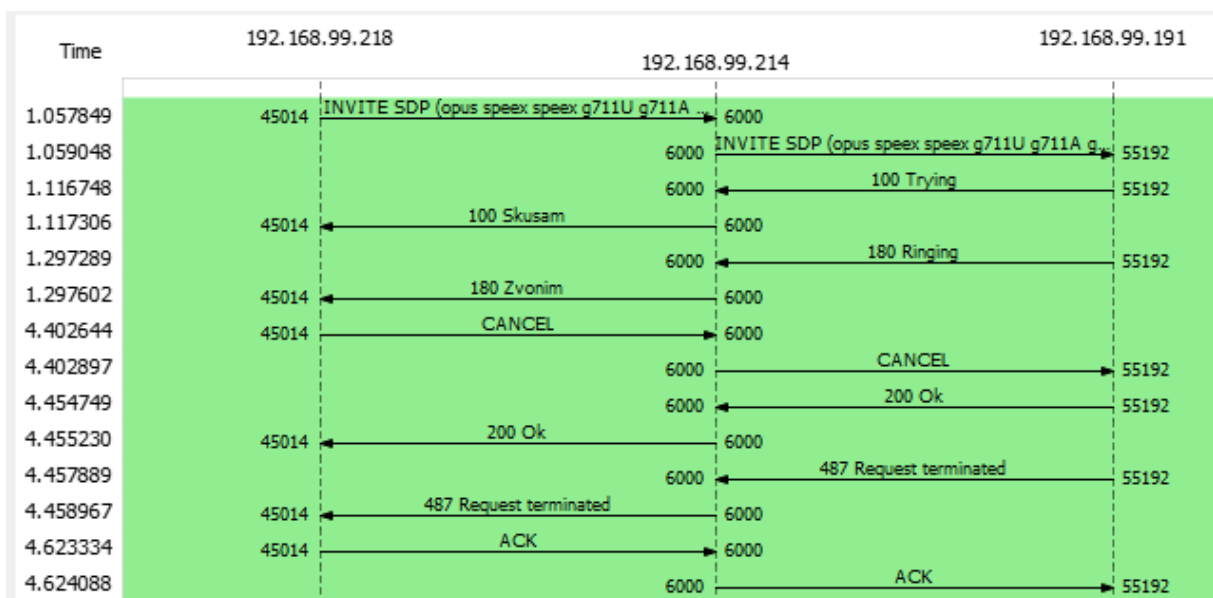


V tomto prípade iniciátor hovoru je 192.168.99.218. Vidíme, že INVITE, Trying a Ringing prebehol, tým pádom na druhej strane hovor už zvoní, ale druhá strana hovor odmietne a pošle Decline, na čo prvá strana odpovie ACK.

Nasledujúci pcap súbor, ktorý si rozoberieme bude request_terminated.

| | | | | | |
|-----|----------|----------------|----------------|---------|-------------------------------------------------------------------------|
| 115 | 1.057849 | 192.168.99.218 | 192.168.99.214 | SIP/SDP | 1253 Request: INVITE sip:444@192.168.99.214:6000 |
| 116 | 1.059048 | 192.168.99.214 | 192.168.99.191 | SIP/SDP | 1391 Request: INVITE sip:444@192.168.99.214:6000 |
| 123 | 1.116748 | 192.168.99.191 | 192.168.99.214 | SIP | 340 Status: 100 Trying |
| 124 | 1.117306 | 192.168.99.214 | 192.168.99.218 | SIP | 276 Status: 100 Skusam |
| 131 | 1.177958 | 192.168.99.218 | 192.168.99.214 | SIP | 641 Request: REGISTER sip:192.168.99.214:6000;transport=UDP (1 binding) |
| 132 | 1.179097 | 192.168.99.214 | 192.168.99.218 | SIP | 644 Status: 200 OK (REGISTER) (1 binding) |
| 147 | 1.297289 | 192.168.99.191 | 192.168.99.214 | SIP | 531 Status: 180 Ringing |
| 148 | 1.297602 | 192.168.99.214 | 192.168.99.218 | SIP | 466 Status: 180 Zvonim |
| 668 | 4.402644 | 192.168.99.218 | 192.168.99.214 | SIP | 399 Request: CANCEL sip:444@192.168.99.214:6000 |
| 669 | 4.402897 | 192.168.99.214 | 192.168.99.191 | SIP | 537 Request: CANCEL sip:444@192.168.99.214:6000 |
| 675 | 4.454749 | 192.168.99.191 | 192.168.99.214 | SIP | 482 Status: 200 Ok (CANCEL) |
| 676 | 4.455230 | 192.168.99.214 | 192.168.99.218 | SIP | 418 Status: 200 Ok (CANCEL) |
| 677 | 4.457889 | 192.168.99.191 | 192.168.99.214 | SIP | 498 Status: 487 Request terminated |
| 678 | 4.458967 | 192.168.99.214 | 192.168.99.218 | SIP | 434 Status: 487 Request terminated |
| 701 | 4.623334 | 192.168.99.218 | 192.168.99.214 | SIP | 442 Request: ACK sip:444@192.168.99.214:6000 |
| 702 | 4.624088 | 192.168.99.214 | 192.168.99.191 | SIP | 580 Request: ACK sip:444@192.168.99.214:6000 |

VoIP call flow:

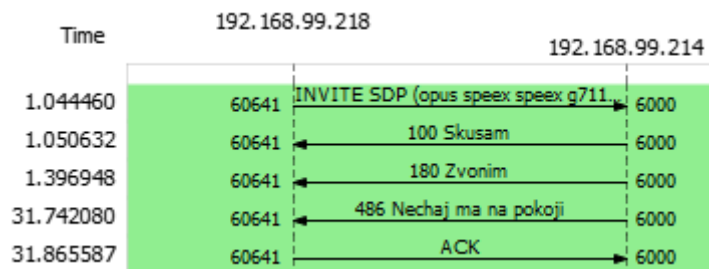


Teraz je iniciátor znova 192.168.99.218. INVITE, Trying a Ringing prebehol znova a následne iniciátor zruší hovor skôr ako bol zdvihnutý. Tým pádom sa vyššie CANCEL na čo druhá strana odpovie ACK a následne Request terminated. Prvá strana prijatie potvrdí pomocou ACK.

Nasledujúci prípad, na ktorý sa pozrieme je pcap súbor busy_call:

| | | | | | |
|------|-----------|----------------|----------------|---------|-------------------------------------------------|
| 122 | 1.044460 | 192.168.99.218 | 192.168.99.214 | SIP/SDP | 323 Request: INVITE sip:100@192.168.99.214:6000 |
| 125 | 1.050632 | 192.168.99.214 | 192.168.99.218 | SIP | 276 Status: 100 Skusam |
| 179 | 1.396948 | 192.168.99.214 | 192.168.99.218 | SIP | 464 Status: 180 Zvonim |
| 5211 | 31.742080 | 192.168.99.214 | 192.168.99.218 | SIP | 433 Status: 486 Nechaj ma na pokoji |
| 5228 | 31.865587 | 192.168.99.218 | 192.168.99.214 | SIP | 442 Request: ACK sip:100@192.168.99.214:6000 |

VoIP call flow:

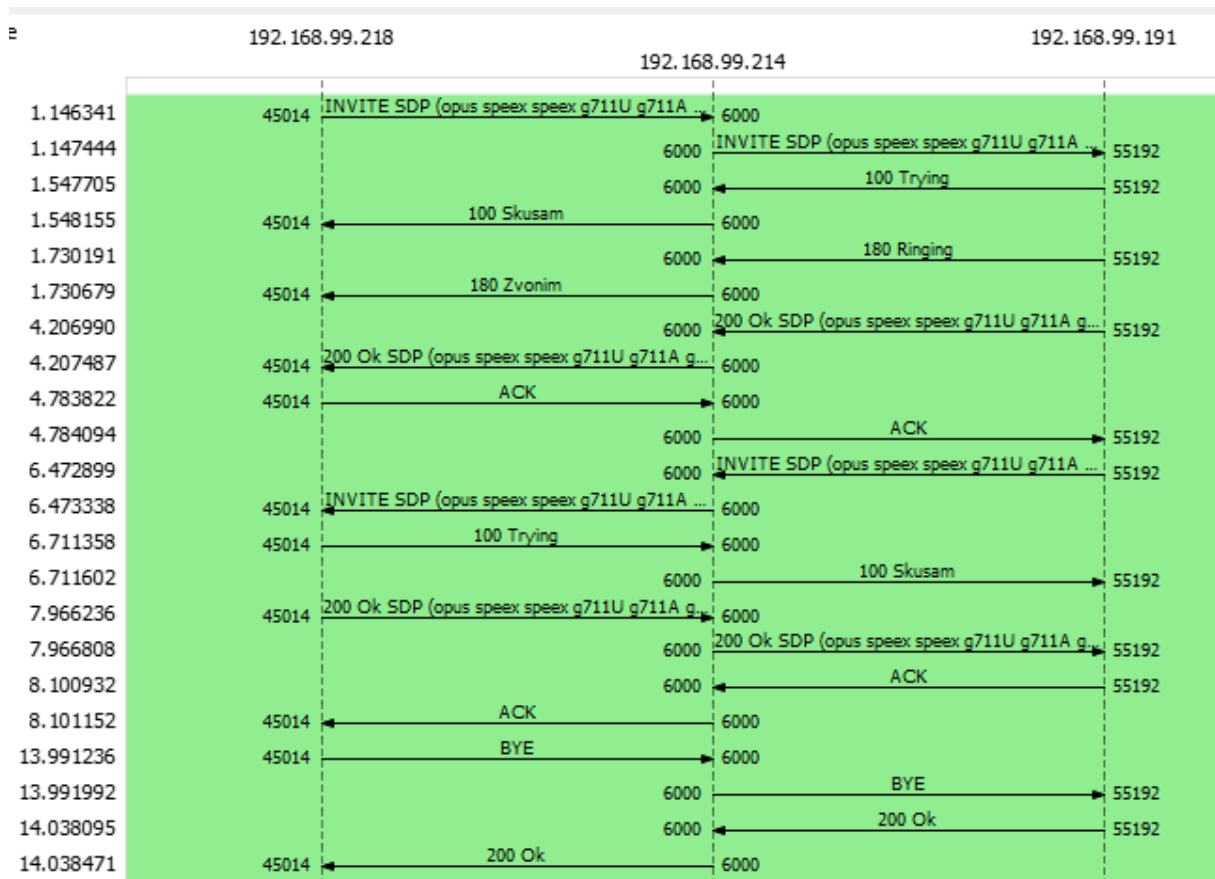


V tomto prípade sa hovor uskutočňoval s počítačom na ktorom je spustený aj samotný proxy server (preto vidíme len 2 uzly). Vidíme, že prebehol ukážkový INVITE, Trying a Ringing ale následne ostal hovor nezdvihnutý a bolo poslané Busy here. Druhá strana ešte odpovedala ACK na potvrdenie prijatia.

Posledný hovor rozobratý v tejto sekcii bude video hovor. Rozoberieme si pcap súbor video_call.

| | | | | | | |
|-----|-----------|----------------|----------------|---------|------|-----------------------------------------------------------------------|
| 32 | 1.146341 | 192.168.99.218 | 192.168.99.214 | SIP/SDP | 1253 | Request: INVITE sip:444@192.168.99.214:6000 |
| 33 | 1.147444 | 192.168.99.214 | 192.168.99.191 | SIP/SDP | 1391 | Request: INVITE sip:444@192.168.99.214:6000 |
| 39 | 1.547705 | 192.168.99.191 | 192.168.99.214 | SIP | 340 | Status: 100 Trying |
| 40 | 1.548155 | 192.168.99.214 | 192.168.99.218 | SIP | 276 | Status: 100 Skusam |
| 43 | 1.730191 | 192.168.99.191 | 192.168.99.214 | SIP | 531 | Status: 180 Ringing |
| 44 | 1.730679 | 192.168.99.214 | 192.168.99.218 | SIP | 466 | Status: 180 Zvonim |
| 365 | 4.206990 | 192.168.99.191 | 192.168.99.214 | SIP/SDP | 1342 | Status: 200 Ok (INVITE) |
| 366 | 4.207487 | 192.168.99.214 | 192.168.99.218 | SIP/SDP | 1278 | Status: 200 Ok (INVITE) |
| 438 | 4.783822 | 192.168.99.218 | 192.168.99.214 | SIP | 459 | Request: ACK sip:444@192.168.99.191:55192;transport=udp |
| 439 | 4.784094 | 192.168.99.214 | 192.168.99.191 | SIP | 560 | Request: ACK sip:444@192.168.99.191:55192;transport=udp |
| 591 | 6.472899 | 192.168.99.191 | 192.168.99.214 | SIP/SDP | 213 | Request: INVITE sip:101@192.168.99.218:45014;transport=udp, in-dialog |
| 593 | 6.473338 | 192.168.99.214 | 192.168.99.218 | SIP/SDP | 314 | Request: INVITE sip:101@192.168.99.218:45014;transport=udp, in-dialog |
| 597 | 6.711358 | 192.168.99.218 | 192.168.99.214 | SIP | 355 | Status: 100 Trying |
| 598 | 6.711602 | 192.168.99.214 | 192.168.99.191 | SIP | 291 | Status: 100 Skusam |
| 624 | 7.966236 | 192.168.99.218 | 192.168.99.214 | SIP/SDP | 247 | Status: 200 Ok (INVITE) |
| 628 | 7.966808 | 192.168.99.214 | 192.168.99.191 | SIP/SDP | 183 | Status: 200 Ok (INVITE) |
| 648 | 8.100932 | 192.168.99.191 | 192.168.99.214 | SIP | 444 | Request: ACK sip:101@192.168.99.218:45014;transport=udp |
| 649 | 8.101152 | 192.168.99.214 | 192.168.99.218 | SIP | 545 | Request: ACK sip:101@192.168.99.218:45014;transport=udp |
| 428 | 13.991236 | 192.168.99.218 | 192.168.99.214 | SIP | 459 | Request: BYE sip:444@192.168.99.191:55192;transport=udp |
| 429 | 13.991992 | 192.168.99.214 | 192.168.99.191 | SIP | 560 | Request: BYE sip:444@192.168.99.191:55192;transport=udp |
| 436 | 14.038095 | 192.168.99.191 | 192.168.99.214 | SIP | 523 | Status: 200 Ok (BYE) |
| 437 | 14.038471 | 192.168.99.214 | 192.168.99.218 | SIP | 459 | Status: 200 Ok (BYE) |

VoIP call flow:



Vidíme, že spočiatku hovor prebieha ako normálny hovor (INVITE, Trying, Ringing,...), ale približne 2s po zdvihnutí môžeme vidieť ďalší INVITE. Tento invite obsahuje v SDP, nielen audio Media Description ale aj video Media Description (napríklad rámec 591). Pošle sa aj Trying, keďže druhá strana musí potvrdiť, že chce zazdieľať kameru. Po vyše sekunde to druhá strana potvrdí a vyše sa OK, na čo opačná strana odpovie ACK. Ukončenie hovoru je štandardné

Logovanie týchto hovorov v súbore log_normal_types.txt vyzerá nasledovne (pripomínam, že poradie bolo 1. normálny hovor, 2. odmietnutý, 3. zrušený, 4. videohovor). Jedine logovanie prípadu, keď strana ostala nezdvihnutá (busy) je v log súbore log_busy.txt.

log_normal_types.txt

|-----

```
from: sip:444@192.168.99.214
to: sip:101@192.168.99.214
ringing: 16:07:54
start: 16:07:56
end: 16:08:00
```

```
from: sip:101@192.168.99.214
to: sip:444@192.168.99.214
ringing: 16:08:33
declined: 16:08:37
```

```
from: sip:101@192.168.99.214
to: sip:444@192.168.99.214
ringing: 16:08:58
request terminated: 16:09:02
```

```
from: sip:101@192.168.99.214
to: sip:444@192.168.99.214
ringing: 16:10:03
start: 16:10:06
end: 16:10:16
```

log_busy.txt:

|-----

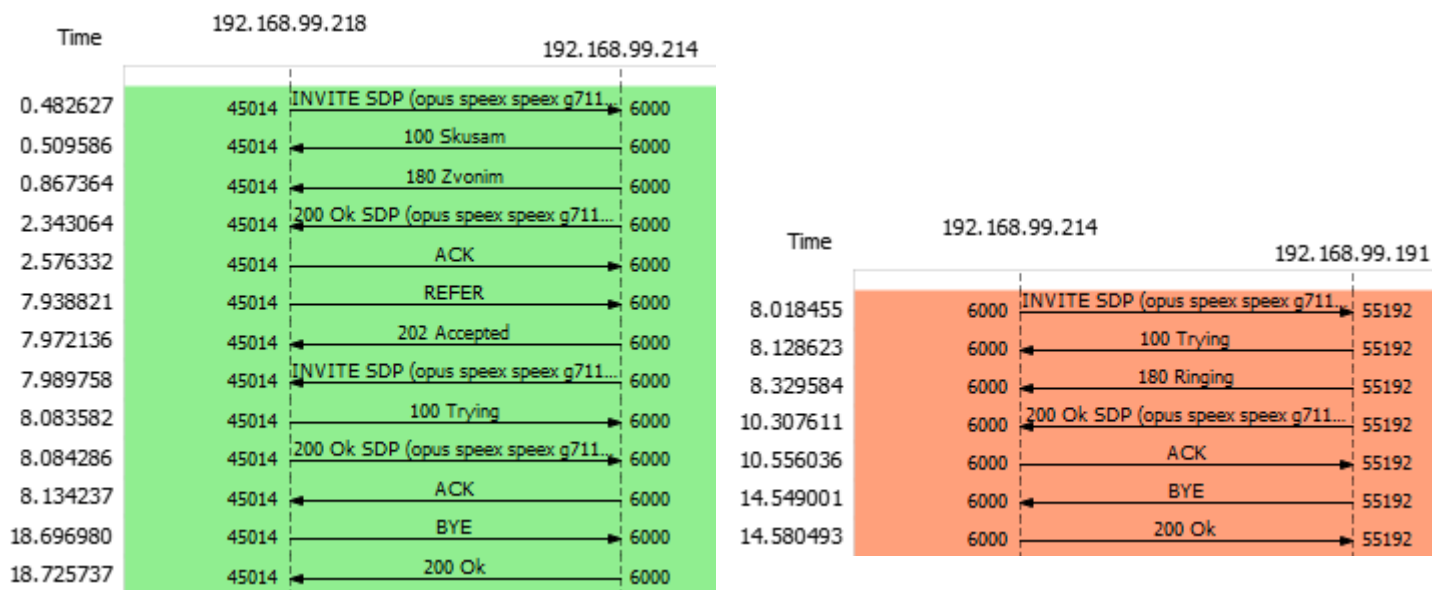
```
from: sip:101@192.168.99.214
to: sip:100@192.168.99.214
ringing: 13:35:47
busy -> ended: 13:36:18
```

4.2 Presmerovaný hovor

Ďalší si rozoberieme pcap súbor s názvom `referred_call`:

| | | | | | |
|-------|-----------|----------------|----------------|---------|----------------------------------------------------------------------------|
| 146 | 0.560000 | 192.168.99.218 | 192.168.99.214 | SIP | 642 Request: REGISTER sip:192.168.99.214:6000;transport=UDP (1 binding) |
| 147 | 0.561455 | 192.168.99.214 | 192.168.99.218 | SIP | 645 Status: 200 OK (REGISTER) (1 binding) |
| 219 | 0.891321 | 192.168.99.218 | 192.168.99.214 | SIP/SDP | 1253 Request: INVITE sip:100@192.168.99.214:6000 |
| 224 | 0.919903 | 192.168.99.214 | 192.168.99.218 | SIP | 276 Status: 100 Skusam |
| 340 | 1.301937 | 192.168.99.214 | 192.168.99.218 | SIP | 464 Status: 180 Zvonim |
| 830 | 2.743594 | 192.168.99.214 | 192.168.99.218 | SIP/SDP | 1291 Status: 200 Ok (INVITE) |
| 883 | 2.941685 | 192.168.99.218 | 192.168.99.214 | SIP | 453 Request: ACK sip:100@192.168.99.214;transport=udp |
| 2999 | 7.847661 | 192.168.99.218 | 192.168.99.214 | SIP | 695 Request: REFER sip:100@192.168.99.214;transport=udp, in-dialog |
| 3008 | 7.866188 | 192.168.99.214 | 192.168.99.218 | SIP | 465 Status: 202 Accepted |
| 3016 | 7.890581 | 192.168.99.214 | 192.168.99.218 | SIP/SDP | 1413 Request: INVITE sip:101@192.168.99.218:45014;transport=udp, in-dialog |
| 3041 | 7.959656 | 192.168.99.218 | 192.168.99.214 | SIP | 353 Status: 100 Trying |
| 3044 | 7.969245 | 192.168.99.218 | 192.168.99.214 | SIP/SDP | 1345 Status: 200 Ok (INVITE) |
| 3075 | 8.048942 | 192.168.99.214 | 192.168.99.191 | SIP/SDP | 1440 Request: INVITE sip:444@192.168.99.214:6000 |
| 3086 | 8.101321 | 192.168.99.191 | 192.168.99.214 | SIP | 338 Status: 100 Trying |
| 3100 | 8.130923 | 192.168.99.214 | 192.168.99.218 | SIP | 541 Request: ACK sip:101@192.168.99.218:45014;transport=udp |
| 3140 | 8.277544 | 192.168.99.191 | 192.168.99.214 | SIP | 529 Status: 180 Ringing |
| 4108 | 11.475924 | 192.168.99.191 | 192.168.99.214 | SIP/SDP | 1315 Status: 200 Ok (INVITE) |
| 4175 | 11.818042 | 192.168.99.214 | 192.168.99.191 | SIP | 540 Request: ACK sip:444@192.168.99.191:55192;transport=udp |
| 6251 | 15.741828 | 192.168.99.191 | 192.168.99.214 | SIP | 438 Request: BYE sip:100@192.168.99.214;transport=udp |
| 6264 | 15.762492 | 192.168.99.214 | 192.168.99.191 | SIP | 458 Status: 200 Ok (BYE) |
| 8232 | 20.902252 | 192.168.99.214 | 192.168.99.218 | SIP/SDP | 1402 Request: INVITE sip:101@192.168.99.218:45014;transport=udp, in-dialog |
| 8312 | 21.099778 | 192.168.99.218 | 192.168.99.214 | SIP | 353 Status: 100 Trying |
| 8317 | 21.113982 | 192.168.99.218 | 192.168.99.214 | SIP/SDP | 1333 Status: 200 Ok (INVITE) |
| 8373 | 21.277397 | 192.168.99.214 | 192.168.99.218 | SIP | 541 Request: ACK sip:101@192.168.99.218:45014;transport=udp |
| 10011 | 25.243669 | 192.168.99.214 | 192.168.99.218 | SIP | 541 Request: BYE sip:101@192.168.99.218:45014;transport=udp |
| 10049 | 25.353495 | 192.168.99.218 | 192.168.99.214 | SIP | 538 Status: 200 Ok (BYE) |

VoIP call flow:



V tomto prípade môžeme vidieť ako najprv 192.168.99.218 zavolať užívateľovi 192.168.99.214, na ktorom je as samotný proxy server. Uskutočnenie hovoru prebehne normálne a potom nastane REFER, ktorý presmeruje používateľa na iného. REFER sa prijme (Accepted) a prebehne nový invite medzi užívateľmi (keďže hovor bude podržaný). Užívateľ 192.168.99.214 pošle INVITE 192.168.99.191 keďže na neho bol presmerovaný. Tam vyzerá hovor ako klasický hovor. V zelenom call flowe vidíme, že kým bol užívateľ podržaný, tak položil hovor. Ak by hovor nepoložil, ale znovu spustil po nejakom čase, prebehol by nový invite a pokračovalo by sa v hovore normálne.

V logování hovorů (soubor log_refered.txt) vyzerajú hovory nasledovne:

|-----

from: sip:100@192.168.99.214
to: sip:444@192.168.99.214
ringing: 16:21:17
start: 16:21:19
end: 16:21:23

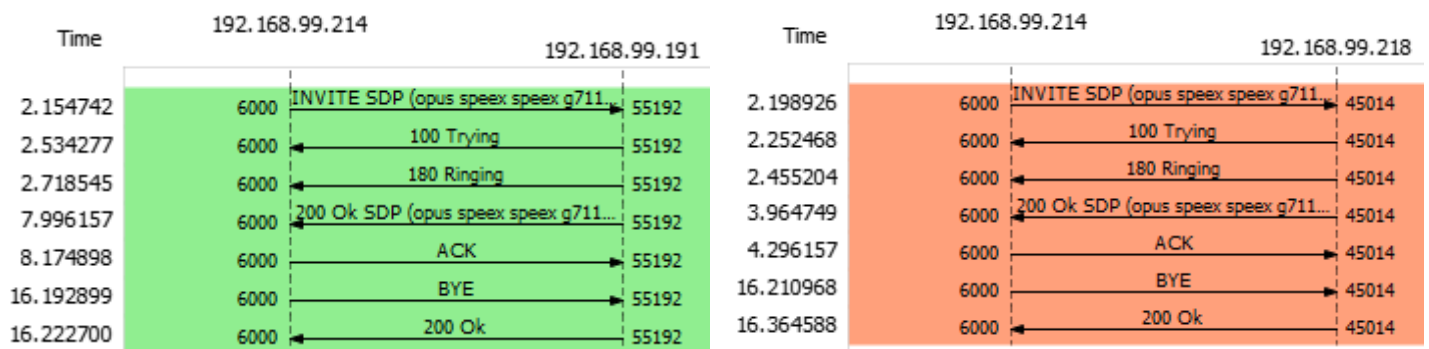
from: sip:101@192.168.99.214
to: sip:100@192.168.99.214
ringing: 16:21:09
start: 16:21:11
end: 16:21:27

4.3 Konferenčný hovor

Naposledy si rozoberieme prípad, kedy užívatelia chcú sprostredkovať konferenčný hovor (pcap súbor group_call):

| | | | | | |
|------|-----------|----------------|----------------|---------|------------------------------------------------------------------------------|
| 33 | 2.154742 | 192.168.99.214 | 192.168.99.191 | SIP/SDP | 1460 Request: INVITE sip:444@192.168.99.214:6000 |
| 34 | 2.198926 | 192.168.99.214 | 192.168.99.218 | SIP/SDP | 1460 Request: INVITE sip:101@192.168.99.214:6000 |
| 36 | 2.252468 | 192.168.99.218 | 192.168.99.214 | SIP | 338 Status: 100 Trying |
| 40 | 2.455204 | 192.168.99.218 | 192.168.99.214 | SIP | 545 Status: 180 Ringing |
| 42 | 2.534277 | 192.168.99.191 | 192.168.99.214 | SIP | 338 Status: 100 Trying |
| 43 | 2.718545 | 192.168.99.191 | 192.168.99.214 | SIP | 529 Status: 180 Ringing |
| 79 | 3.964749 | 192.168.99.218 | 192.168.99.214 | SIP/SDP | 1350 Status: 200 Ok (INVITE) |
| 109 | 4.296157 | 192.168.99.214 | 192.168.99.218 | SIP | 540 Request: ACK sip:101@192.168.99.218:45014;transport=udp |
| 110 | 4.302258 | 192.168.99.218 | 192.168.99.214 | SIP | 684 Request: SUBSCRIBE sip:100@192.168.99.214;conf-id=Lel83;transport=udp |
| 111 | 4.302614 | 192.168.99.214 | 192.168.99.218 | SIP | 673 Status: 200 OK (SUBSCRIBE) |
| 860 | 7.996157 | 192.168.99.191 | 192.168.99.214 | SIP/SDP | 1333 Status: 200 Ok (INVITE) |
| 895 | 8.174898 | 192.168.99.214 | 192.168.99.191 | SIP | 540 Request: ACK sip:444@192.168.99.191:55192;transport=udp |
| 925 | 8.325172 | 192.168.99.191 | 192.168.99.214 | SIP | 668 Request: SUBSCRIBE sip:100@192.168.99.214;conf-id=Lel83;transport=udp |
| 926 | 8.325770 | 192.168.99.214 | 192.168.99.191 | SIP | 657 Status: 200 OK (SUBSCRIBE) |
| 1958 | 11.696404 | 192.168.99.218 | 192.168.99.214 | SIP | 642 Request: REGISTER sip:192.168.99.214:6000;transport=UDP (1 binding) |
| 1959 | 11.696985 | 192.168.99.214 | 192.168.99.218 | SIP | 645 Status: 200 OK (REGISTER) (1 binding) |
| 3646 | 16.192899 | 192.168.99.214 | 192.168.99.191 | SIP | 540 Request: BYE sip:444@192.168.99.191:55192;transport=udp |
| 3653 | 16.210968 | 192.168.99.214 | 192.168.99.218 | SIP | 540 Request: BYE sip:101@192.168.99.218:45014;transport=udp |
| 3654 | 16.222700 | 192.168.99.191 | 192.168.99.214 | SIP | 521 Status: 200 Ok (BYE) |
| 3672 | 16.364588 | 192.168.99.218 | 192.168.99.214 | SIP | 537 Status: 200 Ok (BYE) |
| 3676 | 16.493024 | 192.168.99.191 | 192.168.99.214 | SIP | 669 Request: SUBSCRIBE sip:444@192.168.99.191:55192;transport=udp, in-dialog |
| 3677 | 16.493663 | 192.168.99.214 | 192.168.99.191 | SIP | 655 Status: 200 OK (SUBSCRIBE) |
| 3687 | 16.622822 | 192.168.99.218 | 192.168.99.214 | SIP | 685 Request: SUBSCRIBE sip:101@192.168.99.218:45014;transport=udp, in-dialog |
| 3688 | 16.623784 | 192.168.99.214 | 192.168.99.218 | SIP | 671 Status: 200 OK (SUBSCRIBE) |

VoIP call flow:



Iniciátor konferenčného hovoru je v tomto prípade 192.168.99.214. Vidíme, že sa nasledujúci prípad správa ako 2 rozdielne hovory, ktorý je sprostredkovaný cez 192.168.99.214.

Logovanie hovor je v súbore log_groupcall.txt:

```
-----  
from: sip:100@192.168.99.214  
to: sip:444@192.168.99.214  
ringing: 16:29:33  
start: 16:29:38  
end: 16:29:46  
-----
```

```
-----  
from: sip:100@192.168.99.214  
to: sip:101@192.168.99.214  
ringing: 16:29:32  
start: 16:29:34  
end: 16:29:46  
-----
```

5 Záver

V tomto zadaní sme si mali sprostredkovať SIP PROXY server. Taktiež sme sa mali naučiť ako sa správa, funguje a pracuje so SIP protokolom. Dozvedeli sme sa niečo aj o SDP protokole, ktorý je neodmysliteľnou súčasťou SIP protokolu. Moja implementácia sprostredkováva všetky základné aj rozšírené funkcie v zadaní.