

Mobilné technológie a aplikácie

Zadanie 1

SIP Proxy

Meno: Michal Kuklovský

AIS id: 103013

Cvičiaci: Ing. Adam Ševčík

Cvičenie: Štvrtok 8:00

Repozitár: [link](#)

1 Zadanie

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 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 dvomi **fyzickými** zariadeniami v rámci LAN siete.

Rozsah povinných funkcionalít:

- Registrácia účastníka (bez nutnosti autentifikácie)
- Vytocenie 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 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"

Každá doplnková funkcionality predstavuje plus 1 bod.

Počas prezentácie zadania musíte byť schopní na zariadení, kde beží ústredňa urobiť SIP trace a otvoriť ho pomocou tcpdump alebo Wireshark, a v primeranom rozsahu vysvetliť cvičiacemu, ako daná signalizácia prebieha.

2 Použitá knižnica a úpravy v kóde

github repozitár použitej knižnice: <https://github.com/tirfil/PySipFullProxy>

autor knižnice: Philippe Thirion

použitý súbor: sipfullproxy.py

main.py

Pre účely zadania sme upravili pôvodnú main funkciu a pridali sme ju do samostatného súboru. Doplnili sme výpis aktuálnej ip adresy do terminálu pre zjednodušenie pripojenia z klienta. Poslednou úpravou bolo špecifikovanie namespace premenným a triede zo súboru sipfullproxy.py.

```
if __name__ == "__main__":  
    hostname = socket.gethostname()  
    ipaddress = socket.gethostbyname(hostname)  
    if ipaddress == "127.0.0.1":  
        ipaddress = sys.argv[1]  
  
    print("SIP Proxy - ip address:" + ipaddress)  
  
    logging.basicConfig(format='%(asctime)s: %(levelname)s: %(message)s', filename='calls.log', level=logging.INFO, datefmt='%H:%M:%S')  
    logging.info("=====  
logging.info("SIP Proxy started: " + time.strftime("%a, %d %b %Y %H:%M:%S ", time.localtime()))  
logging.info("SIP Proxy hostname: " + hostname)  
logging.info("SIP Proxy IP address: " + ipaddress + "\n")  
  
sipfullproxy.recordroute = "Record-Route: <sip:%s:%d;lr>" % (ipaddress, PORT)  
sipfullproxy.topvia = "Via: SIP/2.0/UDP %s:%d" % (ipaddress, PORT)  
server = socketserver.UDPServer((HOST, PORT), sipfullproxy.UDPHandler)  
server.serve_forever()
```

Obr.1: Funkcia main v súbore main.py

Upravené časti pôvodného programu a sprevádzkovanie

- zakomentovaný regex, ktorý zakazoval ip adresy podsiete 192.168.0.0/16
- zmenené použitie funkcie join (zmena zo staršej verzie pythonu)
- zmena vyhľadávania v slovníku (zmena zo staršej verzie pythonu)
- upravené kódovanie a dekódovanie prijímaných a odosielaných správ
- info logovanie prerobené na denník hovorov

Odstránené časti pôvodného programu

- pôvodná funkcia main
- pôvodné debug logovanie
- funkcie, ktoré slúžili na autentifikáciu

Pridané časti

- pridanie calls = { } - slovník, ktorý ukladá informácie o hovore za účelom logovania
- pridané regexy na hľadanie stavových kódov 200, 486 a 603

```
rx_200ok = re.compile("^SIP/2.0 200 Ok$")
rx_busy = re.compile("^SIP/2.0 486 Busy here$")
rx_decline = re.compile("^SIP/2.0 603 Decline$")
```

Obr.2: Pridané regulárne výrazy

- slovník podľa ktorého sa menia stavové kódy

```
# response codes
resp = {
    "100 Trying" : "100 Skusam",
    "180 Ringing" : "180 Volam",
    "200 OK" : "200 V pohode",
    "200 Ok" : "200 V pohode",
    "400 Bad Request" : "400 Zly dopyt",
    "406 Not Acceptable" : "406 Neakceptovane",
    "480 Temporarily Unavailable" : "480 Nedostupne",
    "486 Busy here" : "486 Zaneprazdnene",
    "488 Not Acceptable Here" : "488 Neakceptovane",
    "500 Internal Server Error" : "500 Chyba servera",
    "603 Decline" : "603 Odmietnute"
}
```

Obr.3: Upravené stavové kódy

- pridané vymieňanie pôvodných stavových kódov priamo do kódu a vo funkcii processCode() pomocou slovníka

```
for before, after in resp.items():
    data[LINE] = data[LINE].replace(before, after)
```

Obr.4: Výmena stavových kódov

- pridaná funkcia logInvite()
- pridaná funkcia logBye()
- pridaná funkcia logCode()

Logovanie – denník hovorov

Vytváranie denníka hovorov majú na starosti funkcie logInvite(), logBye() a logCode(). Funkcia logInvite() pridáva záznam do slovníka „calls“ a zapisuje do logu začiatok nového hovoru alebo začiatok nového video hovoru.

```

def logInvite(self, logdata):
    callID = self.getID()
    callFrom = self.getOrigin()
    callTo = self.getDestination()

    if callID not in calls:
        calls[callID] = {
            "participants": [],
            "called_at": time.strftime("%H:%M:%S", time.localtime()),
            "picked_up": None,
            "video": False
        }
        calls[callID]["participants"].append(callFrom)
        calls[callID]["participants"].append(callTo)
        logging.info("Call started\n          call ID: %s\n          from: %s\n          to: %s" % (callID, callFrom, callTo))
    else:
        if callFrom not in calls[callID]["participants"]:
            calls[callID]["participants"].append(callFrom)
            logging.info("Call had new participant\n          call ID: %s\n          new: %s" % (callID, callFrom))
        if callTo not in calls[callID]["participants"]:
            calls[callID]["participants"].append(callTo)
            logging.info("Call had new participant\n          call ID: %s\n          new: %s" % (callID, callTo))
        for line in logdata:
            if "video" in line:
                if "video 0" in line and calls[callID]["video"]:
                    calls[callID]["video_ended_at"] = time.strftime("%H:%M:%S", time.localtime())
                    duration = datetime.strptime(calls[callID]["video_ended_at"], '%H:%M:%S') - datetime.strptime(calls[callID]["video_started_at"], '%H:%M:%S')
                    logging.info("Video call ended\n          call ID: %s\n          video duration: %s" % (callID, str(duration)))
                    calls[callID]["video"] = False
                else:
                    calls[callID]["video_started_at"] = time.strftime("%H:%M:%S", time.localtime())
                    logging.info("Video call started\n          call ID: %s" % (callID))
                    calls[callID]["video"] = True
            break

```

Obr.5: Funkcia logInvite()

Funkcia logByte() zapisuje do logu začiatok koniec hovoru a vypíše dĺžku trvania.

```

def logByte(self, data):
    id = self.getID()

    if id in calls:
        if calls[id]["picked_up"] is True:
            calls[id]["ended_at"] = time.strftime("%H:%M:%S", time.localtime())
            duration = datetime.strptime(calls[id]["ended_at"], '%H:%M:%S') - datetime.strptime(calls[id]["picked_up_at"], '%H:%M:%S')
            logging.info("Call ended by %s\n          call ID: %s\n          call duration: %s\n" % (self.getOrigin(), id, str(duration)))

```

Obr.6: Funkcia logByte()

Funkcia logCode() zapisuje do logu kedy bol prijatý hovor a zaznamenáva aj odmietnuté a nezodvihnuté hovory. Ďalej ešte ukladá čas poslednej správy „200 Ok“.

```

def logCode(self, data):
    id = self.getID()
    if rx_200ok.search(data[LINE]):
        if id in calls:
            if calls[id]["picked_up"] is None:
                calls[id]["picked_up"] = True
                calls[id]["picked_up_at"] = time.strftime("%H:%M:%S", time.localtime())
                logging.info("Call picked up by %s\n          call ID: %s" % (self.getDestination(), id))
            if "ended_at" in calls[id]:
                if calls[id]["picked_up"] is True and calls[id]["ended_at"] is not None:
                    calls[id]["end_confirmed_at"] = time.strftime("%H:%M:%S", time.localtime())
            else:
                calls[id]["last_200ok"] = time.strftime("%H:%M:%S", time.localtime())

    if rx_busy.search(data[LINE]):
        if id in calls:
            if calls[id]["picked_up"] is None:
                calls[id]["picked_up"] = False
                logging.info("Call was not picked up\n          call ID: %s\n" % (id))

    if rx_decline.search(data[LINE]):
        if id in calls:
            if calls[id]["picked_up"] is None:
                calls[id]["picked_up"] = False
                logging.info("Call was declined by %s\n          call ID: %s\n" % (self.getDestination(), id))

```

Obr.6: Funkcia logCode()

```

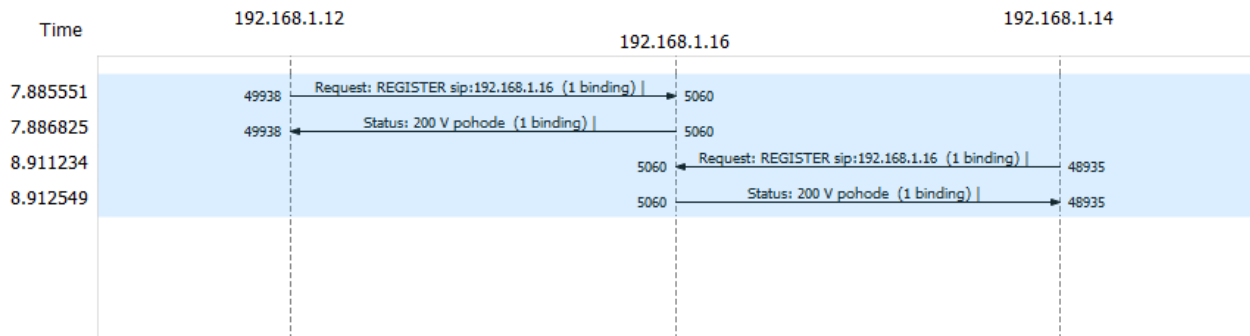
1  04:57:27:INFO:SIP Proxy started: Mon, 28 Feb 2022 04:57:27
2  04:57:27:INFO:SIP Proxy hostname: LAPTOP-OC829A45
3  04:57:27:INFO:SIP Proxy IP address: 192.168.1.20
4
5  04:57:37:INFO:Call started
6      |      |      |      |      |      |      |      |      |      |
7      |      |      |      |      |      |      |      |      |      |
8      |      |      |      |      |      |      |      |      |      |
9  04:58:23:INFO:Call was not picked up
10     |      |      |      |      |      |      |      |      |      |
11     |      |      |      |      |      |      |      |      |      |
12  04:58:27:INFO:Call started
13     |      |      |      |      |      |      |      |      |      |
14     |      |      |      |      |      |      |      |      |      |
15     |      |      |      |      |      |      |      |      |      |
16  04:58:30:INFO:Call was declined by redminote9@192.168.1.20
17     |      |      |      |      |      |      |      |      |      |
18     |      |      |      |      |      |      |      |      |      |
19  04:58:32:INFO:Call started
20     |      |      |      |      |      |      |      |      |      |
21     |      |      |      |      |      |      |      |      |      |
22     |      |      |      |      |      |      |      |      |      |
23  04:58:34:INFO:Call picked up by redminote9@192.168.1.20
24     |      |      |      |      |      |      |      |      |      |
25  04:58:42:INFO:Call ended by redminote5@192.168.1.20
26     |      |      |      |      |      |      |      |      |      |
27     |      |      |      |      |      |      |      |      |      |
28
29  04:58:44:INFO:Call started
30     |      |      |      |      |      |      |      |      |      |
31     |      |      |      |      |      |      |      |      |      |
32     |      |      |      |      |      |      |      |      |      |
33  04:58:47:INFO:Call picked up by redminote9@192.168.1.20
34     |      |      |      |      |      |      |      |      |      |
35  05:00:14:INFO:Call ended by redminote9@192.168.1.20
36     |      |      |      |      |      |      |      |      |      |
37     |      |      |      |      |      |      |      |      |      |
38

```

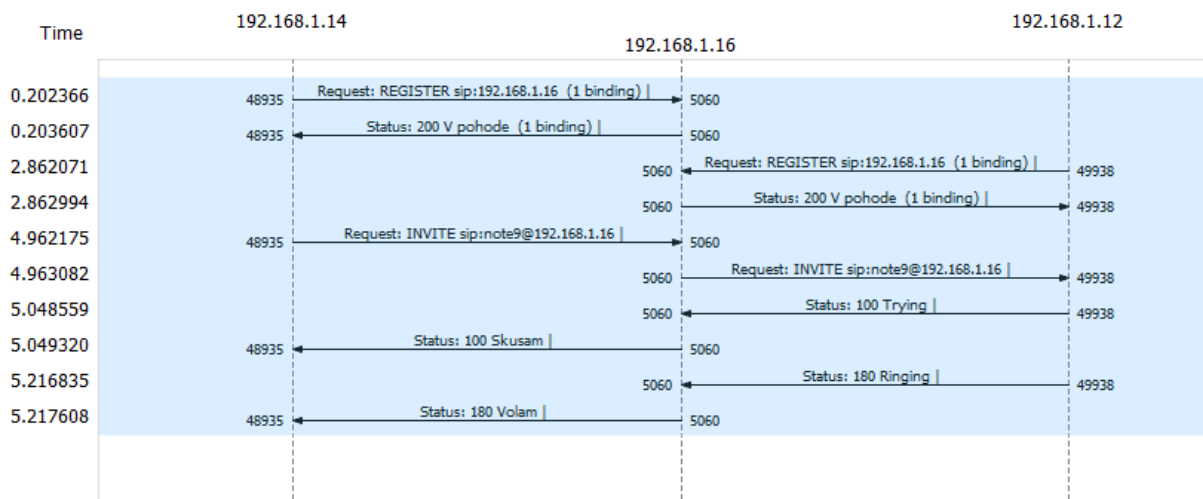
Obr.7: Denník hovorov

3 Grafy k scenárom zo zadania

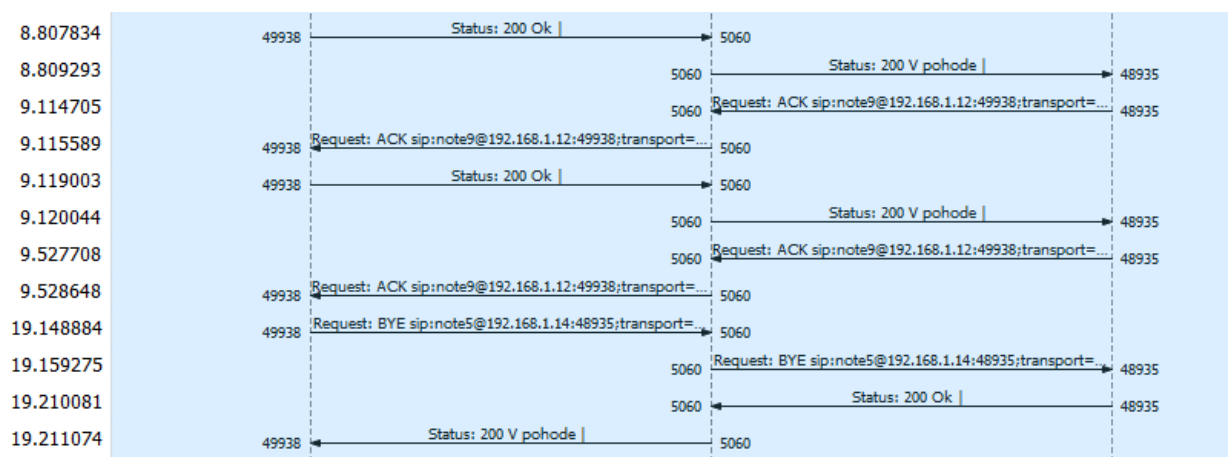
- Registrácia



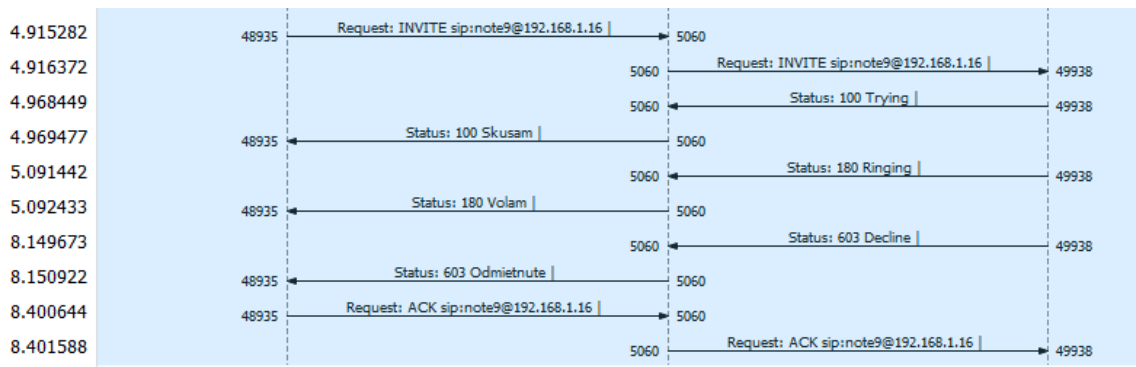
- Vytočenie hovoru a zvonenie na druhej strane



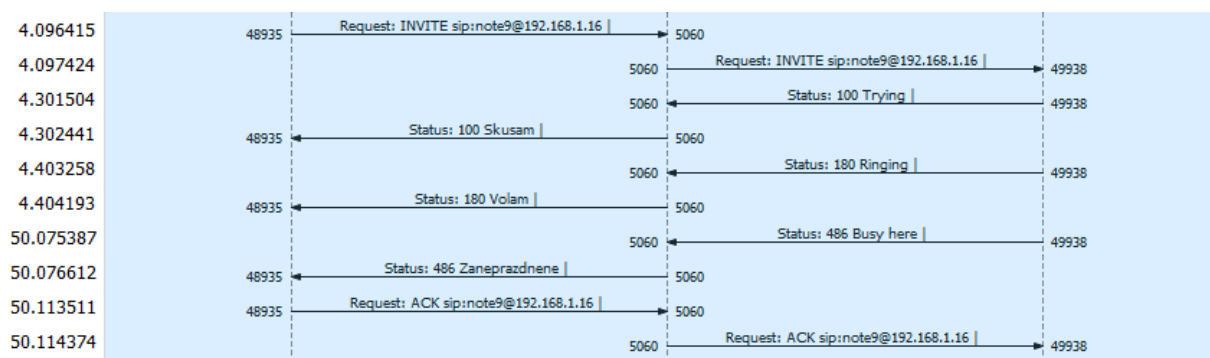
- Prijatie hovoru druhou stranou a ukončenie prijatého hovoru



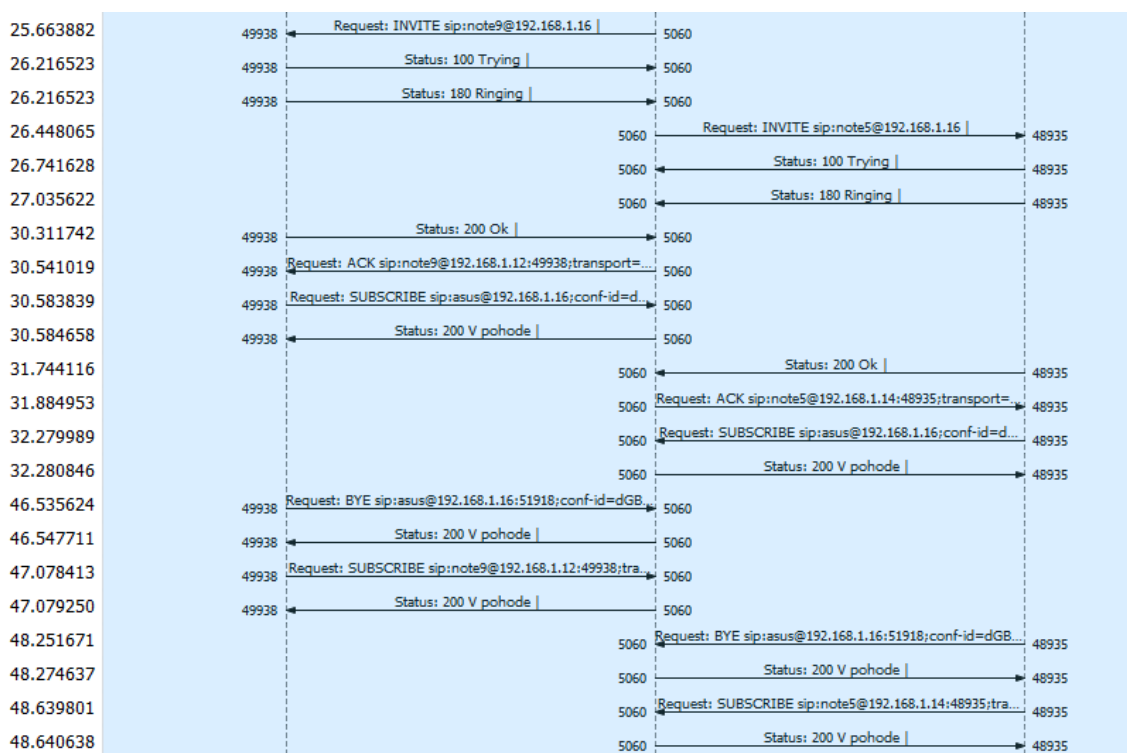
- Ukončenie neprijatého hlasového hovoru



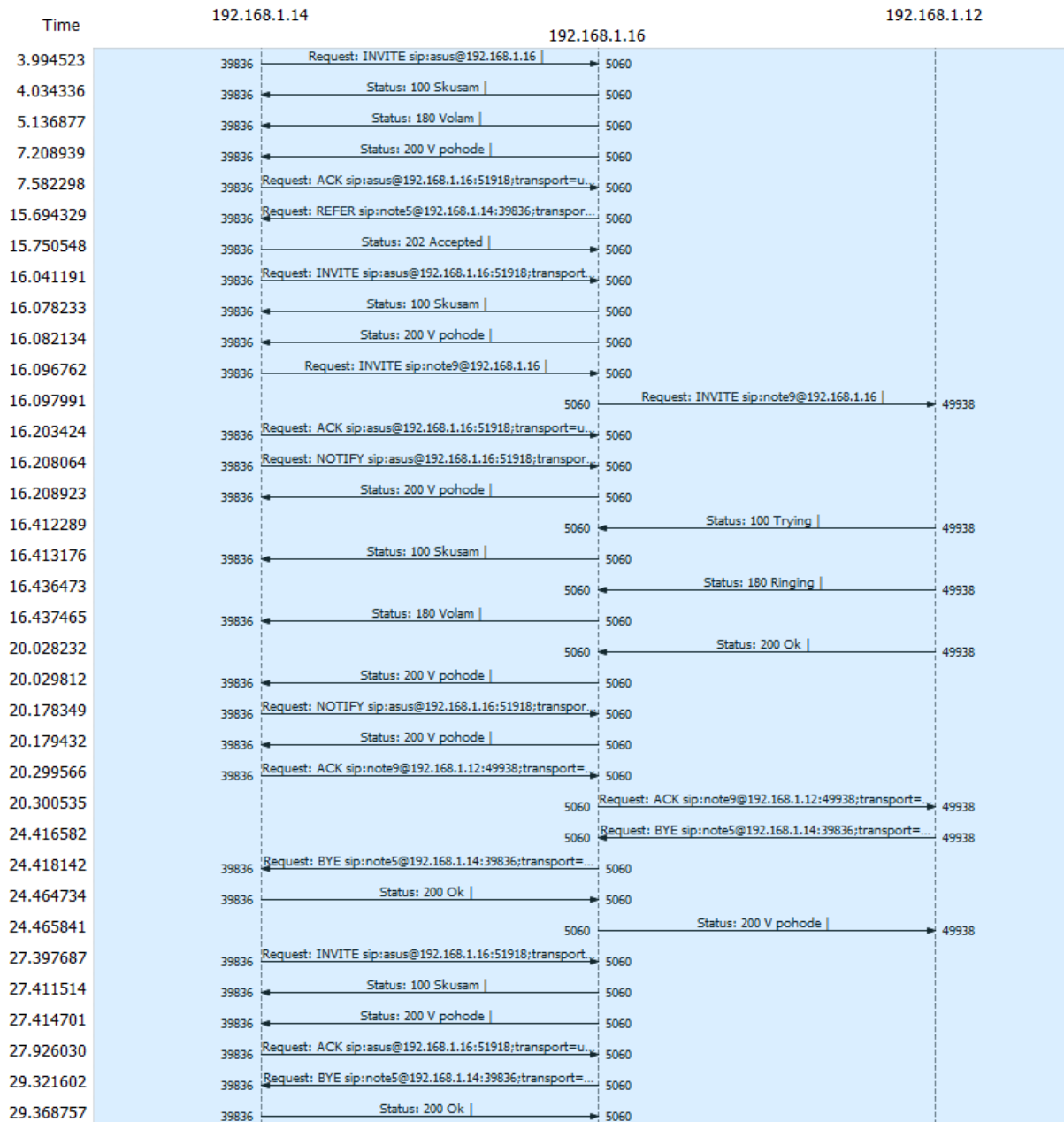
- Nezodvihnutý hovor



- Konferenčný hovor



- Presmerovanie hovoru



- Videohovor

