

Adrian Szacsko

*Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií
Ilkovičova 2, 842 16 Bratislava 4*

xszacsko@stuba.sk

25. január 2022

Zadanie č. 1

SIP Proxy (Telefónna ústredňa)

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

Cvičiaci: Ing. Marek Galinski, PhD.

Cvičenie: Streda 12:00

Obsah

Hlavná myšlienka zadania	3
Implementácia.....	3
Upravené chyby.....	3
Rozšírené funkcionality - Logovanie.....	3
Rozšírené funkcionality – SIP stavové kódy	4
Zakomponovanie prevzatého kódu do knižnice	5
Podporované funkcionality	5
Používateľské rozhranie	5
Použité knižnice a verzia Pythonu	6
Analýza Pcap súborov.....	6
Prijatý hovor	6
Zaneprázdnený hovor.....	7
Odmietnutý hovor	7
Skupinový hovor.....	8
Presmerovaný hovor	9
Videohovor	10

Hlavná myšlienka zadania

Cieľom zadania je zrealizovať VoIP hovor pomocou SIP Proxy serveru. Implementácia SIP Proxy serveru je realizovateľná prostredníctvom použitia kódu stiahnutého z webu, ktorý bude upravený a zakomponovaný do knižnice. Zakázané je však použiť existujúci SIP Proxy ako napr. Asterisk, kde len skompilovanie alebo priamo spustenie cudzej binárky je nutné. Hovor musí byť realizovaný medzi dvomi fyzickými zariadeniami v rámci LAN siete.

Implementácia

Implementácia telefónnej ústredne (SIP Proxy) bola realizovaná v jazyku Python pomocou kódu (sipfullproxy.py) zverejneného od užívateľa „tirfil“ v GitHub (<https://github.com/tirfil/PySipFullProxy/blob/master/sipfullproxy.py>). Daný kód bol zverejnený na platforme GitHub pred niekoľkými rokmi a z dôvodu priebežných zmien programovacieho jazyka Python pri novších verziách je kód nefunkčný. Po analýze daného kódu a následnej úprave chybných riadkov upravený kód bol plne funkčný a umožnil hostovanie telefónnej ústredne.

Upravené chyby

Jednotlivé úpravy kódu, ktoré boli realizované:

1. Importovanie knižnice SocketServer bolo zmenené na socketserver, nakoľko vo verzií Pythonu, ktorú som používal (Python 3.9.7) bol originálny názov knižnice neplatný.
2. Funkcia `join()` bola použitá neplatne a z tohto dôvodu boli riadky `string.join(data, "\r\n")` zmenené na `"\r\n".join(data)`.
3. Funkcia `dict.has_key(key)` je v novších verziách Pythonu používaná a podporovaná, ako `key in dict`.
4. Neplatne vyhodnotené IP adresy, ako napr. 192.168.x.x boli z dôvodu používania SIP Proxy v LAN sieti zmenené na platné.
5. Všetky SIP správy, ktoré náš Proxy server spracúva sú v upravenej verzii kódu prekonvertované na typ String a len pri preposielaní SIP správ sú konvertované späť na typ Bytes. Dôvodom je uľahčenie práce so SIP správami.

Rozšírené funkcionality - Logovanie

Prevzatý kód bol rozšírený s doplnkovými funkcionalitami, ako napr. Logovanie "denníka hovorov", kde je uvedené kto, kedy a komu volal, kedy bol hovor prijatý resp. ukončený. Tieto logy sú zapísané do logs.txt a počas behu programu aktualizované*. Táto funkcia je implementovaná pomocou vytvorených tried Call a Logs. V triede Call sú zaznamenané jednotlivé atribúty hovorov, ako napr. volajúca osoba, volané osoby, začiatok volania, koniec volania. Trieda Log pozbiera jednotlivé SIP správy a rozdelí ich do rôznych volaní. Jednotlivé volania sú uložené do predtým spomenutej triedy Call.

*Poznámka: Jednotlivé volania sú zapísané po ukončení daného hovoru.

```
----Call declined----  
From: sip:100@192.168.0.124  
To: ['sip:102@192.168.0.124']  
Start time: 16:57:30  
End time:16:58:00  
  
----Call----  
From: sip:100@192.168.0.124  
To: ['sip:102@192.168.0.124']  
Start time: 16:58:40  
End time:16:58:53
```

1. Obrázok - Logovanie

```
----Call----  
From: sip:102@192.168.0.124  
To: ['sip:104@192.168.0.124', 'sip:100@192.168.0.124']  
Start time: 17:16:26  
End time:17:17:15  
  
----Call----  
From: sip:102@192.168.0.124  
To: ['sip:104@192.168.0.124', 'sip:100@192.168.0.124']  
Start time: 17:18:58  
End time:17:19:27
```

2. Obrázok - Logovanie medzi viacerými účastníkmi hovoru

Rozšírené funkcionality – SIP stavové kódy

Niektoré SIP stavové kódy pri spracovaní SIP správ sú upravené z anglického jazyka na slovenský jazyk. Tieto stavové kódy sú uložené v priečinku programu do codes.json súboru. Program pri spracovaní správ porovnáva daný stavový kód SIP správy so stavovým kódom uloženým do codes.json súboru. Pri nájdení zhody je daný stavový kód vymenený a aktualizovaná verzia je preposlaná klientom.

```
{  
  "100 Trying": "100 Skusam",  
  "180 Ringing": "180 Zvonim",  
  "200 OK": "200 Vybavene",  
  "400 Bad Request": "400 Zla požiadavka",  
  "408 Request Timeout": "408 Casovy limit vyprsal",  
  "480 Temporarily Unavailable": "480 Docasne nedostupne",  
  "486 Busy Here": "486 Zaneprazdneny"  
}
```

3. Obrázok - Súbor codes.json

Zakomponovanie prevzatého kódu do knižnice

Prevzatý a upravený kód bol zakomponovaný do knižnice pomocou vytvorenej funkcie `start_server()`. Funkcia slúži na zistenie IP adresy, nastavenie portu, vytvorenie proxy servera a vytvorenie nového vlákna pomocou ktorého je možné cez terminál vypnúť server. Funkcia tiež zavolá metódu servera `serve_forever()`, ktorá prijíma a spracúva SIP správy. Port SIP Proxy servera je automaticky nastavená na 5070, ale pri manuálnom zadaní je možné zmeniť.

Podporované funkcionality

- 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)
- Možnosť zrealizovať konferenčný hovor
- Možnosť presmerovať hovor
- Možnosť realizovať videohovor
- Logovanie "denníka hovorov"
- Úprava SIP stavových kódov v zdrojovom kóde proxy

Používateľské rozhranie

Používateľské rozhranie je zrealizované pomocou terminálu, kde je možné nastaviť IP adresu SIP Proxy automaticky alebo manuálne. Pri výbere detekcie IP adresy server automaticky nájde a nastaví svoju IP adresu a port nastaví na 5070, kým pri manuálnom výbere server vyžiada od používateľa IP adresu a server skontroluje platnosť adresy.

Počas behu programu je možnosť SIP Proxy server vypnúť pomocou príkazu „exit“ do konzoly.

```
Server is starting...
Do you want to set up IP address automatically? [y/n] y
IP address: 192.168.0.124
Port: 5070
INFO: Type exit to close and kill the program
|
```

4. Obrázok - Automatický výber IP adresy a portu

```
Server is starting...
Do you want to set up IP address and PORT automatically? [y/n] n
Add IP address: 192.168.81.91
Add port number: sdasd
Warning: Illegal port number!
Add port number: dfsdf
Warning: Illegal port number!
Add port number: 5080
IP address: 192.168.81.91
Port: 5080
INFO: Type exit to close and kill the program
```

5. Obrázok - Manuálne zadanie IP adresy a portu

Použité knižnice a verzia Pythonu

Pri implementácii telefónnej ústredne boli použité nasledovné knižnice:

- copy
- socketserver
- re
- socket
- threading
- time
- logging
- json
- zakomponovaný prevzatý kód do knižnice: server_lib

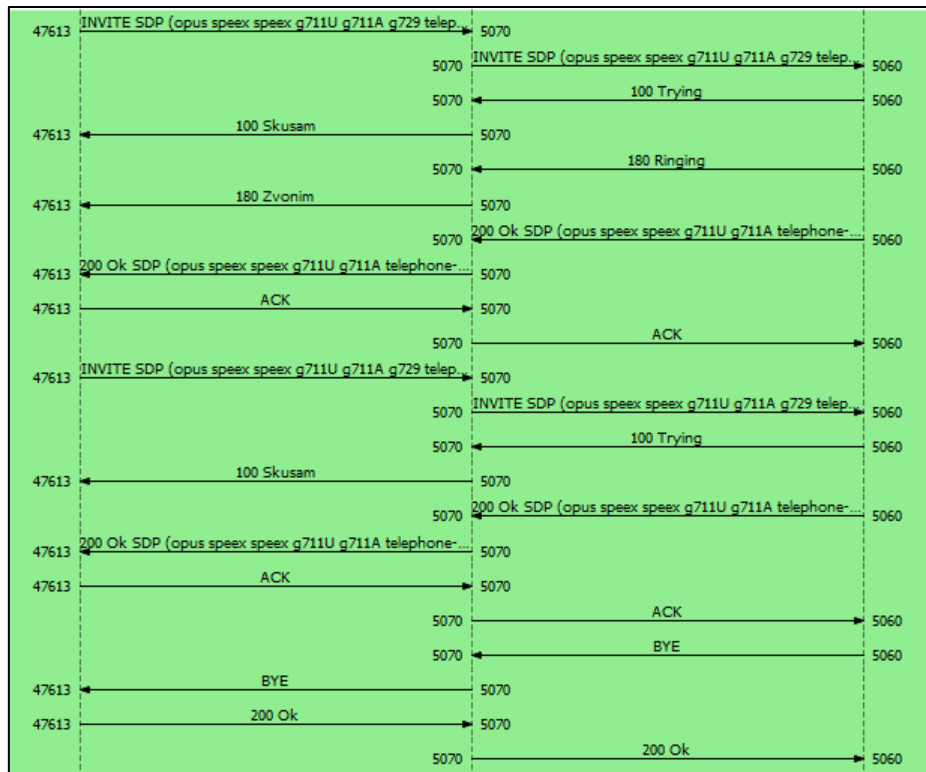
Zadanie bolo zrealizované v jazyku Python 3.9.7

Analýza Pcap súborov

Prijatý hovor

Pri volaní volajúca osoba pošle SIP správu INVITE, v ktorom sa nachádza adresa osoby, ktorému sa volá. Správa INVITE obsahuje SDP protokol, cez ktoré sú poslané podporované kodeky. Volaná osoba odpovie so SIP správami TRYING a RINGING aby notifikoval volajúcu osobu o zvonení zariadenia. Po vyzdvihnutí volajúca osoba pošle OK spolu s SDP protokolom, ktorá obsahuje podporujúce kodeky volaného užívateľa. Po obdržaní správy ACK od volajúcej osoby je hovor začatý.

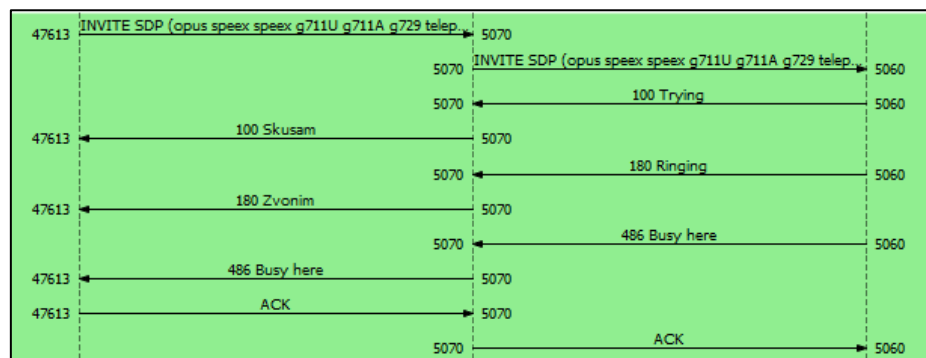
Pri zložení hovoru je poslaná správa BYE, ktorá oznámi druhú stranu o zložení hovoru. Druhá strana po obdržaní BYE správy pošle OK, s ktorým oznámi ukončenie hovoru.



Obrázok 6 - Prijatý hovor

Zaneprázdnený hovor

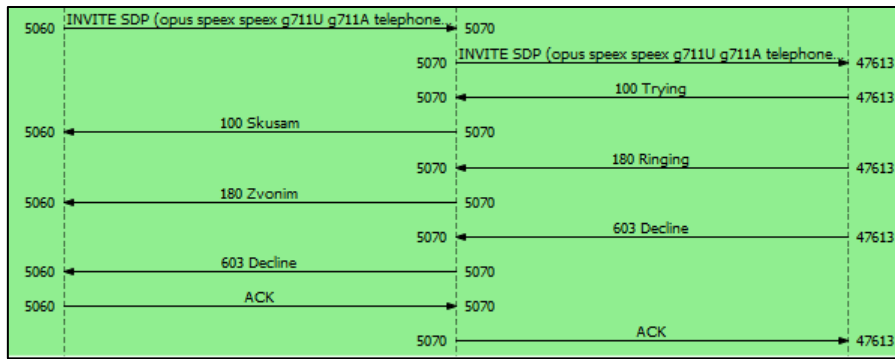
Pri prebiehajúcom volaní po určitej dobe nečinnosti volaná strana pošle správu BUSY HERE, ktorá znamená neúspešné volanie z dôvodu nereagovania na zvonenie. Volajúca strana po obdržaní správy BUSY HERE ukončí volanie a pošle ACK na potvrdenie tejto udalosti.



Obrázok 7 - Zaneprázdnený hovor

Odmietnutý hovor

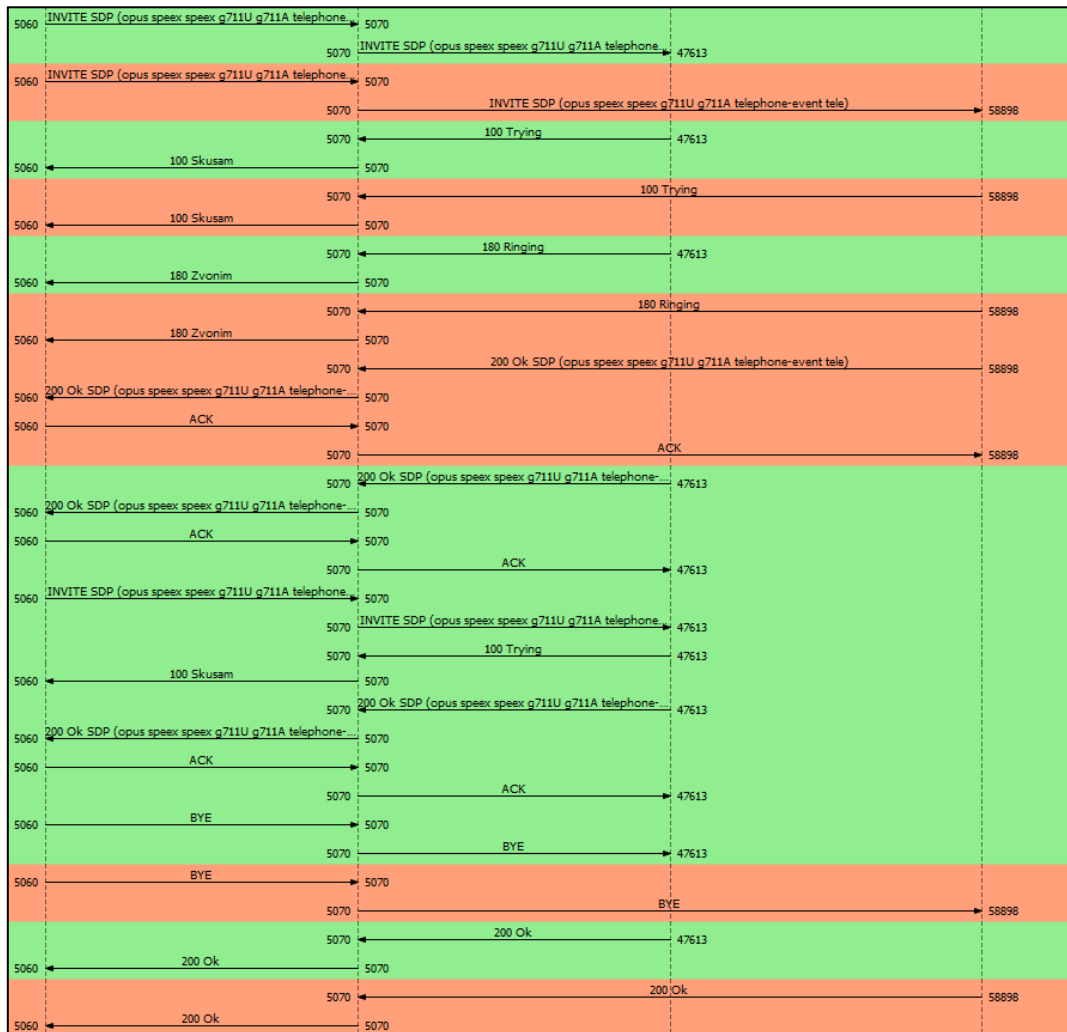
Podobne, ako pri zaneprázdnenej hovore, volaná osoba pošle DECLINE správu, ak pri zvonení volaná osoba odmietne prijatie hovoru. Odpoveďou volajúcej strany na obdržanie DECLINE správy je správa ACK, po ktorej je volanie ukončené.



Obrázok 8 - Odmietnutý hovor

Skupinový hovor

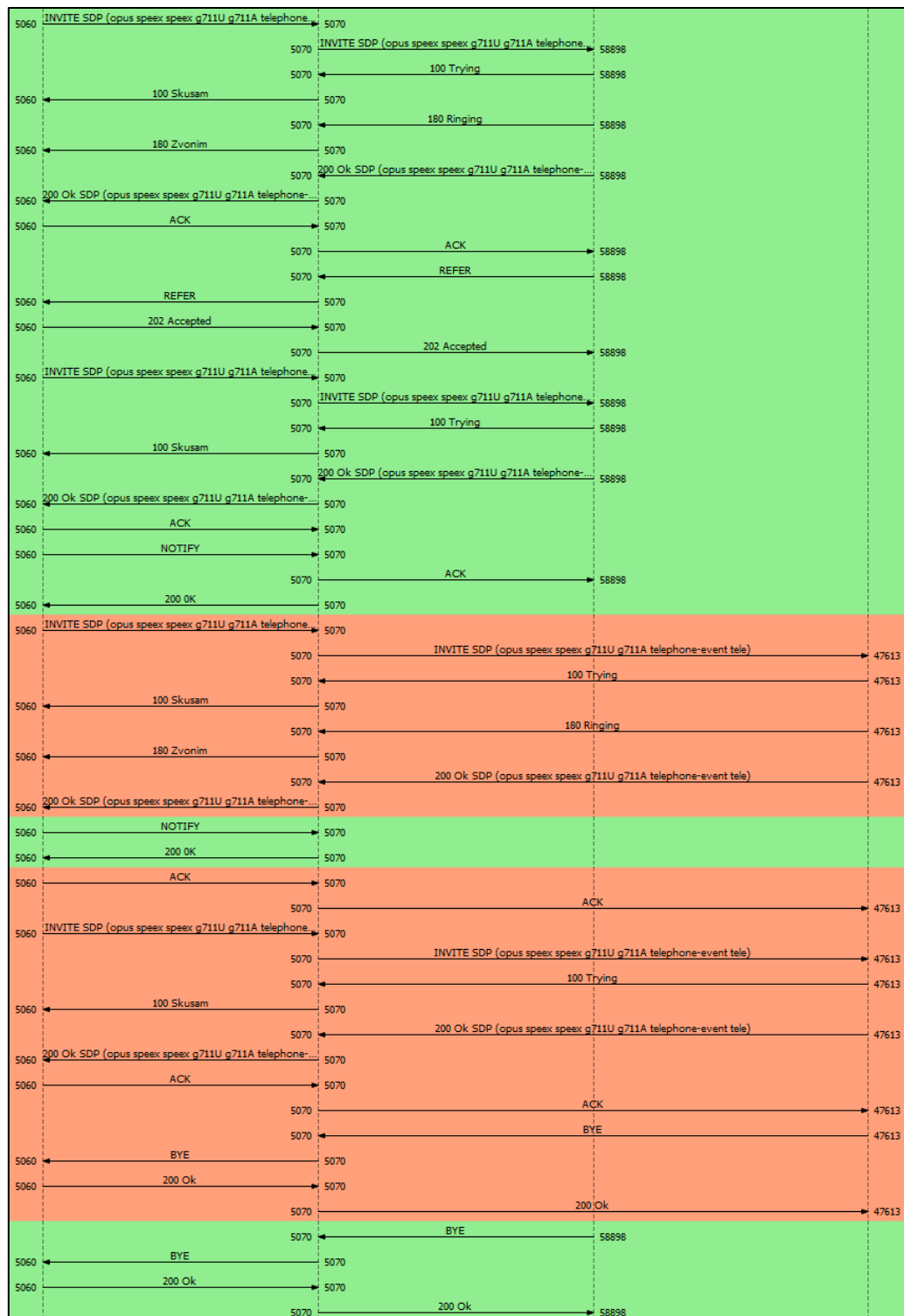
Skupinový hovor je podobný hovor, ako hovor medzi dvomi účastníkmi. Líši sa v tom, že pri začatí hovoru sú správy ako napr. INVITE, TRYING atď. poslané medzi viacerými účastníkmi. Skupinový hovor je možné sledovať ako viaceré individuálne hovory medzi účastníkmi.



Obrázok 9 - SIP správy medzi zariadení v skupinovom hovore

Presmerovaný hovor

Pri presmerovaní hovoru účastník, ktorý chce hovor presmerovať pošle správu REFER (táto správa obsahuje informácie o klientovi, na ktorý má hovor presmerovať) druhým účastníkom, ktorá na to odpovie so správou ACCEPTED a pošle INVITE správu, ktorá slúži na notifikovania druhého užívateľa o zmene typu hovoru (v tomto príklade bude poslaná INVITE správa s predmetom: Call on hold). Po obdržaní INVITE klient pošle TRYING a OK, ak hovor bol úspešne podržaný a druhá strana pošle ACK správu na potvrdenie tejto udalosti. Po poslaní ACK správy sa inicializuje hovor s novým klientom na spôsob štandardného hovoru, ako to už bolo v tejto dokumentácii vysvetlené pri časti Prijatý hovor. Počas presmerovaného hovoru klient oznamuje serverom pomocou NOTIFY, že hovor je ešte stále presmerovaný. Na tieto NOTIFY správy proxy server odpovie s OK správou. Hovor je ukončený so štýlom BYE a odpoveďou OK.



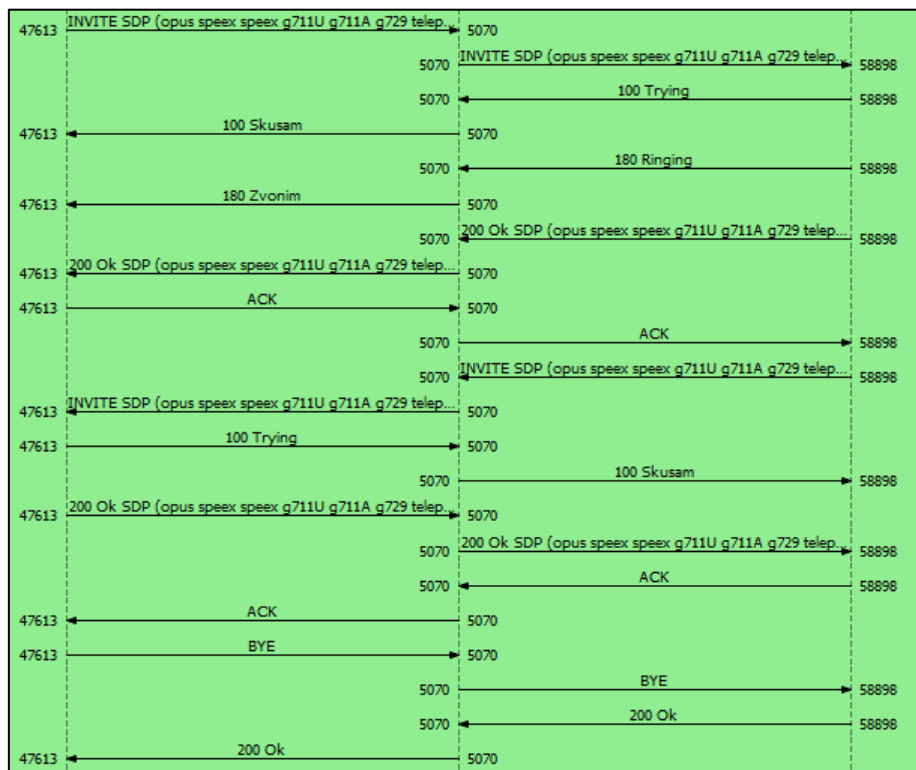
Obrázok 10 - Prenesený hovor

Videohovor

Pri hovore pomocou SIP protokolu je možné uskutočniť videohovor, pri ktorom účastníci začínajúci videohovor pošle správu INVITE(predmet: media change), ktorá obsahuje SDP protokol s kodekmi na zvuk a video. Predtým tieto SDP správy obsahovali len zvukové kodeky a z toho dôvodu sú nutné tieto INVITE správy. Druhý účastník hovoru dostane znamenie, že sa stala inicializácia videohovoru a po povolení tejto udalosti sa pošle správa OK

Adrian Szacsko
ID: 110899

s SDP protokolom obsahujúcim podporované kodeky na videohovor i na zvuk. Po obdržaní správy OK účastník inicializujúci videohovor pošle ACK a výsledkom je fungujúci videohovor.



Obrázok 11 - Videohovor