

SIP Proxy

Mobilné technológie a aplikácie

Matej Janeba

Boris Novák

2021/2022

Slovenská technická univerzita v Bratislave

Fakulta informatiky a informačných technológií

xnovakb@stuba.sk

Hlavná myšlienka zadania:

Na vašom počítači (alebo virtuálnom počítači) sprevádzkujte SIP Proxy, ktorá umožní prepájanie a realizáciu hovorov medzi štandardnými SIP klientami. Doplnujúce 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 dvoma 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.

Doplňkové 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"

Implementácia zadania

Využitie knižnice

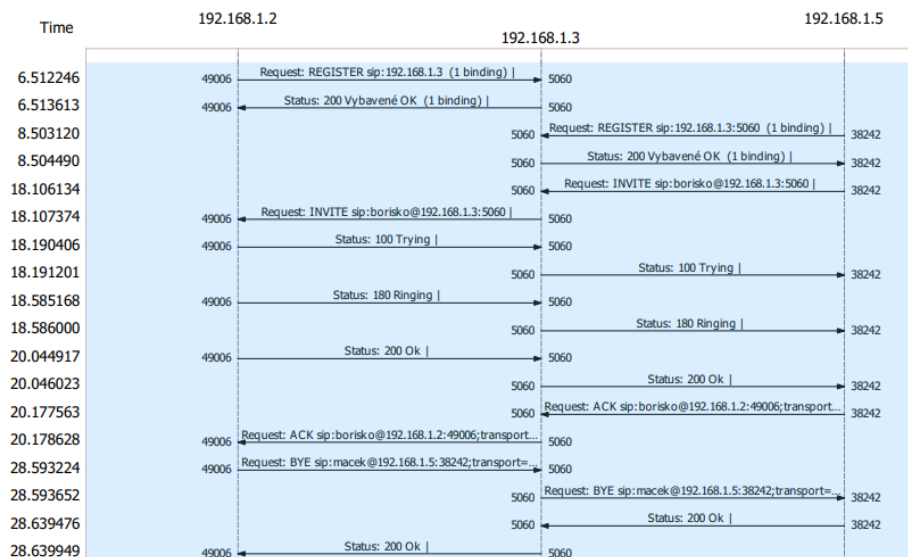
<https://github.com/tirfil/PySipFullProxy>

Implementácia

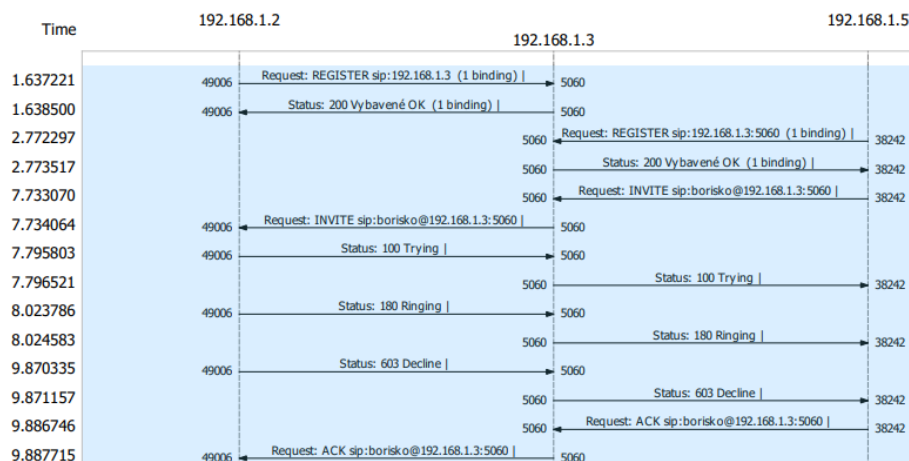
Na implementáciu SIP proxy som použil knižnicu PySipFullProxy, ktorú som použil vo svojom main.py, kde som zobral main funkciu z danej knižnice trochu ju upravil a použil u seba na spustenie SIP proxy

Minimálne požiadavky

Z **minimálnych požiadaviek** som **splnil všetky**, na obrázku môžeme vidieť **registráciu zariadení**, a odpoveď servera „200 Vybavené OK“, kde je splnená aj jedna doplnková funkcionálna a tou je prepísanie hlášky. Následne Hovor začal používateľ „macek“, kde poslal INVITE pre používateľa borisa. SIP Proxy následne preposlala INVITE macekovi. Po **prijatí hovoru** sa odošle od borisa „200 OK“ a po príchode macekovi sa odošle ACK. Pri **ukončení hovoru** borisom sa pošle BYE na maceka, ktorý mu späť pošle „200 OK“ po jeho príchode sa končí komunikácia.



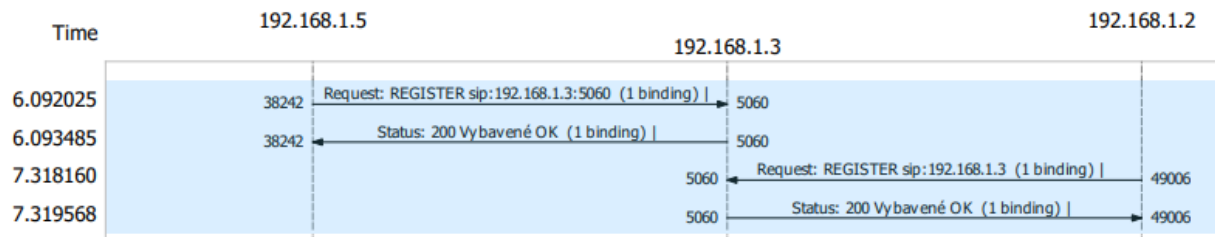
Tu je splnená posledná minimálna požiadavka: **odmietnutie hovoru**, kde po zvonení odošle boris „603 Decline“, po príchode správy macekovi sa odošle ACK na borisa a ukončuje sa komunikácia.



Doplňkové funkcionality

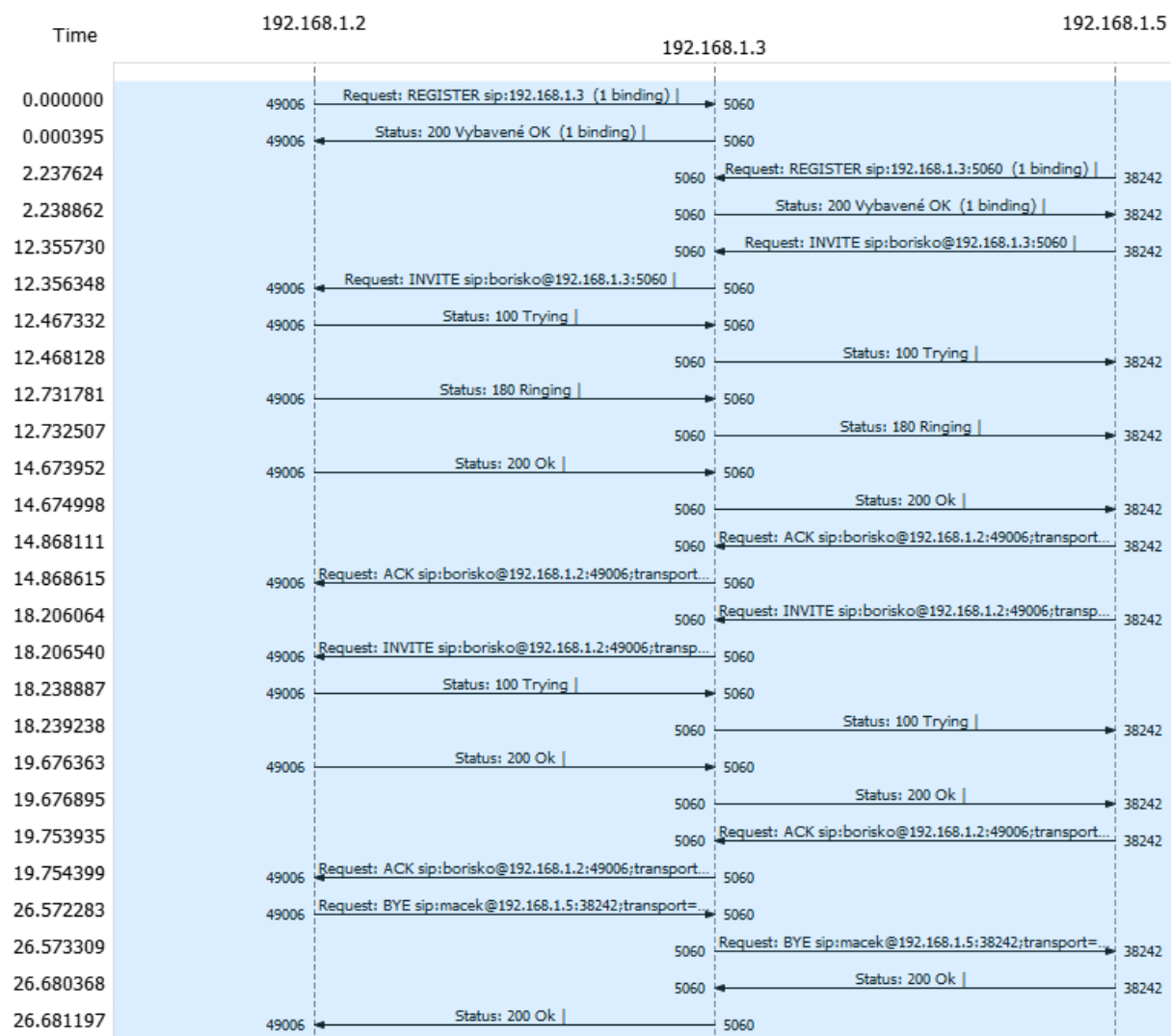
Prepísanie hlášky

Pri registrácii som prepísal stavový kód „200 OK“, na stavový kód „200 Vybavené OK“



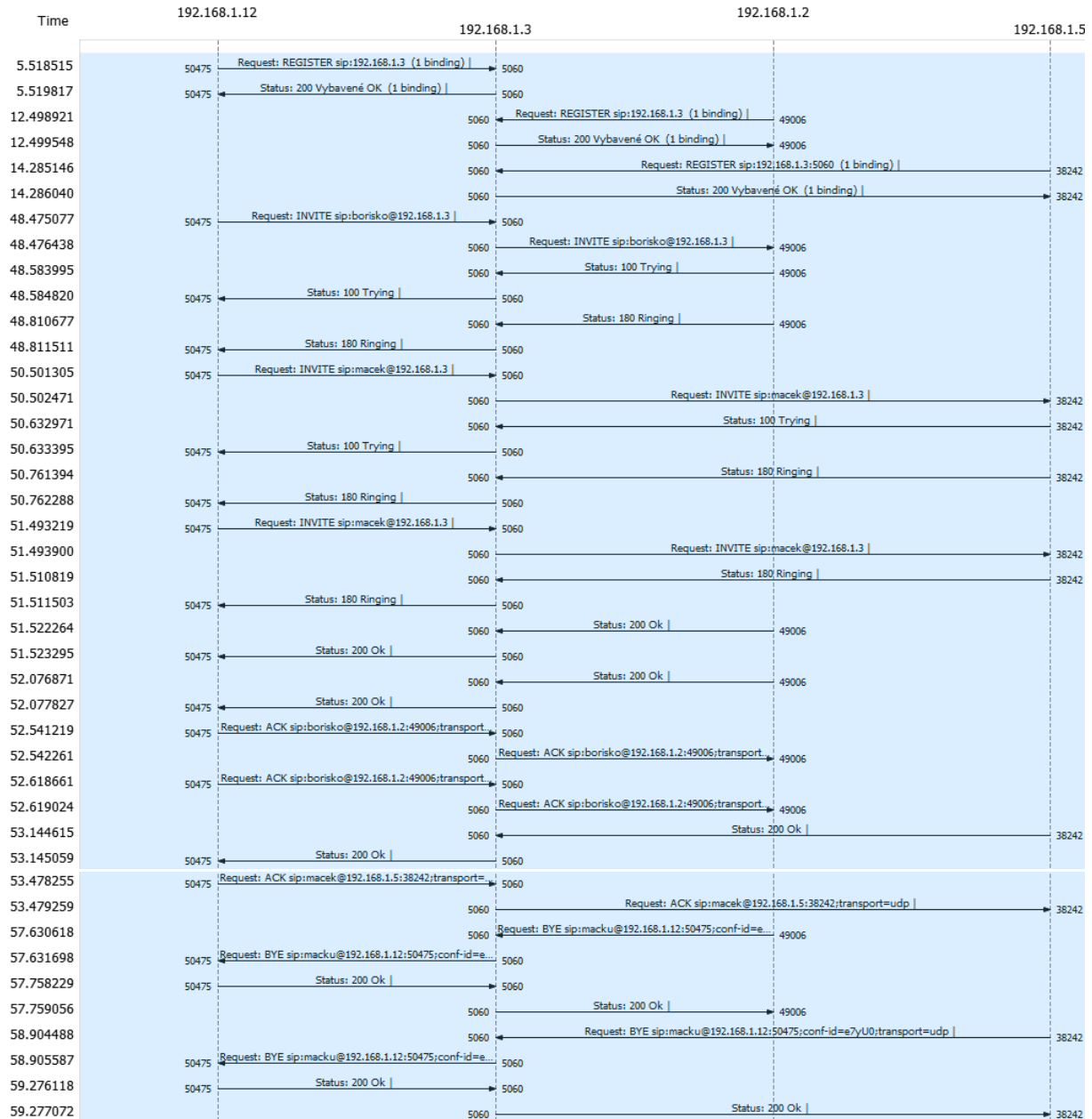
Videohovor

Pri videhovore som najskôr zaregistroval obidve zariadenia, macek a borisko. Následne macek poslal INVITE borisovi, kde po prijatí hovoru dostal spätne od borisa stavový kód „200 OK“, následne macek odoslal ACK. Pri požiadaní o videohovor, macek znova poslal INVITE borisovi a po jeho potvrdení prišiel macekovi stavový kód „200 OK“, znova macek odoslal ACK borisovi a videohovor sa začal. Pri ukončení hovoru borisom sa poslal BYE na maceka a on po jeho prijatí odoslal borisovi „200 OK“ a ukončila sa komunikácia.



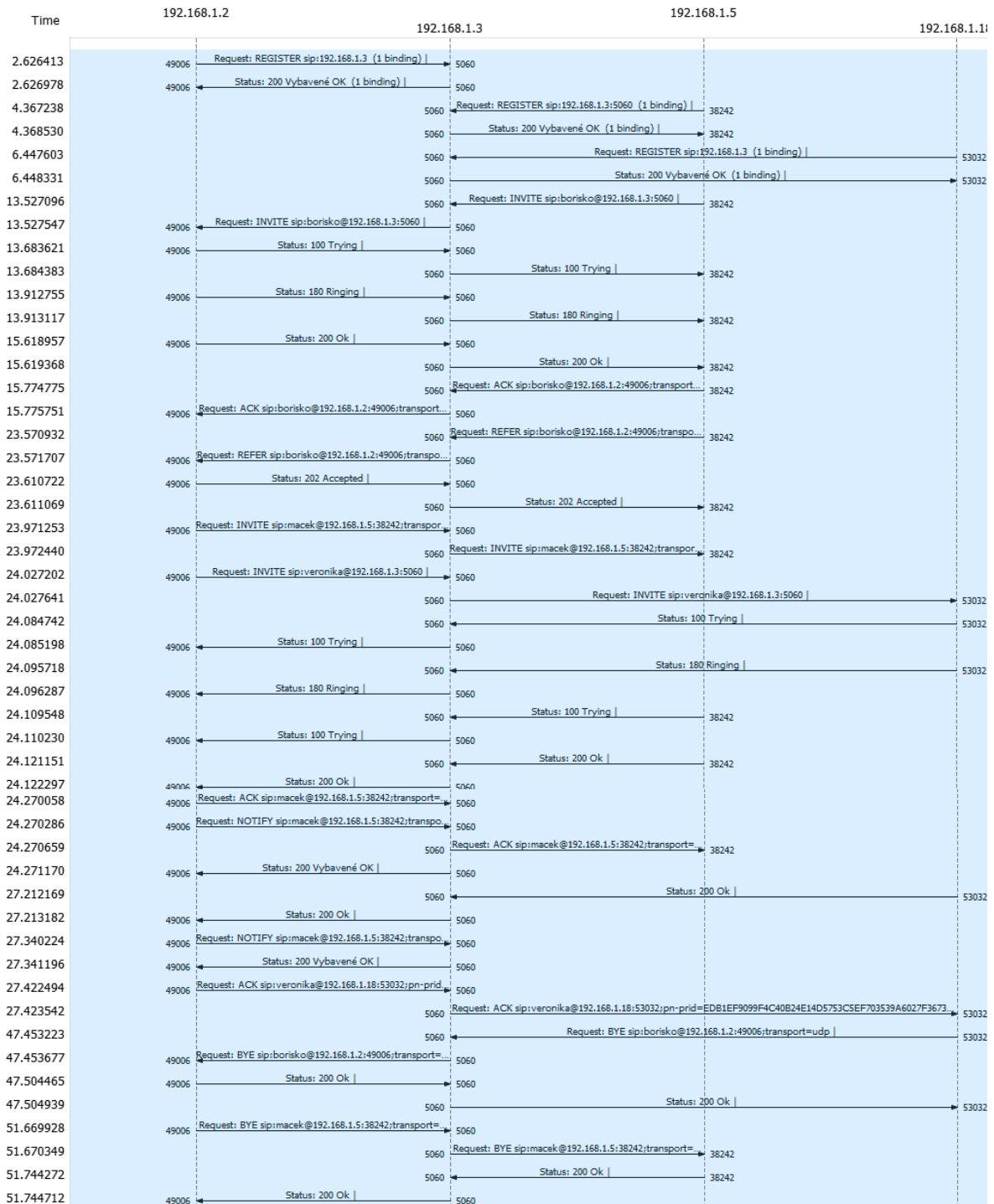
Konferenčný hovor

Pri konferenčnom hovore som zaregistroval používateľov: macek, borisko, macka. Následne macka začala konferenčný hovor a poslala INVITE na borisa a maceka, ktorí po prijatí hovoru odoslali späťne stavový kód „200 OK“. Následne macka odoslala ACK borisovi aj macekovi. Pri odpojení sa z konferenčného hovoru, poslal boris aj macek BYE na macka, on im následne poslal stavový kód „200 OK“ a ukončil s nimi komunikáciu.



Presmerovanie hovoru

Na začiatku som zaregistroval tri zariadenia maceka, boriska a veroniku. Na začiatku poslal macek INVITE borisovi, ktorý potvrdil hovor a macekovi došiel stavový kód „200 OK“, následne macek poslal spätne borisovi ACK a hovor začal. Následne poslal macek REFER o informácii, že transferuje volanie. Boris odpovedal „202 Accepted“ a následne poslal invite macekovi a veronike. Veronika zdvihla a odoslala spätne „200 OK“. Následne boris poslal macekovi ACK a NOTIFY na proxy aby zaregistroval Veroniku do hovoru a dostal od proxy odpoveď „200 Vybavené OK“. Následne sa poslalo ACK na Veroniku. Veronika ukončila hovor a poslala borisovi BYE on odoslal spätne OK a ukončil komunikáciu. Následne ukončil hovor aj boris a poslal BYE macekovi, ktorý mu spätne poslal „200 OK“ a ukončil komunikáciu.



Logovací súbor

Logovací súbor už bol v knižnici, ja som doň pridal hlášky, kedy sa daný hovor začal CALL STARTED : FROM – TO, ukončenie hovoru CALL ENDED a odmietnutie hovoru CALL DECLINED

```
16:54:30:INFO:Mon, 28 Feb 2022 16:54:30
16:54:34:INFO:Boľ zaregistrovaný nový používateľ: macka@147.175.160.245 [Adresa: 147.175.160.245:56666]
16:54:35:INFO:Boľ zaregistrovaný nový používateľ: macek@147.175.160.245 [Adresa: 147.175.162.228:38242]
16:54:44:INFO:Zvonenie macek@147.175.160.245 -> macka@147.175.160.245
16:54:45:INFO:Zaciatok hovoru macek@147.175.160.245 -> macka@147.175.160.245
16:54:48:INFO:Videohovor macek@147.175.160.245 -> macka@147.175.160.245
16:54:50:INFO:Koniec hovoru macka@147.175.160.245 -> macek@147.175.160.245
```

Github

[xNovakb/MTAA \(github.com\)](https://github.com/xNovakb/MTAA)

Zmeny v knižnici

import SocketServer -> import socketserver

dictionary.has_key(key) -> key in dictionary

v def handle(self) data = self.request[0] -> data = self.request[0].decode("utf-8")

viaceré miesta socket.sento(text, something) -> socket.sendto(text.encode("utf-8"), something)

vymazanie rx_invalid, rx_invalid2 a v kóde vymazanie if podmienky na ich kontrolu

vymazanie logging.info(), kvôli prerábke logera a doplnenie môjho kódu pre logovací denník

```
if "20 INVITE" in data[5] and "Ringing" in data[0]:
    logging.info("Zvonenie %s -> %s", self.getOrigin(), self.getDestination())
elif "21 INVITE" in data[5] and "Ok" in data[0]:
    logging.info("Videohovor %s -> %s", self.getOrigin(), self.getDestination())
elif "INVITE" in data[5] and "Ok" in data[0]:
    logging.info("Zaciatok hovoru %s -> %s", self.getOrigin(), self.getDestination())
elif "BYE" in data[5] and "Ok" in data[0]:
    logging.info("Koniec hovoru %s -> %s", self.getOrigin(), self.getDestination())
elif "INVITE" in data[5] and "Decline" in data[0]:
    logging.info("Zamietnutý hovor %s -> %s", self.getOrigin(), self.getDestination())
```

Záver

Na záver by som chcel dodať, že zo začiatku som si myslel, že toto zadanie sa nedá spraviť, ale ako som postupom času hľadal knižnicu a upravoval, kód tak ma to začalo baviť a postupné spojzdenie hovorov a odsledovanie sip komunikácie vo wiresharku mi ukázalo ako to všetko funguje a vďaka tomu mi toto zadanie dalo mnoho nových poznatkov