

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií

SIP Proxy (telefónna ústredňa)

Patrik Velčický

(MTAA zadanie 1 dokumentácia)

Cvičenie: Štvrtok 08:00

Cvičiaci: Ing. Peter Trúchly, PhD.

Rok: 2022

Implementácia

Na implementáciu SIP Proxy som použil kód z githubu s názvom SipFullProxy¹, ktorý bolo potrebné upraviť. Tento upravený kód spúšťam cez môj vlastný main.py. V ktorom získavam IP adresu serveru automaticky pomocou knižnice socket alebo manuálne (vyznačené zelenou farbou), inicializujem logovanie (vyznačené červenou farbou) a spúšťam SIP server pomocou SipFullProxy (vyznačené fialovou farbou).

```
logging.basicConfig(format='%(asctime)s: %(levelname)s: %(message)s', filename='proxy_call_log.log', level=logging.INFO,
                    datefmt='%Y-%m-%d %H:%M:%S')

ipaddress = str()

q = input("A - pre automaticke získanie IP adresy\nLubovolna klavesa - pre manualne zadanie IP adresy\n")

if q == 'A' or q == 'a':
    hostname = socket.gethostname()
    ipaddress = socket.gethostbyname(hostname)
    if ipaddress == "127.0.0.1":
        ipaddress = sys.argv[1]
else:
    ipaddress = input("Zadajte IP adresu servera: ")

print('IP Adresa SIP servera: ' + ipaddress)

sipfullproxy.recordroute = "Record-Route: <sip:%s:%d;lr>" % (ipaddress, sipfullproxy.PORT)
sipfullproxy.topvia = "Via: SIP/2.0/UDP %s:%d" % (ipaddress, sipfullproxy.PORT)
server = socketserver.UDPServer((sipfullproxy.HOST, sipfullproxy.PORT), sipfullproxy.UDPHandler)
server.serve_forever()
```

Ako prvé som upravil názov knižnice, ktorú bolo potrebné importovať z SocketServer na socetserver. Názov SocketServer je platný pre python 2 ale ja som používal python 3 pre, ktorý platí názov socketserver.

Následne bolo potrebné zmeniť ďalšiu syntax join. Ako môžeme vidieť na nasledujúcich obrázkoch.

```
text = string.join(data, "\r\n")
```

```
text = "\r\n".join(data)
```

Ako ďalšie som zmenil syntax pre dictionary z registrar.has_key(destination) na destination in registrar. Ako môžeme vidieť na nasledujúcich obrázkoch.

```
if registrar.has_key(destination)
```

```
if destination in registrar
```

Na to aby sme mohli použiť všetky IP adresy je potrebné zakomentovať nasledujúci kód.

```
# rx_invalid = re.compile("^192\.\.168")
# rx_invalid2 = re.compile("^10\.\.")
```

¹ <https://github.com/tirfil/PySipFullProxy/blob/master/sipfullproxy.py>

```
# if rx_invalid.search(contact) or rx_invalid2.search(contact):
#     if registrar.has_key(fromm):
#         del registrar[fromm]
#     self.sendResponse("488 Not Acceptable Here")
#     return
```

Následne bolo potrebné pridať enkódovanie správ vždy keď je správa odoslaná a dekodovanie správ keď je správa prijatá.

```
def handle(self):
    #socket.setdefaulttimeout(120)
    data = self.request[0].decode('utf-8')
    self.data = data.split("\r\n")
    self.socket = self.request[1]
    request_uri = self.data[0]

    text = "\r\n".join(data).encode('utf-8')
    self.socket.sendto(text, self.client_address)
```

Na zrealizovanie denníka hovorov bolo potrebné upraviť logovanie. Každý príkaz na logovanie levelu INFO som zakomentoval a vo funkcii handle som ho upravil tak aby registroval informácie, ktoré súvisia s hovormi. Tieto informácie ohľadom hovoru uchovávam v proxy_call_log.log.

```
request_uri = self.data[0]
if rx_request_uri.search(request_uri) or rx_code.search(request_uri):
    showtime()
    call = request_uri[:request_uri.find(" ")]
    if call in ['INVITE', 'BYE', 'CANCEL', 'ACK'] or request_uri == 'SIP/2.0 603 Decline':
        call_to = list(filter(lambda x: 'To: ' in x, self.data))
        call_from = list(filter(lambda x: 'From: ' in x, self.data))
        call_id = list(filter(lambda x: 'Call-ID: ' in x, self.data))
        logging.info(">>> " + request_uri + ' ' + call_to[0] + ' ' + call_from[0] + ' ' + call_id[0])
    logging.debug("---\n>> server received [%d]:\n%s\n---" % (len(data), data))
    logging.debug("Received from %s:%d" % self.client_address)
    self.processRequest()
```

Ako posledné som upravil stavový kód SIP z pôvodného 200 OK na 200 OK – V PORADKU.

187	10.294436	147.175.190.171	147.175.176.85	SIP	678	Request: REGISTER sip:147.175.176.85:5060 (1 binding)
188	10.294954	147.175.176.85	147.175.190.171	SIP	706	Status: 200 OK - V PORADKU (REGISTER) (1 binding)
310	15.324720	147.175.176.85	147.175.176.85	SIP	728	Request: REGISTER sip:147.175.176.85:5060 (1 binding)
311	15.325278	147.175.176.85	147.175.176.85	SIP	755	Status: 200 OK - V PORADKU (REGISTER) (1 binding)

Záver

Táto implementácia SIP proxy podporuje všetky funkcionality v zadaní. Môj kód spĺňa všetky body zo zadania povinné aj nepovinné.

Link na GIT repozitár

https://github.com/velco22/MTAA_1