

Mobilné technológie a aplikácie

Zadanie 1 – SIP Proxy

meno: Tomáš Kiss

AIS ID: 110818

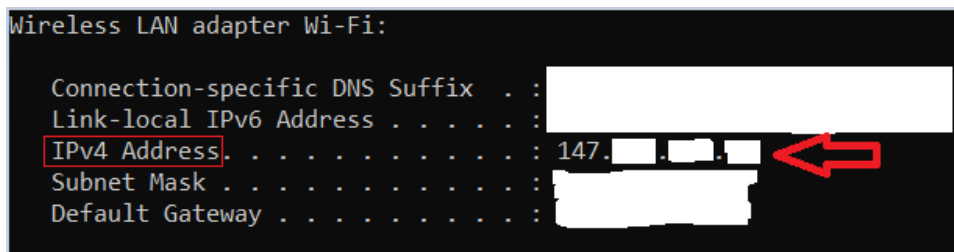
cvičenie: streda 12.00-13.40

cvičiaci: Ing. Marek Galinski, PhD.

V našom zadaní sme používali programovací jazyk Python s dostupnou verziou Python 3.10.0 . Rovnako bola v tomto zadaní použitá knižnica s názvom PySipFullProxy prevzatá z platformy github (<https://github.com/tirfil/PySipFullProxy>). Avšak táto knižnica bola písaná vo verzii pythone 2, takže bolo kód potrebné mierne doladiť na python 3. Okrem knižnice máme vytvorený aj súbor main.py z ktorého voláme tento prevzatý súbor (knižnicu) myprojectSIP.py .

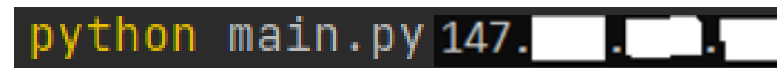
Spôsob spustenia

Na to aby bolo možné použiť náš program je potrebné zistiť akú máme ip adresu. Pre Windows je riešením otvoriť si cmd tam zadať príkaz ipconfig pre zistenie ip adresy zariadenia, na ktorom chcete spustiť proxy server. V prípade, pripojenia pomocou Wi-Fi skopírovať <ip_adresu> nachádzajúcu sa ako na obrázku na mieste -> IPv4 Address .



Ďalej program spustíme cez terminál príkazom:

python main.py <ip_adresa>



kde sa zadáva ip adresa počítača na ktorom má bežať SIP proxy.

SIP client

Pri testovaní hovorov a vytváraní pcap súborov pre scenáre sme používali Linphone čo je free softphone. Voľne dostupný na linku -> (<https://www.linphone.org/>) Týmto klientom je potrebné sa pripojiť na náš sip server spolu ešte s jedným zariadenia na splnenie minimálnych požiadaviek. Avšak na implementovanie scenárov ktoré máme zrealizované my je potrebných viac zariadení aspoň pre konferenčný hovor kde bolo treba troch účastníkov alebo na presmerovanie hovoru. My sme v našej implementácii splnili všetky scenáre okrem scenáru číslo 8 spomenutého ďalej v dokumentácii.

Knižnice

Použili sme nasledovné knižnice: **socket, socketserver, re, time, logging, sys** a v neposlednej rade aj prevzatú knižnicu PySipFullProxy, ktorá je v súbore myprojectSIP.py .

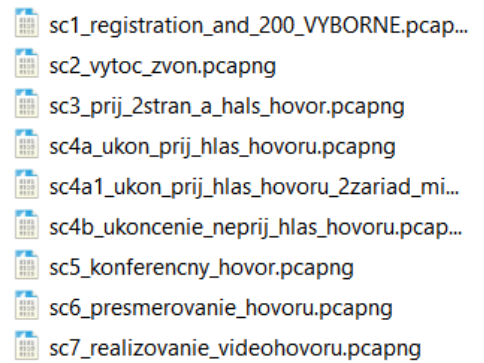
Priložené pcap súbory

Každý ďalší scenár sa nachádza v súbore sc(číslo scenára)_(specifikačné meno súboru).pcapng, teda tak ako idú po rade. ---- >

Podľa zadania sme mali splniť niekoľko scenárov

1. Registrácia účastníka (bez nutnosti autentifikácie)
Toto je možné nájsť v súbore sc1_registration_and_200_VYBORNE.pcapng .
2. Vytočenie hovoru a zvonenie na druhej strane
3. Prijatie hovoru druhou stranou, fungujúci hlasový hovor
4. Ukončenie hlasového hovoru (prijatého aj neprijatého) – **delí sa na dva**
Tento scenár je rozdelený na 3 súbory, jeden ktorý je pre ukončenie neprijatého hovoru a dva sú pre ukončenie prijatého hovoru. Tie pre ukončenie prijatého hovoru boli nahrávané jeden mimo zariadenia na ktorom bežal SIP proxy (sc4_(nazov)) tu je možné si všimnúť že sa nenachádza RTP komunikácia lebo wireshark nezachytáva rtp (real time protocol), nakoľko tento protokol beží na p2p komunikáčnej úrovni a druhý bol medzi dvoma klientami na jednom bežal proxy server (sc4a_(nazov))
5. Možnosť zrealizovať konferenčný hovor (aspoň 3 účastníci)
6. Možnosť presmerovať hovor
7. Možnosť realizovať video hovor
8. **Logovanie “denníka hovorov” – kto kedy komu volal, kedy bol ktorý hovor prijatý, kedy bol ktorý hovor ukončený, do ľubovoľného textového súboru v ľubovoľnom formáte**
9. Úprava SIP stavových kódov v zdrojovom kóde proxy, napr. “486 Busy Here” zmeníte na “486 Obsadené”
V súbore pre scenár jedna sc1_registration_and_200_VYBORNE.pcapng sa nachádza upravená SIP správa 200 VYBORNE z pôvodného 200 OK .

***Červeným sú vysvietené požiadavky ktoré sa v našej implementácii nenachádzajú**



- sc1_registration_and_200_VYBORNE.pcap...
- sc2_vytoc_zvon.pcapng
- sc3_prij_2stran_a_hals_hovor.pcapng
- sc4a_ukon_prij_hlas_hovoru.pcapng
- sc4a1_ukon_prij_hlas_hovoru_2zariad_mi...
- sc4b_ukoncenie_neprij_hlas_hovoru.pcap...
- sc5_konferencny_hovor.pcapng
- sc6_presmerovanie_hovoru.pcapng
- sc7_realizovanie_videohovoru.pcapng

Github

Odkaz na môj verejný repozitár: https://github.com/TomasKiss18/mtaa_zadanie1