Dokumentácia

Počítačové a Komunikačné siete

Analyzátor sieťovej komunikácie

Jakub Martinák

Faculty of Informatics and Information Technology, Slovak Technical University

Obsah

Abstrakt

Tento projekt sa zameriava na analýzu sieťovej komunikácie na rôznych vrstvách. Rieši sa rozdeľovanie komunikácie medzi ukončené a neukončené, filtrovanie podľa zadaného parametru, celková analýza hexdumpu z pcap súboru.

1 Úvod

V dnešnej dobe, keď technológie neustále pokračujú v rýchlom tempe a siete sú základom takmer všetkého, čo robíme, je kľúčové rozumieť podrobnostiam sieťovej komunikácie. Tento projekt je venovaný hĺbkovej analýze sieťovej komunikácie na rôznych vrstvách. Skúma a rieši kritické aspekty tejto oblasti, ako je rozlišovanie medzi ukončenými a neukončenými komunikáciami, prispôsobené filtrovanie informácií na základe konkrétnych parametrov a podrobná analýza hexdumpov extrahovaných z pcap súborov.

2 Diagramy jednotlivých funkcií

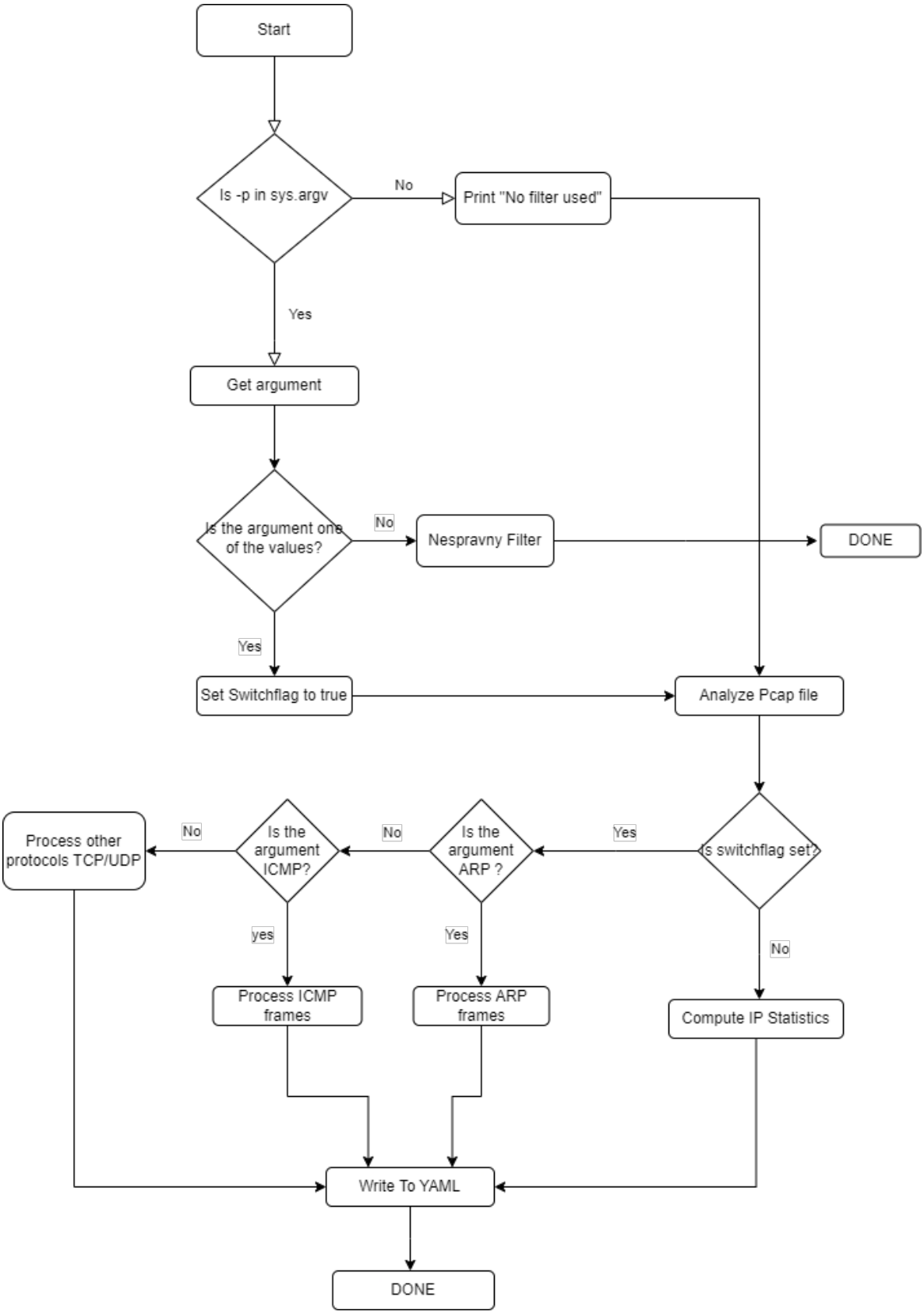
2.1 Hlavná funkcia: `'__main__'`:

Zhrnutie: Hlavná funkcia začína výpisom "STARTING". Potom definuje cestu k súboru a niekoľko premenných a flagov. Funkcia overuje, či je v argumentoch príkazového riadku zadaná možnosť '-p' a na základe toho vykonáva rôzne filtračné operácie na údajoch PCAP. Nakoniec tieto údaje zapíše do súboru vo formáte YAML a vypíše "DONE".

2.1.1 Podrobný popis:

1. Začne s printovaním reťazca "STARTING".
2. Definuje cestu k súboru PCAP.
3. Overuje, či je zadaná voľba '-p' v argumentoch príkazového riadku:
   - Ak je, pokračuje v overovaní, či nasledujúci argument je platný filter. V prípade chýb vypíše chybové hlásenia a ukončí program.
   - Ak nie je, tlačí "No filter used".
4. Analyzuje dáta PCAP s použitím funkcie `analyze_pcap()`.
5. Na základe hodnoty flagu `switch_flag` a hodnoty argumentu vykonáva rôzne filtračné a spracovateľské operácie.
6. Nakoniec upraví výsledné údaje do formátu YAML a uloží ich do súboru.
7. Funkcia končí printovaním reťazca "DONE".

Diagram tejto funkcie je možný vidieť na Obr. 1



Obr. 1: Diagram Hlavnej Funkcie

2.2 Funkcia: `analyze_eth_packet(number, pckt, filter_flag)`

Zhrnutie: Táto funkcia analyzuje paket Ethernetu, určuje jeho typ (napr. ETHERNET II, ARP, TCP, UDP, ICMP, IEEE 802.3 RAW, IEEE 802.3 LLC, IEEE 802.3 LLC & SNAP) a extrahuje rôzne detaily týkajúce sa paketu, ako sú MAC adresy, IP adresy, porty, protokoly, príznaky a iné špecifické polia.

Parametre:

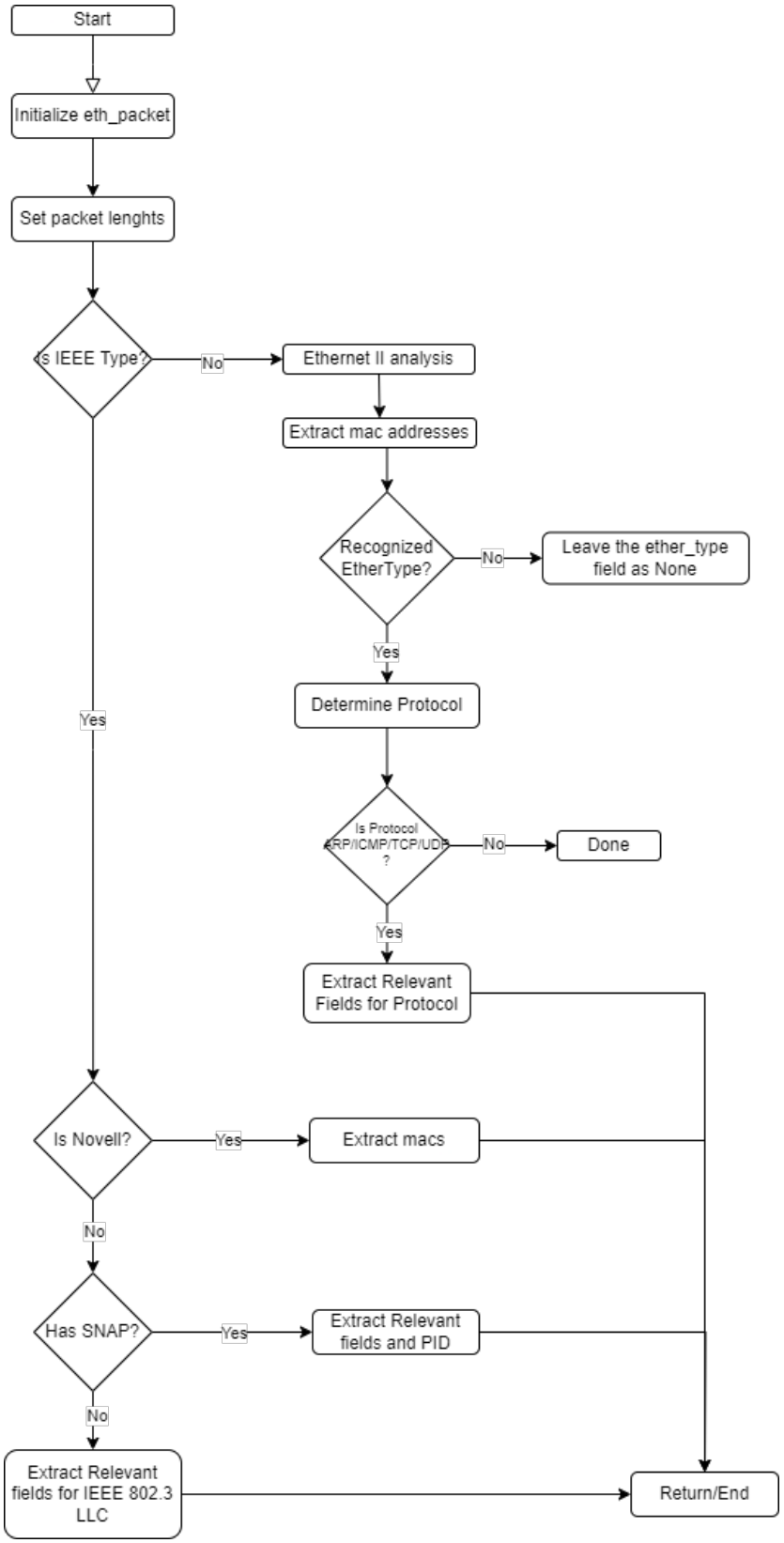
- `number (int)`: Číslo rámca paketu.
- `pckt (packet)`: Samotný paket.
- `filter_flag (bool)`: flag na určenie, či filtrovať a spracovávať konkrétne protokoly (napr. detaily TCP/UDP).

Return:

- `dict`: dict obsahujúci extrahované informácie z paketu.

Detaily:

1. **Inicializácia**:
   - Nastavenie predlohy dict pre paket Ethernetu s predvolenými poliami a hodnotami.
2. **Spracovanie každého paketu**:
   - Výpočet dĺžok rámcov v pcap a médiu.
   - Určenie, či je paket typu ETHERNET II na základe hodnoty v bajtoch 12-14.
3. **Spracovanie ETHERNET II**:
   - Ak je to ETHERNET II, extrahuje sa MAC adresa, a určuje sa EtherType (ARP, IP atď.).
   - Pre ARP extrahuje detaily ARP (napr. opcode, zdrojová a cieľová IP).
   - Pre nie-ARP extrahuje detaily IP a ďalšie podrobnosti pre TCP alebo UDP.
4. **Spracovanie IEEE 802.3 RAW, LLC a SNAP**:
   - Ak to nie je ETHERNET II, určuje sa, či je to IEEE 802.3 RAW, LLC alebo SNAP a extrahuje sa zodpovedajúce detaily.
5. **Return**:
   - Vracia sa konštruovaný dict `eth_packet` so všetkými extrahovanými detailmi.



Obr. 2: Diagram analyze_eth_packet funkcie

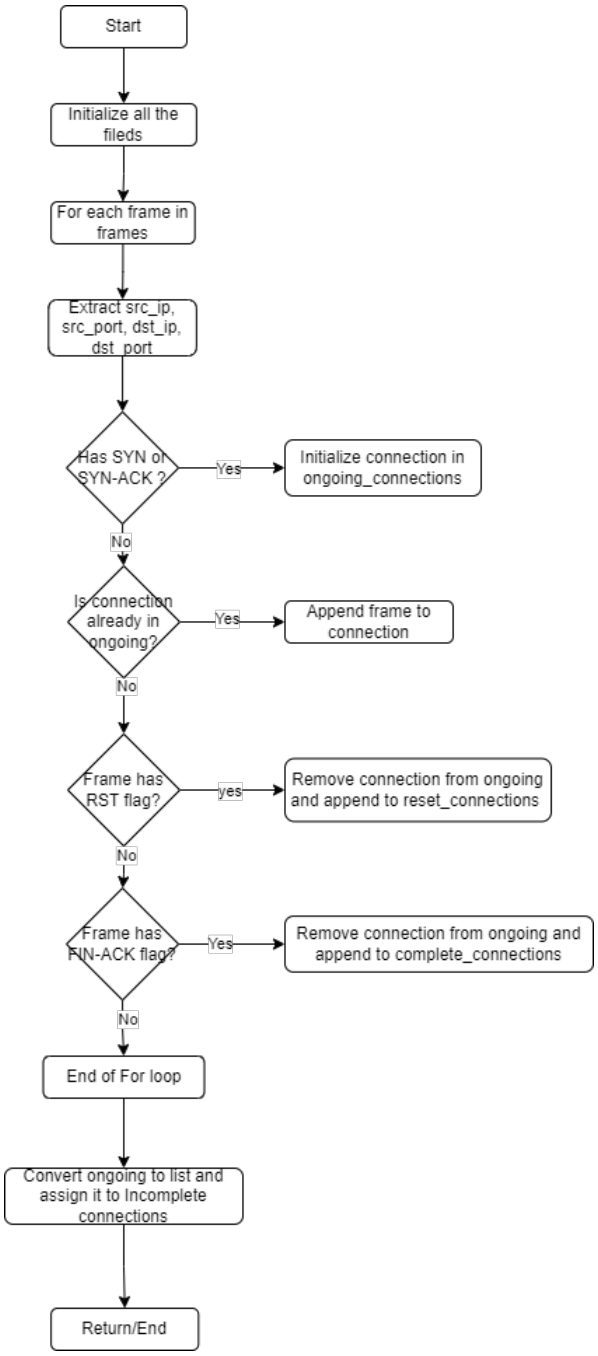
2.3 Funkcia: `track_connections(frames)`

Zhrnutie:

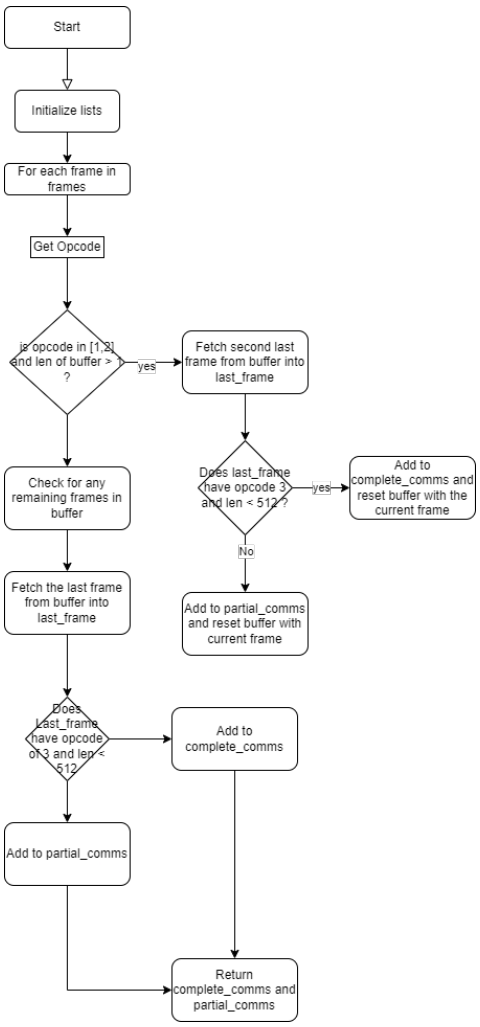
Funkcia `track_connections` spracúva zoznam rámcov na sledovanie stavu sieťových pripojení. Očakáva sa, že každý rámec bude mať polia `'src_ip'`, `'src_port'`, `'dst_ip'`, `'dst_port'` a `'flags'`. Funkcia rozdeľuje pripojenia do troch kategórií:

- **Ukončené pripojenia:** Pripojenia, ktoré boli úspešne ukončené s príznakom 'FIN-ACK'.
- **Resetované pripojenia:** Pripojenia, ktoré boli predčasne ukončené s príznakom 'RST'.
- **Nedokončené/prebiehajúce pripojenia:** Pripojenia, ktoré neboli ani ukončené, ani resetované na konci zoznamu rámcov.

Funkcia vráti dict s vyššie uvedenými kategóriami ako kľúčmi a zoznamom pridružených rámcov ako hodnoty.



Obr. 3: Diagram track_connections funkcie

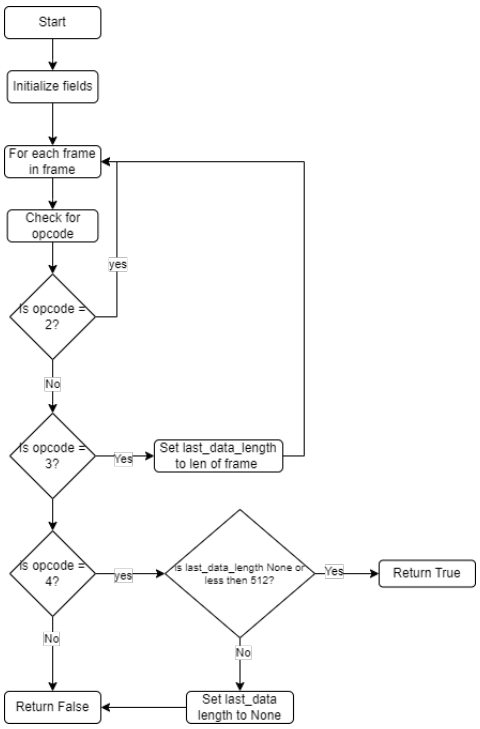


Obr. 4: Diagram categorize_tftp_frames funkcie

2.4 funkcia: `categorize_tftp_frames(ramce)`

Zhrnutie: Funkcia `categorize_tftp_frames` rozdeľuje rámce TFTP do dvoch kategórií: kompletných a neúplných komunikácií. Funkcia identifikuje začiatok a koniec každej komunikácie a pridáva ju do príslušnej kategórie.
Podrobný popis:

1. Inicializuje dočasný zásobník pre ukladanie rámcov.
2. Prechádza všetkými rámce a kontroluje operčné kódy.
3. Ak nájde novú požiadavku na čítanie alebo zápis a v zásobníku sú už rámce, rozhodne sa, či predchádzajúca komunikácia bola kompletná alebo neúplná.
4. Priradí komunikáciu k príslušnej kategórii.
5. Na konci funkcie kontroluje zvyšné rámce v zásobníku a priradí ich k príslušnej kategórii.

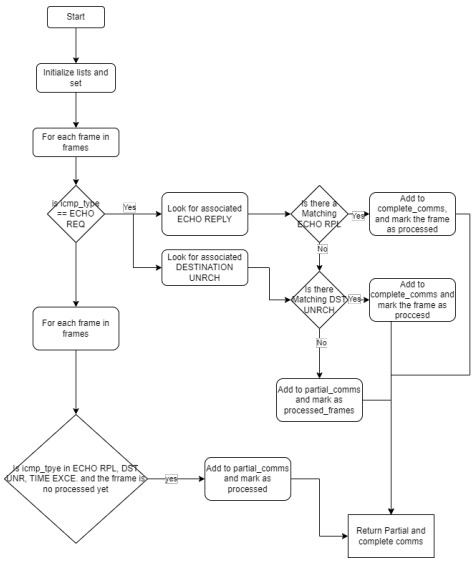


Obr. 5: Diagram is_tftp_complete funkcie

2.4.1 funkcia: `is_tftp_complete(ramce)`

Zhrnutie: Funkcia `is_tftp_complete` overuje, či je komunikácia TFTP kompletná alebo či je neúplná. Ak je posledný dátový rámec kratší ako 512 bajtov alebo chýba úplne, predpokladá sa, že komunikácia je kompletná. V opačnom prípade je komunikácia považovaná za neúplnú.
Podrobný popis:

1. Kontroluje operčný kód rámca.
2. Ak je operčný kód pre zápis, pokračuje v spracovaní.
3. Kontroluje operčný kód pre dáta a získava dĺžku dát.
4. Kontroluje operčný kód pre potvrdenie.
5. Ak je posledná dĺžka dát kratšia ako 512 bajtov, komunikácia sa považuje za kompletnú.
6. V opačnom prípade sa predpokladá, že komunikácia je neúplná.



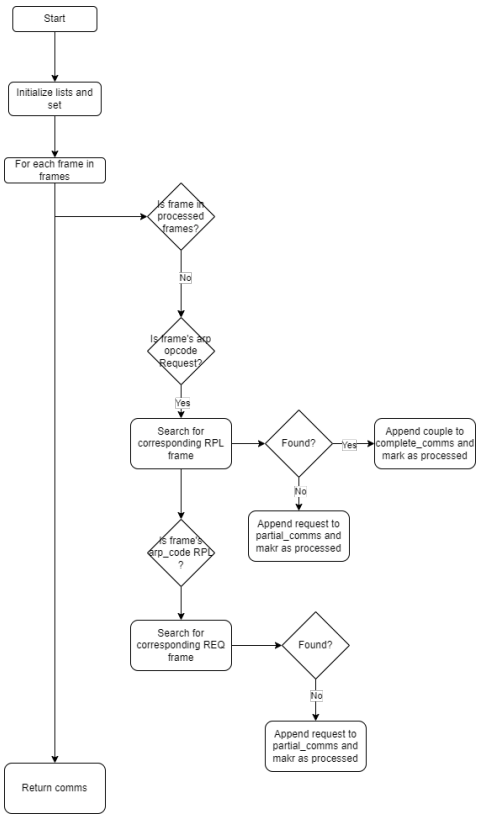
Obr. 6: Diagram process_icmp_frames funkcie

2.5 Funkcia: `process_icmp_frames(frames)`

Zhrnutie:
Funkcia `process_icmp_frames` spracováva zoznam rámcov ICMP a rozdeľuje ich na kompletné a čiastočné komunikácie. Ak rámec je typu ËCHO REQUEST", funkcia hľadá zodpovedajúci ËCHO REPLYälebo "DESTINATION UNREACHABLE"rámec. Ak nájde zodpovedajúci rámec, považuje komunikáciu za kompletnú, v opačnom prípade za čiastočnú. Nakoniec funkcia prechádza zoznamom rámecov a pridáva všetky nepracované ËCHO REPLY", "DESTINATION UNREACHABLEä "TIME EXCEEDED"rámce do čiastočných komunikácií.

2.5.1 Podrobný popis:

1. Začne s iteráciou cez zadané rámce.
2. Ak je rámec typu ËCHO REQUEST":
   - Hľadá zodpovedajúci ËCHO REPLYälebo "DESTINATION UNREACHABLE"rámec.
   - Ak nájde zodpovedajúci rámec, pridá túto komunikáciu do zoznamu kompletných komunikácií.
   - Ak nenájde zodpovedajúci rámec, pridá túto komunikáciu do zoznamu čiastočných komunikácií.
3. Po spracovaní všetkých rámecov, funkcia prechádza zoznamom rámecov znova a pridáva všetky nepracované rámce s typmi ËCHO REPLY", "DESTINATION UNREACHABLEä "TIME EXCEEDED"do čiastočných komunikácií.
4. Vracia dict s kompletnými a čiastočnými komunikáciami.



Obr. 7: Diagram process_arp_frames funkcie

2.6 Funkcia: `process_arp_frames(frames)`

Zhrnutie:
Funkcia `process_arp_frames` spracováva zoznam rámecov ARP a rozdeľuje ich na kompletné a čiastočné komunikácie. Ak je rámec typu "REQUEST", funkcia vyhľadáva zodpovedajúci rámec "REPLY". Ak nájde zodpovedajúci rámec "REPLY", považuje komunikáciu za kompletnú. Ak nenájde zodpovedajúci rámec alebo rámec je typu "REPLY"bez predchádzajúceho "REQUEST", považuje komunikáciu za čiastočnú.

2.6.1 Podrobný popis:

1. Začne s iteráciou cez zadané rámce.
2. Ak je rámec ešte nebol spracovaný a je typu "REQUEST":
   - Hľadá zodpovedajúci rámec "REPLY".
   - Ak nájde zodpovedajúci rámec, pridá túto komunikáciu do zoznamu kompletných komunikácií.
   - Ak nenájde zodpovedajúci rámec, pridá túto komunikáciu do zoznamu čiastočných komunikácií.
3. Ak je rámec typu "REPLY", funkcia hľadá zodpovedajúci rámec "REQUEST":
   - Ak nenájde zodpovedajúci rámec, pridá aktuálny "REPLY"rámec do zoznamu čiastočných komunikácií.
4. Vracia slovník s kompletnými a čiastočnými komunikáciami.

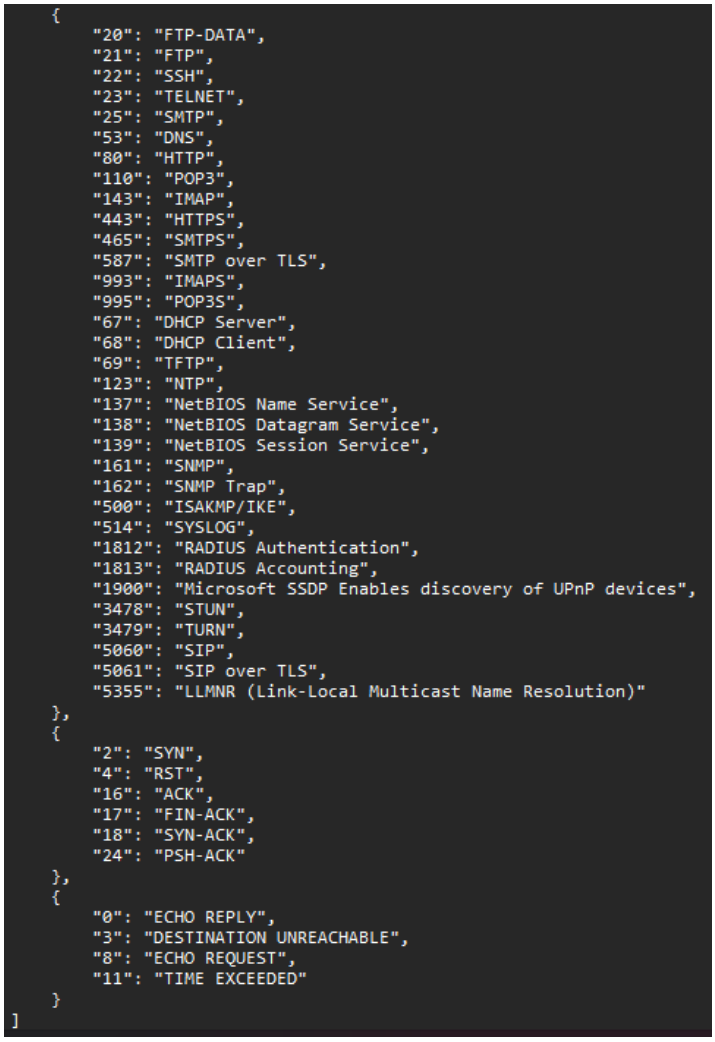
3 Štruktúra Externých Súborov

Externý súbor s názvom *constants.txt* obsahuje všetky potrebné hodnoty na riešenie daného zadania. Obashuje Ether Typy, Sapy, IP Protokoly, PID, Porty, základné flagy komunikácie a ICMP typy.

Externý súbor vyzerá následovne:


```
[
    {
        "2048": "IPv4",
        "2049": "X.75 Internet",
        "2053": "X.25 Level 3",
        "2054": "ARP",
        "32821": "Reverse ARP",
        "32923": "Appletalk",
        "33011": "AppleTalk AARP (Kinetics)",
        "33024": "IEEE 802.1Q VLAN-tagged frames",
        "33079": "Novell IPX",
        "34525": "IPv6",
        "34827": "PPP",
        "34887": "MPLS",
        "34888": "MPLS wit upstream-assigned label",
        "34915": "PPPoE Discovery Stage",
        "34916": "PPPoE Session Stage",
        "35020": "LLDP",
        "36864": "Loopback (0x9000)"
    },
    {
        "0": "Null SAP",
        "2": "LLC Sublayer Management / Individual",
        "3": "LLC Sublayer Management / Group",
        "6": "IP (DoD Internet Protocol)",
        "14": "PROWAY (IEC 955) Network Management, Maintenance and Installation",
        "66": "STP",
        "78": "MMS (Manufacturing Message Service) EIA-RS 511",
        "94": "ISI IP",
        "126": "X.25 PLP (ISO 8208)",
        "142": "PROWAY (IEC 955) Active Station List Maintenance",
        "224": "IPX(Novell Netware)",
        "240": "NetBIOS",
        "244": "LAN Management",
        "254": "ISO Network Layer Protocols"
    },
    {
        "1": "ICMP",
        "2": "IGMP",
        "6": "TCP",
        "9": "IGRP",
        "17": "UDP",
        "47": "GRE",
        "50": "ESP",
        "51": "AH",
        "57": "SKIP",
        "88": "EIGRP",
        "89": "OSPF",
        "115": "L2TP"
    },
    {
        "267": "PVSTP+",
        "768": "XEROX NS IDP",
        "8192": "CDP",
        "8196": "DTP",
        "12320": "VTP",
        "32923": "Appletalk",
        "33011": "Apple Talk AARP"
    },
```


Obr. 8: Externy subor cast 1



Obr. 9: Externy subor cast 2

4 Používateľské rozhranie

Používateľské rozhranie sa čisto skladá iba z konzolového vstupu a ľahkej úpravy vstupu súboru pcap na úplnom začiatku skritu. Po nastavení požadovaného súbor užívateľ v danom adresári zadá do konzole príkaz *python ./analyzer.py* a ak si žiada použiť aj filter použije prepínač -p nasludúc filtrom napr.(HTTP, HTTPS, SSH ...).

5 Implementačné prostredie

Skript bol robený v prostredí programu s názvom VSCODE a porgramovacím jazykom Python(3.12.0)
VSCODE je spravovaný spoločnosťou Microsoft.

6 Zhodnotenie a Rozšírenie

V rámci projektu sme sa podrobne venovali analýze sieťovej komunikácie na viacerých vrstvách. Jeho hlavným cieľom bolo rozlíšiť medzi ukončenou a neukončenou komunikáciou, čo v praxi umožňuje lepšie pochopiť a sledovať sieťové interakcie a potenciálne problémy. Ďalšou základnou funkciou projektu bolo filtrovanie komunikácie podľa zadaného parametru. Táto funkcia je obzvlášť užitočná pri práci s veľkými datasetmi, kde je potrebné rýchlo identifikovať a zameriavať sa na špecifické typy komunikácie.

Projekt tiež obsahoval rozsiahlu analýzu hexdumpu z pcap súborov. Hexdump je nástroj, ktorý poskytuje podrobný pohľad na dáta v binárnej forme, čo môže byť neoceniteľné pri hĺbkovej analýze a debugovaní sieťovej komunikácie.

Je však dôležité zdôrazniť, že implementácia fragmentácie ICMP packetov nebola súčasťou tohto projektu. ICMP fragmentácia je komplexný proces, ktorý by si vyžadoval dodatočné zdroje a čas na správnu implementáciu. Aj napriek tejto obmedzenosti projekt poskytuje solídne základy a nástroje na analýzu sieťovej komunikácie a môže slúžiť ako základ pre ďalší vývoj v tejto oblasti.

Rozšírenie je možné dosiahnúť doimplementáciou fragmentácie ICMP packetov a tým by tento projekt bol kompletný.