**Slovenská technická univerzita v Bratislave**

Fakulta informatiky a informačných technológií

**Počítačové a komunikačné siete**

**Analyzátor sieťovej komunikácie**

Contents

[1. Zadanie úlohy 4](#_Toc148392258)

[2. Štruktúra súborov na prácu s portami a protokolmi 5](#_Toc148392259)

[3. Práca s programom 5](#_Toc148392260)

[4. Flowchart načítania programu a určenia typu 7](#_Toc148392261)

[5. Flowchart pre Ethernet II atribúty 8](#_Toc148392262)

[6. Komunikácie 9](#_Toc148392263)

[7. Zvolené prostredie 10](#_Toc148392264)

[8. Záver a možné zlepšenie programu 10](#_Toc148392265)

# Zadanie úlohy

Návrh a implementácia analyzátora Ethernet siete na záznam komunikácie v súbore .pcap a poskytovanie nasledujúcich informácií o komunikácii. Výstup sa exportuje do formátu YAML.

**Funkcionalita:**

* Zoznam všetkých rámcov v hexadecimálnom tvare postupne, ako boli zaznamenané v súbore.
* Zoznam IP adries a zapuzdreného protokolu na vrstvách 2-4 pre rámce Ethernet II.
* Poskytnite nasledujúcu štatistiku pre pakety IPv4 na konci výstupu z úlohy 2:
* Program s analýzou komunikácie pre vybrané protokoly.

Tento popis zabezpečuje, že analyzátor bude schopný spracovať záznamy komunikácie v súbore .pcap a poskytne informácie jednotlivých rámcoch. Výstup bude vo formáte YAML, ktorý je ľahko spracovateľný pre ďalšiu analýzu alebo spracovanie dát.

**Výpis informácií rámca:**

* Poradové číslo rámca
* Dĺžka rámca
* Typ rámca
* Sap (IEEE 802.3 LLC)
* Pid (IEEE 802.3 LLC + SNAP)
* Zdrojová MAC adresa
* Cieľová MAC adresa
* Ethertype
* Zdrojová IP adresa
* Cieľová IP adresa
* Protokol
* Zdrojový port
* Cieľový port
* Aplikačný protokol
* Dáta v hexadecimálnom tvare

Taktiež, okrem výpisu je potrebné implementovať:

* Načítanie dát pre prácu s protokolmi a portami z externých súborov
* Štatistiku IP adries
* Filter

# Štruktúra súborov na prácu s portami a protokolmi

Z projektového adresára majú tieto súbory cestu /type\_files/‘subor’.txt

Ether\_type.txt:

ether\_type:  
 0806: ARP  
 0800: IPv4  
 88CC: LLDP  
 86DD: IPv6  
 9000: ECTP

# Práca s programom

Po spustení programu sú používateľovi do konzole vypísané všetky možné súbory .pcap, ktoré môže načítať.

A screen shot of a computer program

Description automatically generated

Následne si program vypýta od používateľa vstup, a to je meno súboru bez koncovky .pcap.

A screen shot of a computer

Description automatically generated

Ako ďalšie si program od vstup pre filter.

Pri stlačení klávesy ENTER keď si program vypýta vstup pre filter, sa filter uloží ako prázdny string, čo indikuje klasický výpis (Zadanie 1) pre určený .pcap súbor

Následne už program robí analýzu rámcov podľa zadaného filtra a výpis do .yaml súboru, ktorý sa vytvára do projektového adresára s názvom output.txt.

# A black screen with white text Description automatically generatedFlowchart načítania programu a určenia typu

# Flowchart pre Ethernet II atribúty

A diagram of a computer

Description automatically generated

# Komunikácie

* Komunikácia TCP protokolov (Connection-oriented protocol communication):

Ako prvé sa z vyfiltrovaných rámcov urobia páry portov (tuple), ktoré sa uložia do listu. Následne sa pre každú komunikáciu overia „handshake-y“. 3-way handshake otvára komunikáciu, a preto sa overuje ako prvý. Začne for each cyklus rámcov, kde sa overuje, či sú z streamu momentálnej komunikácie, a ak áno, pozerá sa 48. bajt v hexadecimálnom kóde rámca. Následne sa podľa stavových flagov overujú postupne bajty 2, 18, 16 = SYN, SYN/ACK, ACK.

Pokiaľ sa úspešne našli tieto 3 bajty, nastavíme flag pre 3-way handshake na true a ideme overovať 4-way closing handshake.

Pre 4-way handshake som zvolil inú stratégiu na overovanie, kvôli množstvu kombinácií, ktorými sa môže komunikácia ukončiť.

Najprv sa for each cyklom prejdú všetky rámce, kde overujeme, či sú z streamu komunikácie. Následne sa pre každý rámec odstrihne z hexadecimálneho kódu 48. bajt, ktorý sa prilepí k pomocnému stringu. Nasledovný string sa rozdelí, a získavame pole bajtov, ktoré taktiež uložíme odzadu, aby sa ľahšie pristupovalo k hodnotám, ktoré sú z konečných rámcov. Kombinácie ktoré sú v if podmienkach obsiahnuté, sú:

* RST
* RST/ACK
* FIN/ACK,FIN/ACK
* FIN/ACK/PUSH,FIN/ACK/PUSH,ACK
* FIN/ACK,ACK,FIN/ACK,ACK
* FIN,FIN/ACK,ACK

Ak sa potvrdí ukončenie komunikácie, flag pre 4-way handshake sa nastaví na True a metóda vracia nasledovne:

if three\_way\_handshake\_flag and four\_way\_handshake\_flag:  
 return True  
if three\_way\_handshake\_flag and not four\_way\_handshake\_flag:  
 return 0  
else:  
 return False

-Vráti True, pokiaľ je komunikácia kompletná,

-Vráti 0, pokiaľ je nekompletná,

-Vráti False

Pokiaľ sa z metódy na overenie handshake-ov vráti True, vyhladajú sa všetky rámce, ktoré boli pre túto komunikáciu použité, a vytvorí sa objekt Communication, ktorý reprezentuje úplnú komunikáciu a obsahuje všetky potrebné atribúty pre exportovanie do YAML. Ak sa vráti 0, vytvorí sa objekt Communication, ktorý reprezentuje neúplnú komunikáciu a obsahuje všetky potrebné atribúty pre exportovanie do YAML.

* Komunikácia UDP protokolov:

Vynechané

* Komunikácia ICMP protokolov:

Vynechané

* Komunikácia ARP protokolov:

Vynechané

* IP fragmentácia:

Vynechané

# Zvolené prostredie

Implementácia bola napísaná v Pythone v prostredí The Jetbrains IDE Pycharm.

Ruamel yaml bol použitý na export YAML.

# Záver a možné zlepšenie programu

* Doimplementovanie ostatných komunikácií
* Vytvorenie GUI, alebo zlepšenie konzolového menu
* Support pre viaceré filtre pre jeden súbor