

# PKS – Zadanie 1 – Dokumentácia

## POČÍTAČOVÉ A KOMUNIKAČNÉ SIETE

ak. rok 2020/21, zimný semester

### Zadanie 1: Analyzátor sieťovej komunikácie

#### Zadanie úlohy

Navrhните a implementujte programový analyzátor Ethernet siete, ktorý analyzuje komunikácie v

sieti zaznamenané v .pcap súbore a poskytuje nasledujúce informácie o komunikáciách.

Vypracované zadanie musí spĺňať nasledujúce body:

1) **Výpis všetkých rámcov v hexadecimálnom tvare** postupne tak, ako boli zaznamenané v súbore.

Pre každý rámec uveďte:

a) Poradové číslo rámca v analyzovanom súbore.

b) Dĺžku rámca v bajtoch poskytnutú pcap API, ako aj dĺžku tohto rámca prenášaného po médiu.

c) Typ rámca – Ethernet II, IEEE 802.3 (IEEE 802.3 s LLC, IEEE 802.3 s LLC a SNAP, IEEE 802.3 – Raw).

d) Zdrojovú a cieľovú fyzickú (MAC) adresu uzlov, medzi ktorými je rámec prenášaný. Vo výpise jednotlivé **bajty rámca usporiadajte po 16 alebo 32 v jednom riadku**. Pre prehľadnosť

výpisu je vhodné použiť neproporcionálny (monospace) font.

2) Pre rámce typu **Ethernet II a IEEE 802.3 vypíšte vnorený protokol**. Študent musí vedieť vysvetliť,

aké informácie sú uvedené v jednotlivých rámcoch Ethernet II, t.j. vnáranie protokolov ako aj ozrejmiť dĺžky týchto rámcov.

3) Analýzu cez vrstvy vykonajte pre rámce Ethernet II a protokoly rodiny TCP/IPv4:

**Na konci výpisu z bodu 1)** uveďte pre IPv4 pakety:

a) Zoznam IP adries všetkých prijímajúcich uzlov,

b) IP adresu uzla, ktorý sumárne prijal (bez ohľadu na odosielaťľa) najväčší počet paketov a koľko paketov prijal (berte do úvahy iba IPv4 pakety).

IP adresy a počet poslaných paketov sa musia zhodovať s IP adresami vo výpise Wireshark -> Statistics -> IPv4 Statistics -> Source and Destination Addresses.

4) V danom súbore analyzujte komunikácie pre zadané protokoly:

a) HTTP

b) HTTPS

c) TELNET

d) SSH

e) FTP riadiace

f) FTP dátové

g) TFTP, **uveďte všetky rámce komunikácie**, nielen prvý rámec na UDP port 69

h) ICMP, uveďte aj typ ICMP správy (pole Type v hlavičke ICMP), napr. Echo request, Echo reply, Time exceeded, a pod.

i) **Všetky** ARP dvojice (request – reply), uveďte aj IP adresu, ku ktorej sa hľadá MAC (fyzická)

adresa a pri ARP-Reply uveďte konkrétny pár - IP adresa a nájdená MAC adresa. V prípade, že bolo poslaných viacero rámcov ARP-Request na rovnakú IP adresu, vypíšte všetky. Ak sú v súbore rámce ARP-Request bez korešpondujúceho ARP-Reply (alebo naopak ARPReply bez ARP-Request), vypíšte ich samostatne.

**Vo všetkých výpisoch treba uviesť aj IP adresy a pri transportných protokoloch TCP a UDP aj porty komunikujúcich uzlov.**

V prípadoch komunikácií so spojením vypíšte iba jednu kompletnú komunikáciu - obsahuje otvorenie (SYN) a ukončenie (FIN na oboch stranách alebo ukončenie FIN a RST alebo ukončenie

iba s RST) spojenia a aj prvú nekompletnú komunikáciu, ktorá obsahuje iba otvorenie spojenia.

Pri výpisoch vyznačte, ktorá komunikácia je kompletná.

Ak počet rámcov komunikácie niektorého z protokolov z bodu 4 je väčší ako 20, vypíšte iba 10

prvých a 10 posledných rámcov tejto komunikácie. **(Pozor: toto sa nevzťahuje na bod 1, program**

**musí byť schopný vypísať všetky rámce zo súboru podľa bodu 1.)** Pri všetkých výpisoch musí byť

poradové číslo rámca zhodné s číslom rámca v analyzovanom súbore.

5) Program musí byť organizovaný tak, aby čísla protokolov v rámci Ethernet II (pole Ethertype),

IEEE 802.3 (polia DSAP a SSAP), v IP pakete (pole Protocol), ako aj čísla portov v transportných

protokoloch boli programom **načítané z jedného alebo viacerých externých textových súborov.**

Pre známe protokoly a porty (minimálne protokoly v bodoch 1) a 4) budú uvedené aj ich názvy.

Program bude schopný uviesť k rámcu názov vnoreného protokolu po doplnení názvu k číslu protokolu, resp. portu do externého súboru. Za externý súbor sa nepovažuje súbor knižnice, ktorá

je vložená do programu.

6) V procese analýzy rámcov pri identifikovaní jednotlivých polí rámca ako aj polí hlavičiek vnorených protokolov nie je povolené použiť funkcie poskytované použitým programovacím jazykom alebo knižnicou. **Celý rámec je potrebné spracovať postupne po bajtoch.**

7) Program musí byť organizovaný tak, aby bolo možné jednoducho rozširovať jeho funkčnosť

výpisu rámcov pri doimplementovaní jednoduchej funkčnosti na cvičení.

8) Študent musí byť schopný preložiť a spustiť program v miestnosti, v ktorej má cvičenia. V prípade

dištančnej výučby musí byť študent schopný prezentovať podľa pokynov cvičiaceho program online, napr. cez Webex, Meet, etc.

V danom týždni, podľa harmonogramu cvičení, musí študent priamo na cvičení doimplementovať

do funkčného programu (podľa vyššie uvedených požiadaviek) ďalšiu prídavnú funkčnosť.

**Program musí mať nasledovné vlastnosti (minimálne):**

1) Program musí byť implementovaný v jazykoch C/C++ alebo Python s využitím knižnice pcap, skompilovateľný a spustiteľný v učebniach. Na otvorenie pcap súborov použite

knižnice *libpcap* pre linux/BSD a *winpcap/npcap* pre Windows. Použité knižnice a funkcie musia byť schválené cvičiacim. V programe môžu byť použité údaje o dĺžke rámca zo struct *pcap\_pkthdr* a funkcie na prácu s pcap súborom a načítanie rámcov:

*pcap\_createsrcstr()*  
*pcap\_open()*  
*pcap\_open\_offline()*  
*pcap\_close()*  
*pcap\_next\_ex()*  
*pcap\_loop()*

Použitie funkcionality *libpcap* na priamy výpis konkrétnych polí rámca (napr. *ih->saddr*) bude mať za následok nulové hodnotenie celého zadania.

2) Program musí pracovať s dátami optimálne (napr. neukladať MAC adresy do 6x int).

3) Poradové číslo rámca vo výpise programu musí byť zhodné s číslom rámca v analyzovanom súbore.

4) Pri finálnom odovzdaní, pre každý rámec vo všetkých výpisoch uviesť použitý protokol na 2. -

4. vrstve OSI modelu. (ak existuje)

5) Pri finálnom odovzdaní, pre každý rámec vo všetkých výpisoch uviesť zdrojovú a cieľovú adresu / port na 2. - 4. vrstve OSI modelu. (ak existuje)

Nesplnenie ktoréhokoľvek bodu minimálnych požiadaviek znamená neakceptovanie riešenia cvičiacim.

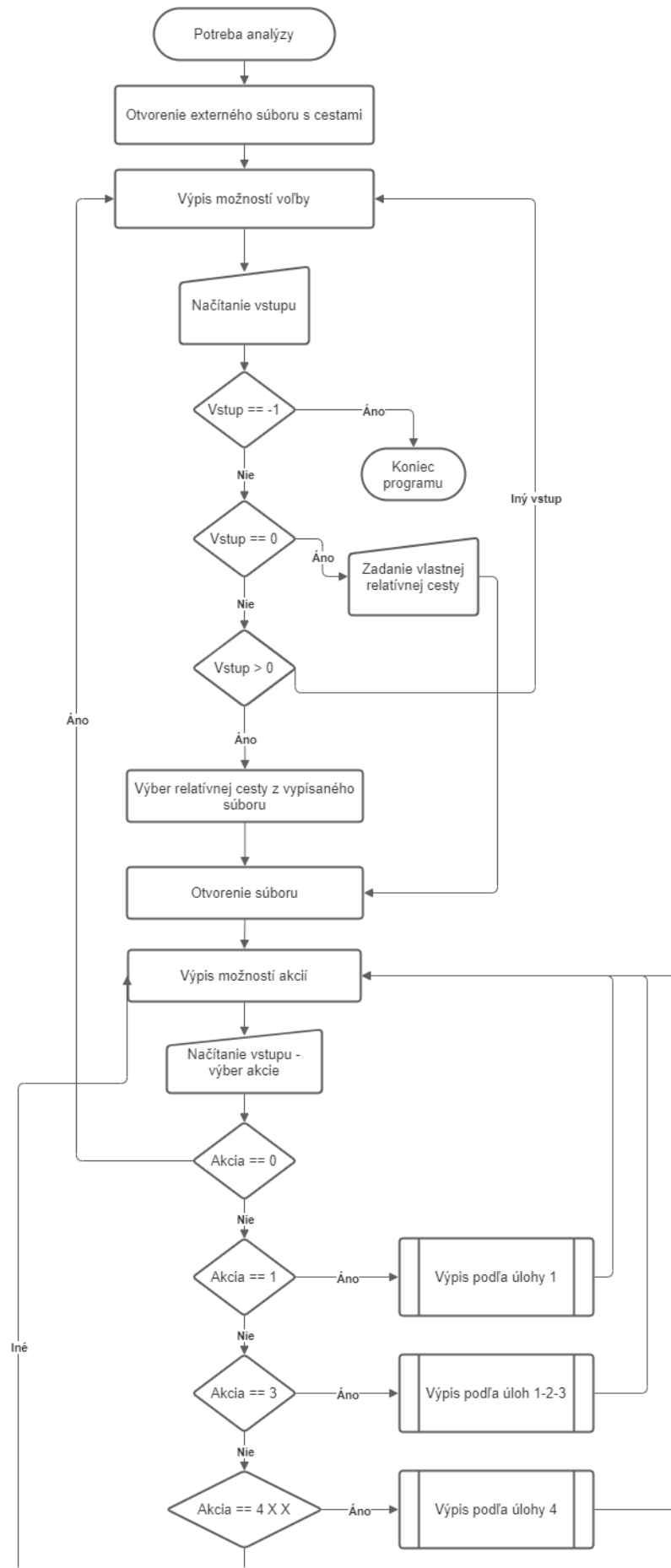
**Súčasťou riešenia je aj dokumentácia, ktorá musí obsahovať najmä:**

- a) zadanie úlohy,
- b) blokový návrh (konceptia) fungovania riešenia,
- c) navrhnutý mechanizmus analyzovania protokolov na jednotlivých vrstvách,
- d) príklad štruktúry externých súborov pre určenie protokolov a portov,
- e) opísané používateľské rozhranie,
- f) voľbu implementačného prostredia.

## Blokový návrh fungovania riešenia

### 1. Blokový diagram riadiaceho programu

Tento diagram zobrazuje riadiacu funkciu programu a postupnosť krokov pre výber súboru, jeho analýzu a následné ukončenie programu. Táto časť úzko súvisí s používateľským rozhraním, preto bude podrobne opísaná v časti na to určenej.

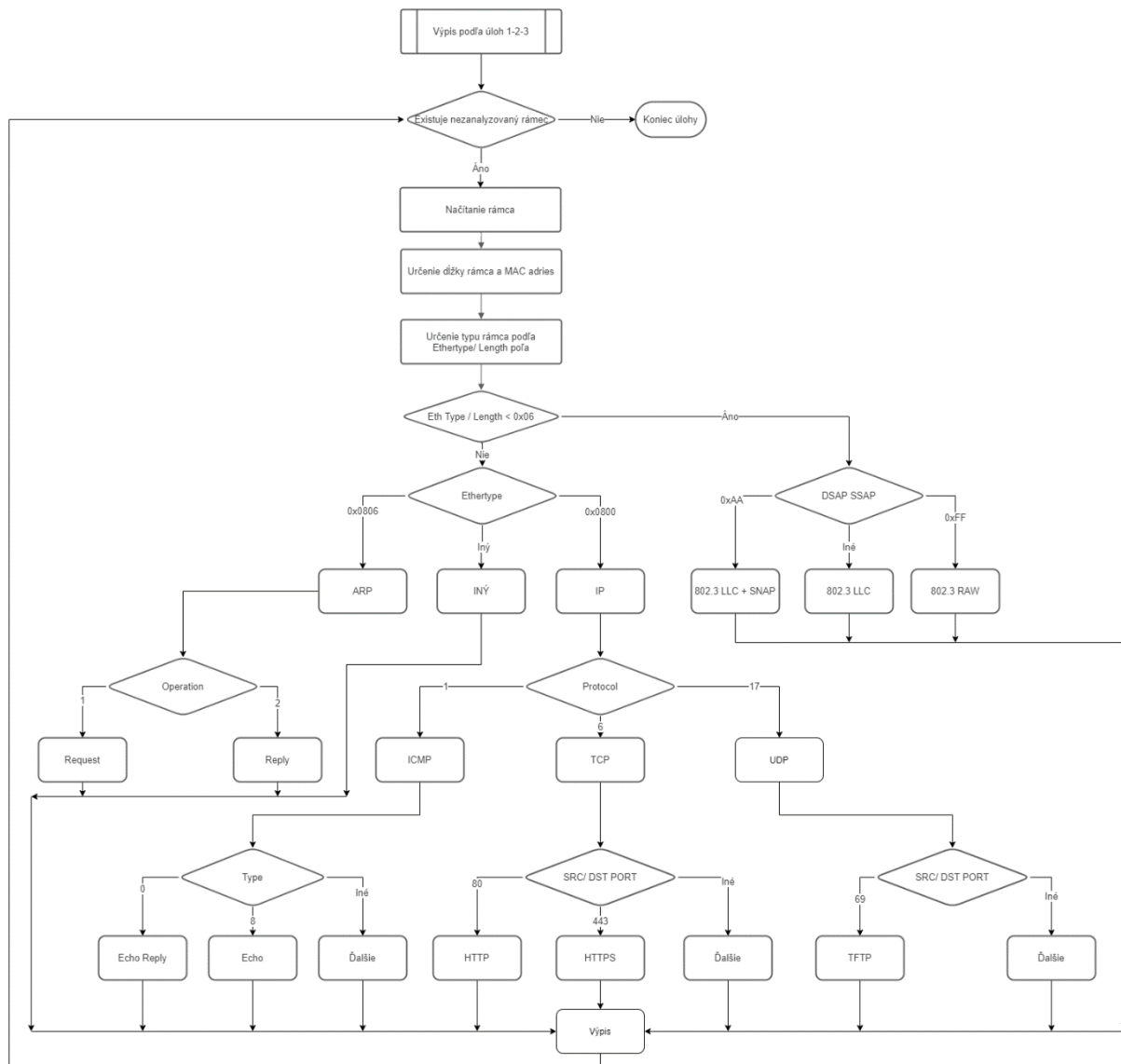


2. Blokový diagram pre funkciu 1-2-3 (pre samotnú funkciu 1 ho neuvádzam, keďže ten je podmnožinou tohto diagramu)

Tento diagram zobrazuje funkciu zadanú podľa bodov 1-2-3. Funkcia analyzuje všetky rámce pcap súboru. Po zistení MAC adresy a dĺžky rámca sa začína analýza. Najskôr sa zistí, či je rámec typu Ethernet II alebo 802.3 podľa porovnania hodnoty na mieste ethertype/ length. Ak je rámec 802.3, na základe DSAP a SSAP sa zistí, že je rámec RAW, LLC alebo LLC + SNAP.

Ak je rámec Ethernet II, zanalyzuje sa ethertype. Ak je ethertype ARP, zistí sa operácia. Ak je ethertype IPv4, zistí sa protokol. Ak je protokol ICMP, zistí sa typ ICMP. Ak je protokol TCP, zistí sa port a korešpondujúci názov komunikácie. Ak je protokol UDP, zistí sa port a korešpondujúci názov komunikácie. Ak je ethertype iný, zistí sa jeho typ.

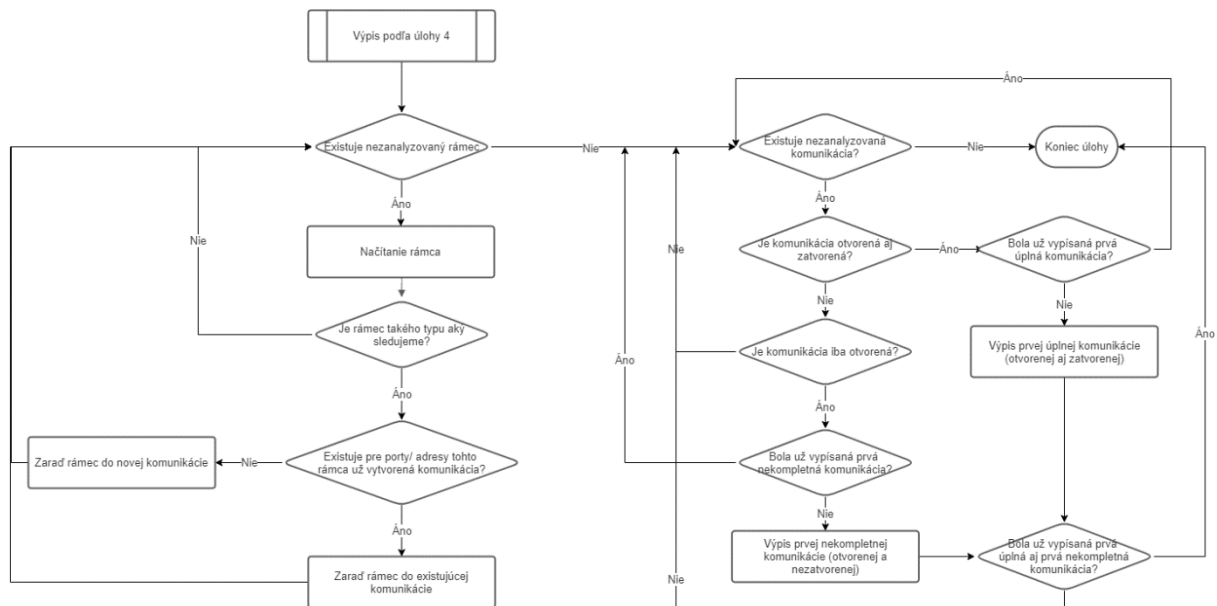
Všetky tieto informácie sa vypíšu a zistí sa, či existuje neanalyzovaný rámec. Ak existuje, pokračuje sa jeho analýzou. Ak neexistuje, funkcia končí a riadiaci program opäť žiada zadanie akcie.



### 3. Blokový diagram funkcie 4

Tento diagram zobrazuje zarad'ovanie rámcov do komunikácií a následný výpis prvej úplnej (otvorenej a zatvorenej) a prvej nekompletnej (iba otvorenej, tj. otvorenej a nezatvorenej) komunikácie. Najskôr sa zistí, či sú všetky rámce zanalyzované. Ak nie sú, zanalyzujú sa a podľa požadovaného typu sa zaradia do komunikácií. Ak všetky rámce sú zaradené do komunikácií, začne sa analýza komunikácií. Postupne sa analyzujú komunikácie. Ak sa nájde prvá otvorená a zatvorená komunikácia, vypíše sa ako kompletná. Ak sa nájde prvá otvorená a nezatvorená komunikácia, vypíše sa ako nekompletná. Ak sú obe komunikácie vypísané, alebo sú všetky komunikácie zanalyzované, funkcia končí.

Výpis komunikácie v tejto funkcii prebieha podobne ako je opísané v diagrame pre úlohy 1-2-3, preto diagram samostatného výpisu pre túto funkciu neprikladám.



### Mechанизmus analýzy protokolov na jednotlivých vrstvách

Po otvorení a načítaní súboru a zvolení funkcie pre analýzu sa zistí MAC adresa cieľa z prvých 6 bajtov rámca a MAC adresa zdroja z nasledujúcich 6 bajtov. Určí sa veľkosť rámca poskytnutá API a veľkosť rámca prenášaná na médiu. Veľkosť rámca na médiu rátam ako maximum(64, (veľkosť rámca z API + 4)).

Následne sa rozhodne, či je na 13-14 bajte ethertype, alebo length tak, že sa hodnota tohto čísla porovná s 0x0600. Ak je hodnota väčšia ako 0x0600, ide o ethertype a teda je rámec Ethernet II. Ak je ale hodnota menšia, rámec je 802.3. Ak je na 15-16 bajtoch FF FF, ide o 802.3 RAW, ak je tam AA AA ide o 802.3 LLC + SNAP, ak je tam iná hodnota, podľa externého súboru sa zistí, aký názov jej prislúcha. Typ rámca sa vypíše.

Počas analýzy Ethernet II sa zistí vnorený protokol. Ak ide o ARP (ethertype je 0x0806), zanalyzuje sa a vypíše sa operácia ARP správy, MAC adresa zdroja, IP adresa zdroja a IP adresa cieľa. MAC adresa cieľa sú samé nuly, pretože je cieľom protokolu ARP túto MAC adresu zistiť.

Ak je v Ethernet II vnorený IPv4 protokol (ethertype je 0x0800), zistí sa veľkosť IP hlavičky. Ak je na pozícii dĺžky IP hlavičky hodnota 0x45, hlavička má 20b. Cifra na pozícii jednotiek určuje počet 32 bitových slov ktoré tvoria hlavičku.

Vypíše sa zdrojová a cieľová IP a vnorený protokol. Ak je vnorený protokol ICMP (protokol je 1), zistí sa typ tohto protokolu. Ak je to 0, je to echo reply, ak 8, je to echo atď podľa ťaháku.

Ak je v IPv4 vnorený protokol TCP (protokol je 6), zistia sa jeho porty. Ak je jeden z portov 80, ide o HTTP, ak je to 443, ide o HTTPS atď podľa ťaháku.

Ak je v IPv4 vnorený protokol UDP (protokol je 17), zistia sa jeho porty. Ak je jeden z portov 69, ide o TFTP atď podľa ťaháku.

Ak je v IPv4 vnorený iný protokol ako ICMP/TCP/UDP vypíše sa podľa externého súboru.

Ak v Ethernet II nie je vnorený ani IPv4 ani ARP, na základe ethertypeu sa zistí typ vnoreného protokolu, nájde sa v externom súbore a vypíše sa.

Pri triedení rámcov do komunikácií sa pri TCP/UDP používa spájanie podľa rovnakých portov. Pri UDP, keďže ide o komunikáciu bez spojenia, sa vypíše komunikácia na konkrétnom porte. Pri TCP, keďže ide o komunikáciu so spojením sa vypíše prvá úplná komunikácia, teda taká, čo obsahuje otvorenie (SYN, SYNACK, ACK) a zatvorenie (FIN na oboch stranách, FIN a RST alebo RST. Taktiež sa vypíše prvá nekompletná komunikácia, ktorá je len otvorená (SYN, SYNACK, ACK) a nie je nijakým spôsobom ukončená.

Pri ICMP sa spájajú echo s echo reply podľa IP adres uzlov.

Pri ARP sa spája jeden a viac ARP request s jedným ARP reply, pričom sa sledujú IP a MAC adresy. Tieto dvojice, prípadne samostatné ARP requesty alebo samostatné ARP reply sa vypíšu.

## Príklad štruktúry externých súborov

Odovzdávam kód programu v súbore `Smrecek-Zadanie1-Cele.py`, ktorý pre svoju funkčnosť potrebuje textový súbor `zoznamSuborov.txt` obsahujúci relatívne adresy k pcap súborom a textový súbor `protokoly.txt`. Pcap súbory sú uložené v adresári `vzorky_pcap_na_analyzu`, ktorý sa nachádza v adresári spolu so `Smrecek-Zadanie1-Cele.py` a `zoznamSuborov.txt`.

Vstupom je súbor `.pcap` ktorý je možno vybrať zo zoznamu pcap súborov, alebo k nemu napísať relatívnu cestu. Relatívna cesta však začína vždy v adresári s `Smrecek-Zadanie1-Cele.py` súborom. Výstup programu sa vypisuje do súboru `vystup.txt`, ktorý sa v prípade jeho neexistencie vytvorí, v prípade existencie prepíše.

Príklad obsahu súboru `zoznamSuborov.txt`:

Uvádzam len niekoľko prvých relatívnych ciest z tohto súboru.

vzorky\_pcap\_na\_analyzu/eth-1.pcap  
vzorky\_pcap\_na\_analyzu/eth-2.pcap  
vzorky\_pcap\_na\_analyzu/eth-3.pcap  
vzorky\_pcap\_na\_analyzu/eth-4.pcap  
vzorky\_pcap\_na\_analyzu/eth-5.pcap

...

Príklad obsahu súboru `protokoly.txt`:

Pre prehľadnosť udávam len prvé 3 riadky ku každému nadpisu. Pri pridávaní protokolu do súboru je nutné ho pridať na nový riadok k správne nadpisu. Záznam je číslo v hexadecimálnom tvare začínajúce 0x a je od názvu protokolu oddelené jednou medzerou. Pri pridávaní nového nadpisu treba nadpis oddeliť znakom # a podoň uviesť kódy protokolov/portov. Medzi záznamami nesmú vznikáť voľné riadky.

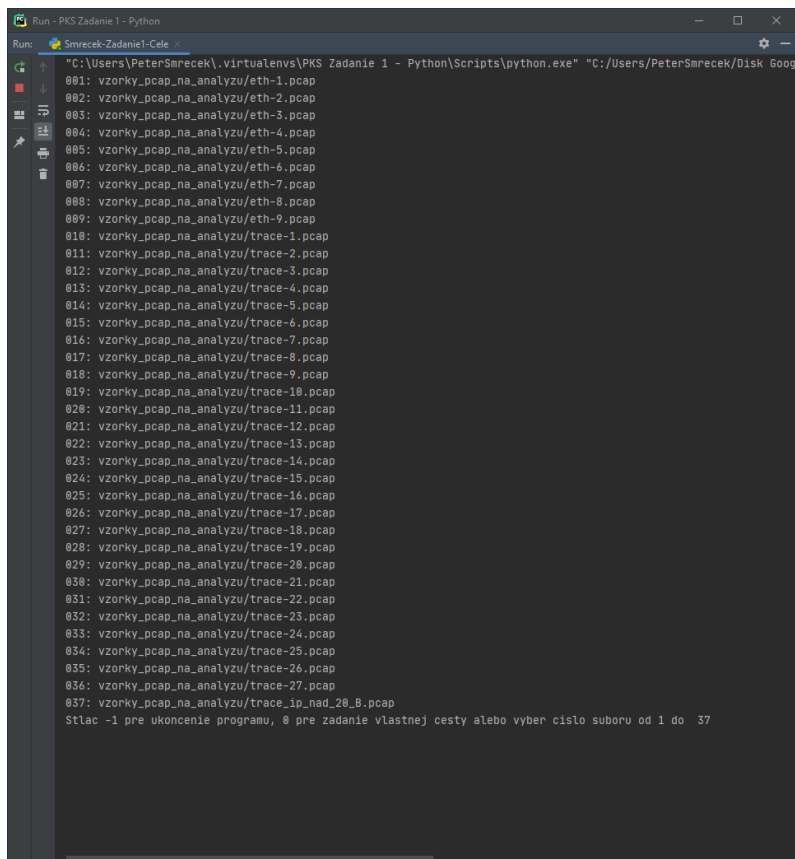
```
#Ethertypes
0x0800 IPv4
0x0801 X.75 Internet
0x0805 X.25 Level 3
...
#SAPs
0x00 NULL SAP
0x02 LLC Sublayer Management or Individual
0x03 LLC Sublayer Management or Group
...
#IP
0x01 ICMP
0x02 IGMP
0x06 TCP
...
#TCP
0x07 ECHO
0x13 CHARGEN
0x14 FTP DATA
...
#UDP
0x07 Echo
0x13 Chargen
0x25 Time
...
#ICMP
0x00 Echo Reply
0x03 Destination Unreachable
0x04 Source Quench
#ARP
0x01 Request
0x02 Reply
```



## Opis používateľského rozhrania

Riešenie zadania som implementoval ako konzolovú aplikáciu. K používateľskému rozhraniu patrí diagram 1 z bodu Blokový návrh fungovania riešenia.

Po spustení programu sa vypíše na obrazovku súbor `zoznamSuborov.txt` s očíslovanými riadkami. Volajme to Menu 1. Je možné zvoliť jedno z týchto čísel a následne



```
Run - PKS Zadanie 1 - Python
Run: Smrecek-Zadanie1-Cele
"C:\Users\PeterSmrecek\virtualenvs\PKS Zadanie 1 - Python\Scripts\python.exe" "C:/Users/PeterSmrecek/Disk Goog
001: vzorky_pcap_na_analyzu/eth-1.pcap
002: vzorky_pcap_na_analyzu/eth-2.pcap
003: vzorky_pcap_na_analyzu/eth-3.pcap
004: vzorky_pcap_na_analyzu/eth-4.pcap
005: vzorky_pcap_na_analyzu/eth-5.pcap
006: vzorky_pcap_na_analyzu/eth-6.pcap
007: vzorky_pcap_na_analyzu/eth-7.pcap
008: vzorky_pcap_na_analyzu/eth-8.pcap
009: vzorky_pcap_na_analyzu/eth-9.pcap
010: vzorky_pcap_na_analyzu/trace-1.pcap
011: vzorky_pcap_na_analyzu/trace-2.pcap
012: vzorky_pcap_na_analyzu/trace-3.pcap
013: vzorky_pcap_na_analyzu/trace-4.pcap
014: vzorky_pcap_na_analyzu/trace-5.pcap
015: vzorky_pcap_na_analyzu/trace-6.pcap
016: vzorky_pcap_na_analyzu/trace-7.pcap
017: vzorky_pcap_na_analyzu/trace-8.pcap
018: vzorky_pcap_na_analyzu/trace-9.pcap
019: vzorky_pcap_na_analyzu/trace-10.pcap
020: vzorky_pcap_na_analyzu/trace-11.pcap
021: vzorky_pcap_na_analyzu/trace-12.pcap
022: vzorky_pcap_na_analyzu/trace-13.pcap
023: vzorky_pcap_na_analyzu/trace-14.pcap
024: vzorky_pcap_na_analyzu/trace-15.pcap
025: vzorky_pcap_na_analyzu/trace-16.pcap
026: vzorky_pcap_na_analyzu/trace-17.pcap
027: vzorky_pcap_na_analyzu/trace-18.pcap
028: vzorky_pcap_na_analyzu/trace-19.pcap
029: vzorky_pcap_na_analyzu/trace-20.pcap
030: vzorky_pcap_na_analyzu/trace-21.pcap
031: vzorky_pcap_na_analyzu/trace-22.pcap
032: vzorky_pcap_na_analyzu/trace-23.pcap
033: vzorky_pcap_na_analyzu/trace-24.pcap
034: vzorky_pcap_na_analyzu/trace-25.pcap
035: vzorky_pcap_na_analyzu/trace-26.pcap
036: vzorky_pcap_na_analyzu/trace-27.pcap
037: vzorky_pcap_na_analyzu/trace_ip_nad_28.pcap
Stlac -1 pre ukoncenie programu, 0 pre zadanie vlastnej cesty alebo vyber cislo suboru od 1 do 37
```

Menu 1

program otvorí pcap súbor s cestou, ktorá bola vybraná. Ak používateľ zvolí 0, môže zadať vlastnú relatívnu cestu, ktorá však začína v priečinku so zdrojovým kódom. Príklad takejto cesty môže byť napríklad `ine_vstupy/vstup1.pcap`, čo je vlastne cesta k pcap súboru, ktorý sa nachádza v priečinku `ine_vstupy`, ktorý sa nachádza v priečinku so zdrojovým kódom. Následne program otvorí pcap súbor s cestou, ktorá bola zadaná.

Ak používateľ chce program ukončiť, zvolí -1. Program sa vtedy korektne ukončí.

Ak používateľ miesto voľby čísla s relatívnou cestou, alebo možnosti zadať vlastnú cestu, alebo možnosti ukončiť

program zadal iný vstup, vstup nebude akceptovaný.

Príklady vstupu pre Menu 1:

Voľba pre Menu 1	Popis
-1	Ukončí program
0 ine_vstupy/vstup1.pcap	Otvorí možnosť zadať vlastnú rel. cestu Vlastná relatívna cesta (následný vstup do Menu 2)
1	Otvorí súbor v zozname označený č. 1 (následný vstup do Menu 2)
2	Otvorí súbor v zozname označený č. 2 (následný vstup do Menu 2)

Ak používateľ vybral jedno z čísel, alebo zadal vlastnú cestu, otvorí sa na pozadí pcap súbor a zobrazí sa ďalšie Menu. Môžeme ho volať Menu 2. Tu si používateľ vyberie možnosť 0 pre návrat do Menu 1, možnosť 1 pre výpis do externého súboru podľa bodu 1 zadania,

možnosť 3 pre výpis do externého súboru podľa bodov 1-2-3. Používateľ môže zvoliť možnosť 4 a 0 (3 znaky oddelené medzerami) pre výpis všetkých HTTP rámcov daného pcap súboru, alebo 4 a 1 pre výpis prvej úplnej a prvej nekompletnej komunikácie z daného pcap súboru. Tento výpis zodpovedá zadaniu úlohy 4 a). Analogicky takto môže vypísať všetky výpisy úlohy 4.

Príklady vstupu pre Menu 2:

Voľba pre Menu 2	Popis
0	Návrat do Menu 1 (pre potreby výberu iného pcap súboru alebo ukončenia programu)
1	Výpis podľa bodu 1
3	Výpis podľa bodov 1-2-3
4 a 0	Vyfiltruje všetky HTTP komunikácie zo súboru
4 a 1	Vypíše prvú úplnú a prvú neúplnú HTTP komunikáciu podľa bodu 4a)
4 b 0	Vyfiltruje všetky HTTPS komunikácie zo súboru
4 b 1	Vypíše prvú úplnú a prvú neúplnú HTTPS komunikáciu podľa bodu 4b)
...	Analogicky pre úlohy 4c) - 4f)
4 g 0	Vypíše prvú TFTP komunikáciu zo súboru podľa bodu 4g)
<del>4 g 1</del>	Táto možnosť nie je implementovaná, lebo UDP nenadväzuje spojenie
4 h 0	Vyfiltruje všetky ICMP komunikácie zo súboru
4 h 1	Vypíše všetky ICMP echo – reply dvojice zo súboru podľa bodu 4h)
4 i 0	Vyfiltruje všetky ARP komunikácie zo súboru
4 i 1	Vypíše všetky ARP páry a samostatné nespárené requesty a reply podľa bodu 4i)

Pre možnosti 0, 1, 3 je na vstupe jedno číslo, pre možnosti úlohy 4 sú na vstupe 3 znaky oddelené dvoma medzerami vo formáte [4]\_[a-i]\_[0-1] kde \_ symbolizuje medzeru. Iné vstupy nebudú akceptované.

Možnosti [4]\_[a-f | h-i]\_[0] nie sú zadáním priamo požadované, fungujú len ako filter komunikácií. [4]\_[g]\_[0] korešponduje so zadáním úlohy 4g).

Možnosti [4]\_[a-f | h-i]\_[1] korešpondujú s bodmi úlohy 4. Možnosť [4]\_[g]\_[1] nie je implementovaná.

Po zvolení nejakej možnosti sa do súboru `vystup.txt` vypíše výstup podľa zvolenej úlohy. Následne sa opäť vypíše Menu 1. Nad tým istý pcap súborom vybraným z Menu 1 je možné urobiť ľubovoľný počet operácií z Menu 2 až do stlačenia možnosti 0 pre návrat do Menu 1 a výber iného pcap súboru alebo ukončenie programu. Každá voľba z Menu 2 prepíše súbor `vystup.txt`.

```
028: vzorky_pcap_na_analyzu/trace-19.pcap
029: vzorky_pcap_na_analyzu/trace-20.pcap
030: vzorky_pcap_na_analyzu/trace-21.pcap
031: vzorky_pcap_na_analyzu/trace-22.pcap
032: vzorky_pcap_na_analyzu/trace-23.pcap
033: vzorky_pcap_na_analyzu/trace-24.pcap
034: vzorky_pcap_na_analyzu/trace-25.pcap
035: vzorky_pcap_na_analyzu/trace-26.pcap
036: vzorky_pcap_na_analyzu/trace-27.pcap
037: vzorky_pcap_na_analyzu/trace_ip_nad_20_B.pcap
Stlač -i pre ukoncenie programu, 0 pre zadanie vlastnej cesty alebo vyber cislo suboru od 1 do 37

Bola zvolena cesta C:/Users/PeterSmrecek/Disk Google (xsmrecek@stuba.sk)/Programovanie/20-21/PKS/Zadanie1/PKS Zadanie 1 - Python\vzorky_pcap_na_analyzu/eth-5.pcap
Zvol 0 pre vyber ineho pcap suboru
Zvol 1 pre vypis podla bodu 1
Zvol 3 pre vypis podla bodov 1-2-3 spolu
Zvol 4 pre ulohu 4, zvol bod ulohy 4 a-1, zvol 0 ci vypisat iba ramce alebo 1 pre vypis komunikacii podla zadania
Zadaj cisla [0,1,3] alebo vstup vo formate [4] [a-1] [0-1]
```

## Menu 2

Príklady vstupu pre celý program od spustenia do konca:

Voľba	Popis
5	Otvorí eth-5.pcap
3	Vypíše do súboru výstup podľa bodov 1-2-3
4 d 1	Premaže súbor a vypíše doňho výstup podľa bodu 4d)
0	Návrat do Menu 1
-1	Ukončenie programu

## Implementačné prostredie a kód

Program som písal v programovacom jazyku Python v prostredí PyCharm. Verzia Pythonu je 3.8.3.

Pre potreby otvorenia pcap súboru používam Scapy knižnicu. Ďalšie packages, ktoré používam v kóde sú `os`, `sys`, `datetime` a `Counter` z `collections`.

Program je spustiteľný ak sú dostupné všetky potrebné externé súbory a nainštalované všetky knižnice.

V zadaní sa nachádzajú zakomentované debuggované funkcie a funkcie merania času. Po ich odkomentovaní je možné merať čas čítania, prepisu aj analýzy pcap súboru.

Kód obsahuje komentáre na kritických miestach. Každá funkcia má svoj doc komentár so stručným popisom jej činnosti. Komentáre ku kódu aj s ukážkami v tejto dokumentácii neuvádzam, keďže kód je rozsiahly, priamo v kóde okomentovaný a odovzdaný spolu s touto dokumentáciou.