Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií
Ilkovičova 2
842 16 Bratislava 4
Akademický rok 2019/2020
Zadanie 2: Analyzátor sieťovej komunikácie
Vypracoval: Nicolas Mikulík
AIS ID: 96973

Zadanie úlohy

Navrhnite 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 vysielajúcich uzlov,
b) IP adresu uzla, ktorý sumárne odoslal (bez ohľadu na príjemcu) najväčší počet paketov a koľko paketov odoslal (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 riadiacef) 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 ARP-Reply 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 danom týždni, podľa harmonogramu cvičení, musí študent priamo na cvičení doimplementovať
do funkčného programu (podľa vyššie uvedených požiadaviek) ďalšiu prídavnú funkčnosť.

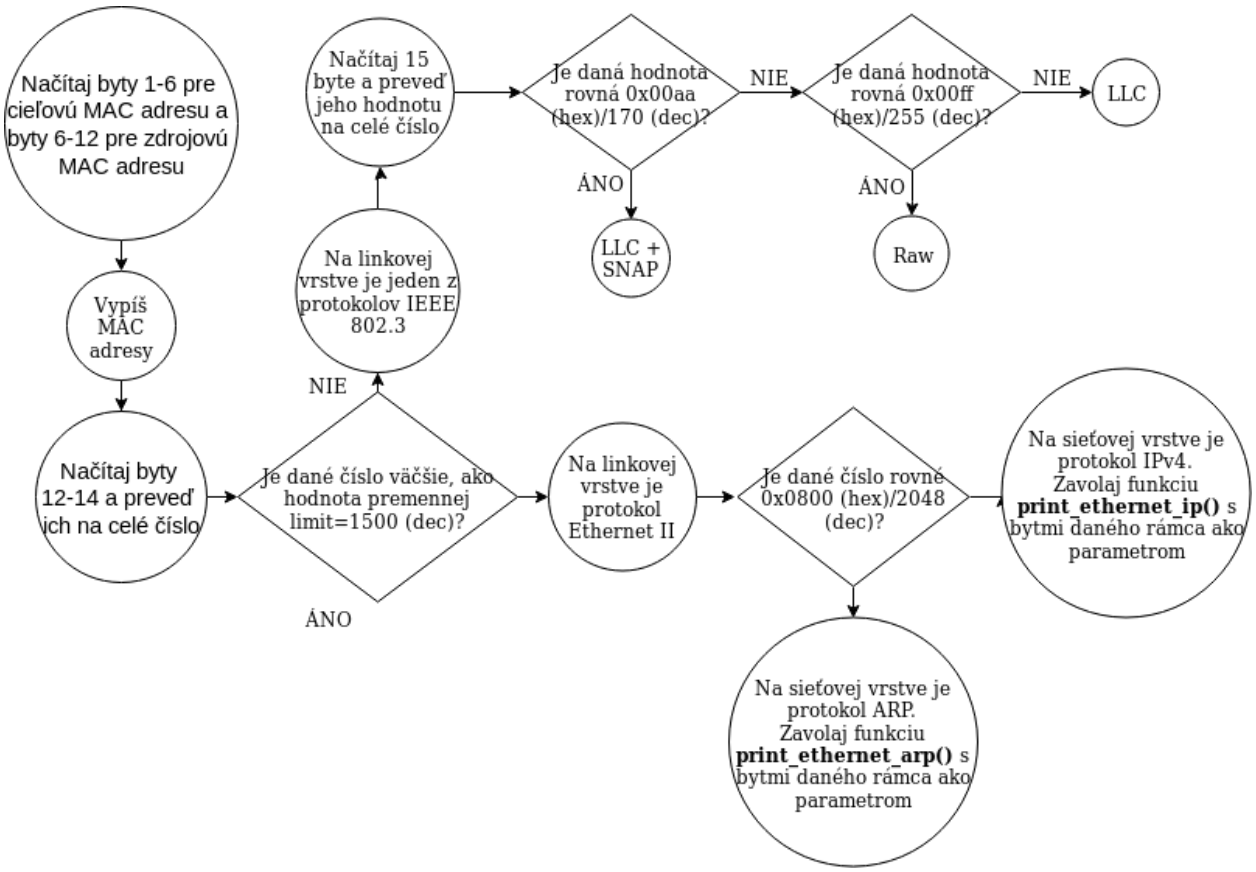
<u>Program musí mať nasledovné vlastnosti (minimálne):</u>
1) Program musí byť implementovaný v jazykoch C/C++ alebo Python s využitím knižnice pcap, skompilovateľný a spustiteľný v učebniach. Na otvorenie pcap súborov použite knižnice libpcap pre linux/BSD a winpcap pre Windows. Použité knižnice a funkcie 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 (koncepcia) 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.

Mechanizmus analyzovania:

Prvých štrnásť byteov rámca je analyzovaných vždy: byty 1-6 pre cieľovú MAC adresu, byty 6-12 pre zdrojovú MAC adresu, byty 12-14 pre určenie, či je protokol na linkovej vrstve Ethernet II alebo jeden z protokolov IEEE 802.3.



Štruktúra externého súboru:

Analyzátor využíva jeden externý konfiguračný súbor netcom_constants.txt, ktorého obsah je nasledovný (prvé číslo v riadku nie je súčasťou súboru, slúži len v tomto príklade na znázornenie poradového čísla riadku):

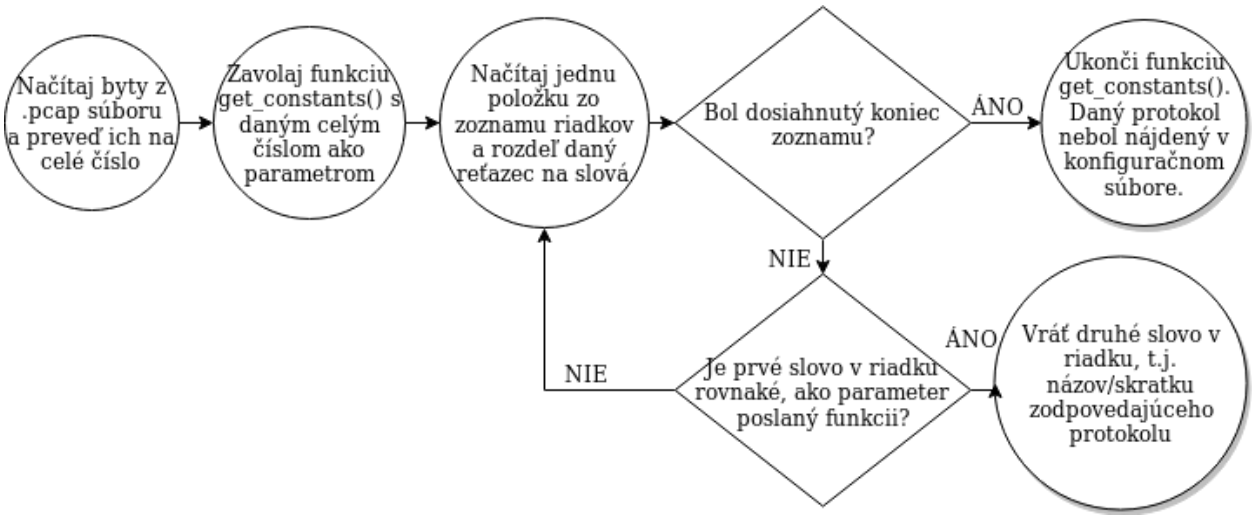
```
1  '''ETHERNET_IEEE'''
2  1500 Threshold
3  '''TYPE'''
4  2048 IPV4 0x0800
5  34525 IPV6 0x86dd
6  2054 ARP 0x0806
7  '''LSAP'''
8  170 SNAP 0x00aa
9  255 RAW 0x00ff
10 '''IP_PROTOCOL_NUMBER'''
11 1 ICMP 0x0001
12 6 TCP 0x0006
13 17 UDP 0x0011
14 88 EIGRP 0x0058
```

```
14 88 EIGRP 0x0058
15 '''TCP_PORT'''
16 139 NETBIOS_SES 0x008b
17 20 FTP_DATA 0x0014
18 21 FTP_CONTROL 0x0015
19 22 SSH 0x0016
20 23 TELNET 0x0017
21 80 HTTP 0x0050
22 443 HTTPS 0x01bb
23 '''UDP_PORT'''
24 69 TFTP 0x0045
25 53 DNS 0x0035
26 137 NETBIOS_NAM 0x0089
27
```

Riadok opisujúci protokol obsahuje dekadické číslo označujúce protokol, názov/skratku protokolu a hexadecimálne číslo označujúce protokol.

Aby sa znížil počet prístupov do súboru, pri spustení analyzátora je celý tento konfiguračný súbor načítaný do zoznamu, v ktorom je jedna položka zoznamu jedným riadkom súboru. Do premennej limit je z naplneného zoznamu načítaná hodnota 1500, s ktorou je porovnávaná hodnota 3 poľa rámca načítaného z .pcap súboru, pre určenie, či protokol na linkovej vrstve je Ethernet II alebo IEEE 802.3.

Pri analýze rámcov je každý port vyhľadaný v zozname riadkov z konfiguračného súboru, čo je opísané diagramom:



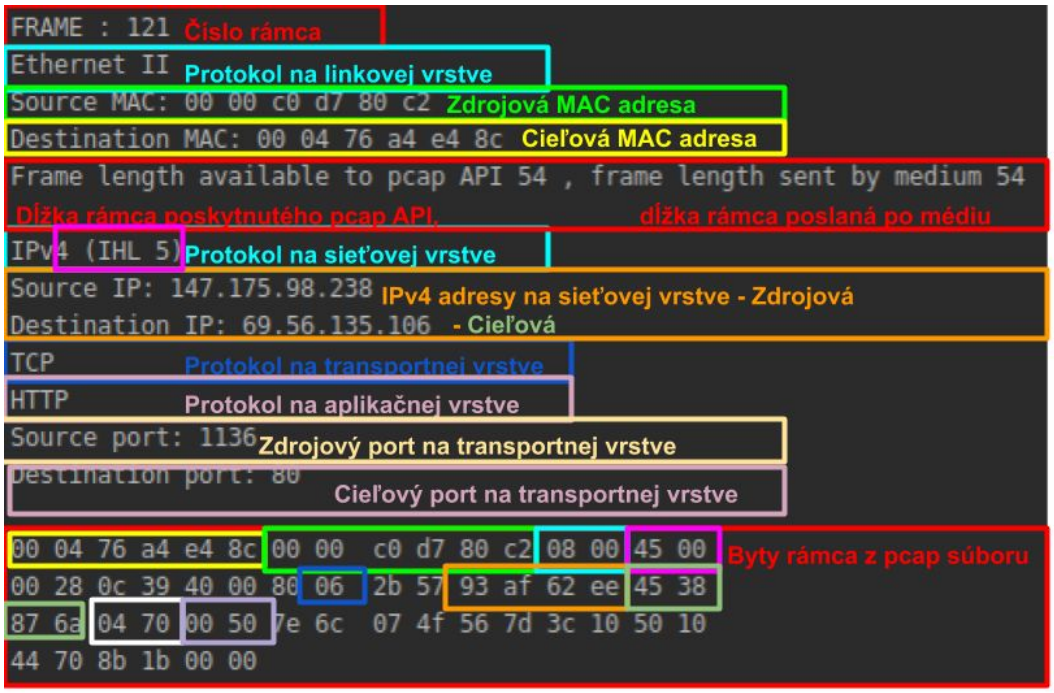
Používateľské rozhranie:

Používateľské rozhranie je konzolové. Pri spustení programu je užívateľ vyzvaný, aby zadal cestu k .pcap súboru. Pre zatvorenie aplikácie stačí napísať príkaz “exit”.
Po zadaní cesty k .pcap súboru je na štandardný výstup vypísaná analýza rámcov daného súboru, pričom riadky sú v poradí:

- číslo rámca
- typ protokolu na linkovej vrstve: Ethernet II / IEEE 802.3 SNAP / IEEE 802.3 Raw / IEEE 802.3 LLC
- zdrojová MAC adresa
- cieľová MAC adresa
- ak je typ protokolu na linkovej vrstve IEEE, nasleduje výpis byteov .pcap súboru,
- ak je typ protokolu na linkovej vrstve Ethernet II, nasleduje:
    - dĺžka súboru poskytnutá pcap API, dĺžka súboru prenesená po médiu
    - v prípade, že protokol na sieťovej vrstve je IPv4:
        - zdrojová IP adresa
        - cieľová IP adresa
        - typ protokolu na transportnej vrstve: TCP, UDP, ICMP
        - zdrojový port na transportnej vrstve
        - cieľový port na transportnej vrstve
    - výpis byteov .pcap súboru

Protokoly na všetkých vrstvách sú vo výpise uvádzané názvom, resp. skratkou, napr. Ethernet II, TCP. Hexadecimálne a decimálne označenia protokolov sú v konfiguračnom súbore netcom_constants.txt.

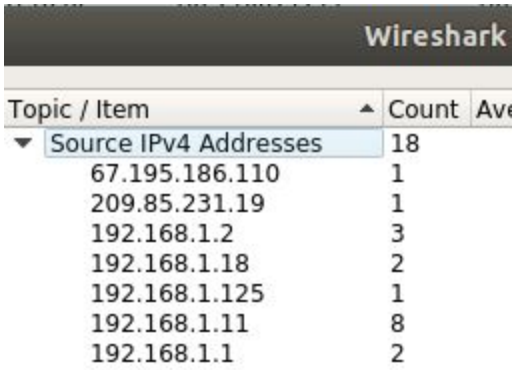
Ukážka výpisu jedného rámca:



Po výpise všetkých rámcov nasleduje výpis všetkých odosielajúcich IP adries, každá IP adresa je na samostatnom riadku. Potom je vypísaná IP adresa, ktorá odoslala najviac paketov, spolu s počtom ňou odoslaných paketov.
Porovnanie výpisu IPv4 adries vlastného analyzátora a Wiresharku pre súbor eth-8.pcap:

```
IP addresses of sending nodes:
192.168.1.2
192.168.1.11
192.168.1.125
192.168.1.18
209.85.231.19
192.168.1.1
67.195.186.110

Highest number of packets (14) was sent by 192.168.1.11
```



Po IPv4 adresách nasleduje výpis rámcov komunikácií pre protokoly v tomto poradí:
HTTP, HTTPS, TELNET, SSH, FTP riadiace, FTP dátové, TFTP, ICMP, ARP.
Ak neboli zachytené žiadne rámce niektorého z protokolov, analyzátor vypíše na štandardný výstup oznámenie, v tvare “No “+protokol+”communication recorded.”, ako príklad slúži výpis pre súbor eth-8.pcap:

```
No HTTPS communication recorded.
No TELNET communication recorded.
No SSH communication recorded.
No FTP-DATA communication recorded.
No FTP-CONTROL communication recorded.
```

Pri prvých piatich zo spomenutých protokol sú vo výpise rámcov komunikácií vypísané aj príznaky (flags) nastavené v TCP hlavičke v danom rámci.
Výpis prvých dvoch rámcov HTTP komunikácie zo súboru trace-1.pcap spolu s nastavenými príznakmi SYN a SYN ACK:

```
HTTP communication nr.  1
Frame : 7
Ethernet II
Source MAC: 00 d0 59 a9 3d 68
Destination MAC: 00 06 25 da af 73
IPv4 (IHL 5)
Source IP: 192.168.1.105
Destination IP: 128.119.245.12
TCP
Flags: [SYN]
HTTP
Source port: 1058
Destination port: 80
Frame length available to pcap API 62 , frame length sent by medium
 62

00 06 25 da af 73 00 d0  59 a9 3d 68 08 00 45 00
00 30 00 f8 40 00 80 06  c2 3a c0 a8 01 69 80 77
f5 0c 04 22 00 50 65 14  99 a6 00 00 00 00 70 02
fa f0 4d 6c 00 00 02 04  05 b4 01 01 04 02

Frame : 8
Ethernet II
Source MAC: 00 06 25 da af 73
Destination MAC: 00 d0 59 a9 3d 68
IPv4 (IHL 5)
Source IP: 128.119.245.12
Destination IP: 192.168.1.105
TCP
Flags: [SYN] [ACK]
HTTP
Source port: 80
Destination port: 1058
Frame length available to pcap API 62 , frame length sent by medium
 62

00 d0 59 a9 3d 68 00 06  25 da af 73 08 00 45 60
```

Pre TFTP komunikáciu sú vypísané všetky rámce príslušnej komunikácie, nie len prvý s cieľovým portom 69.
Výpis prvých dvoch rámcov TFTP komunikácie zo súboru eth-8.pcap:

```
TFTP communication nr.  1
Frame : 7
Ethernet II
Source MAC: 00 a1 b0 08 01 af
Destination MAC: 00 0f 3d 95 d9 99
IPv4 (IHL 5)
Source IP: 192.168.1.11
Destination IP: 192.168.1.18
UDP
TFTP
Source port: 1279
Destination port: 69
Frame length available to pcap API 63 , frame length sent by medium
 63

00 0f 3d 95 d9 99 00 a1  b0 08 01 af 08 00 45 00
00 31 46 81 00 00 80 11  70 cd c0 a8 01 0b c0 a8
01 12 04 ff 00 45 00 1d  95 f1 00 02 77 6f 72 6c
64 2e 63 70 70 00 6e 65  74 61 73 63 69 69 00

Frame : 8
Ethernet II
Source MAC: 00 0f 3d 95 d9 99
Destination MAC: 00 a1 b0 08 01 af
IPv4 (IHL 5)
Source IP: 192.168.1.18
Destination IP: 192.168.1.11
UDP
TFTP
Source port: 2050
Destination port: 1279
Frame length available to pcap API 60 , frame length sent by medium
 60

00 a1 b0 08 01 af 00 0f  3d 95 d9 99 08 00 45 00
00 20 0a 7b 40 00 40 11  ac e4 c0 a8 01 12 c0 a8
01 0b 08 02 04 ff 00 0c  6f 63 00 04 00 00 00 02
```

Pre ICMP komunikácie sú vypísané aj typy správ, napr. Time Exceeded, Request. ICMP správy typu Request a Reply sú vo výpise spojené do dvojíc tak, ako boli odoslané ping request a reply v reálnej komunikácii. Toto priradenie je zabezpečené načítaním hodnoty poľa sequence number zo 7-8 byteu ICMP hlavičky a následným ukladaním do hashovacej tabuľky (python dictionary) podľa kľúča tvoreného reťazcom v tvare: zdrojová IPv4 adresa + cieľová IPv4 adresa + ICMP sequence number.

```
Frame : 42
Ethernet II
Source MAC: cc 08 09 d4 00 00
Destination MAC: 02 00 4c 4f 4f 50
IPv4 (IHL 5)
Source IP: 12.0.0.1
Destination IP: 12.0.0.5
ICMP - Type: Destination Unreachable - Port Unreachable
Frame length available to pcap API 70 , frame length sent by medium 70

02 00 4c 4f 4f 50 cc 08  09 d4 00 00 08 00 45 c0
00 38 00 18 00 00 ff 01  a2 e7 0c 00 00 01 0c 00
00 05 03 03 4b 98 00 00  00 00 45 00 02 20 38 09
00 00 7f 11 e9 be 0c 00  00 05 0c 00 00 01 05 e6
c2 fd 02 0c e6 74

Frame : 51
Ethernet II
Source MAC: cc 08 09 d4 00 00
Destination MAC: 02 00 4c 4f 4f 50
IPv4 (IHL 5)
Source IP: 12.0.0.1
Destination IP: 12.0.0.5
ICMP - Type: Destination Unreachable - Port Unreachable
Frame length available to pcap API 70 , frame length sent by medium 70

02 00 4c 4f 4f 50 cc 08  09 d4 00 00 08 00 45 c0
00 38 00 1d 00 00 ff 01  a2 e2 0c 00 00 01 0c 00
00 05 03 03 a6 1d 00 00  00 00 45 00 02 20 38 20
```

Rámce ARP komunikácií sú zoskupené podľa zdrojovej MAC adresy a cieľovej IP adresy, ktorej MAC adresa je žiadaná.
Výpis jednej kompletnej ARP komunikácie zo súboru eth-8.pcap:

```
ARP communication nr.  1
ARP-Request, IP address:  192.168.1.11 MAC: ???
Sender IP:  192.168.1.18, Target IP:  192.168.1.11
FRAME : 11
Frame length available to pcap API 60 , frame length sent by medium 60
Ethernet II - ARP
Source MAC: 00 0f 3d 95 d9 99
Destination MAC: 00 a1 b0 08 01 af
00 a1 b0 08 01 af 00 0f  3d 95 d9 99 08 06 00 01
08 00 06 04 00 01 00 0f  3d 95 d9 99 c0 a8 01 12
00 00 00 00 00 00 c0 a8  01 0b 00 00 00 00 a0 54
00 00 a0 54 00 00 05 00  00 00 00 10

ARP-Reply, IP address:  192.168.1.11
MAC: 00 a1 b0 08 01 af
Sender IP:  192.168.1.11, Target IP:  192.168.1.18
FRAME : 12
Frame length available to pcap API 42 , frame length sent by medium 42
Ethernet II - ARP
Source MAC: 00 a1 b0 08 01 af
Destination MAC: 00 0f 3d 95 d9 99
00 0f 3d 95 d9 99 00 a1  b0 08 01 af 08 06 00 01
08 00 06 04 00 02 00 a1  b0 08 01 af c0 a8 01 0b
00 0f 3d 95 d9 99 c0 a8  01 12
```

Implementačné prostredie:

Analyzátor je implementovaný v programovacom jazyku Python3 s importovanou knižnicou struct, ktorá je použitá na konverziu byteov načítaných z .pcap súboru na celé čísla.
Rámce komuniikácií jednotlivých protokolov sú ukladané do hashovacích tabuliek (python dictionary), v ktorých je ako kľúč využitý reťazec znakov napr. v tvare :
“zdrojová IPv4 adresa+zdrojový UDP port+cieľová IPv4 adresa” pre TFTP komunikáciu alebo
“zdrojová MAC adresa+zdrojová IPv4 adresa+cieľová IPv4 adresa” pre ARP komunikáciu.

Zdroje:
[1] “Sample Captures”, Wireshark Wiki, 28. September 2019, <https://wiki.wireshark.org/SampleCaptures#TFTP>
[2] “Address Resolution Protocol”, Wikimedia Foundation, Inc., 24. Október 2019, <https://en.wikipedia.org/wiki/Address_Resolution_Protocol>
[3] “A look at the pcap file format”, Hani Benhabiles, 13. Október 2012, <http://www.kroosec.com/2012/10/a-look-at-pcap-file-format.html>

[4] “7.3. struct — Interpret strings as packed binary data”, The Python Software Foundation, 19. Október 2019, <https://docs.python.org/2/library/struct.html>
[5] “pcap struct pcap_pkthdr len vs caplen”, Stack Exchange Inc, 29. september 2009, https://stackoverflow.com/questions/1491660/pcap-struct-pcap-pkthdr-len-vs-caplen
[6] Sort dictionary by values, Stack Exchange Inc, 5. Marec 2009. https://stackoverflow.com/questions/613183/how-do-i-sort-a-dictionary-by-value