Andrii Rybak
ID: 105840

Fakulta informatiky a informačných technológií

Počítačové a komunikačné siete

Zadanie 1 – Analyzátor sieťovej komunikácie

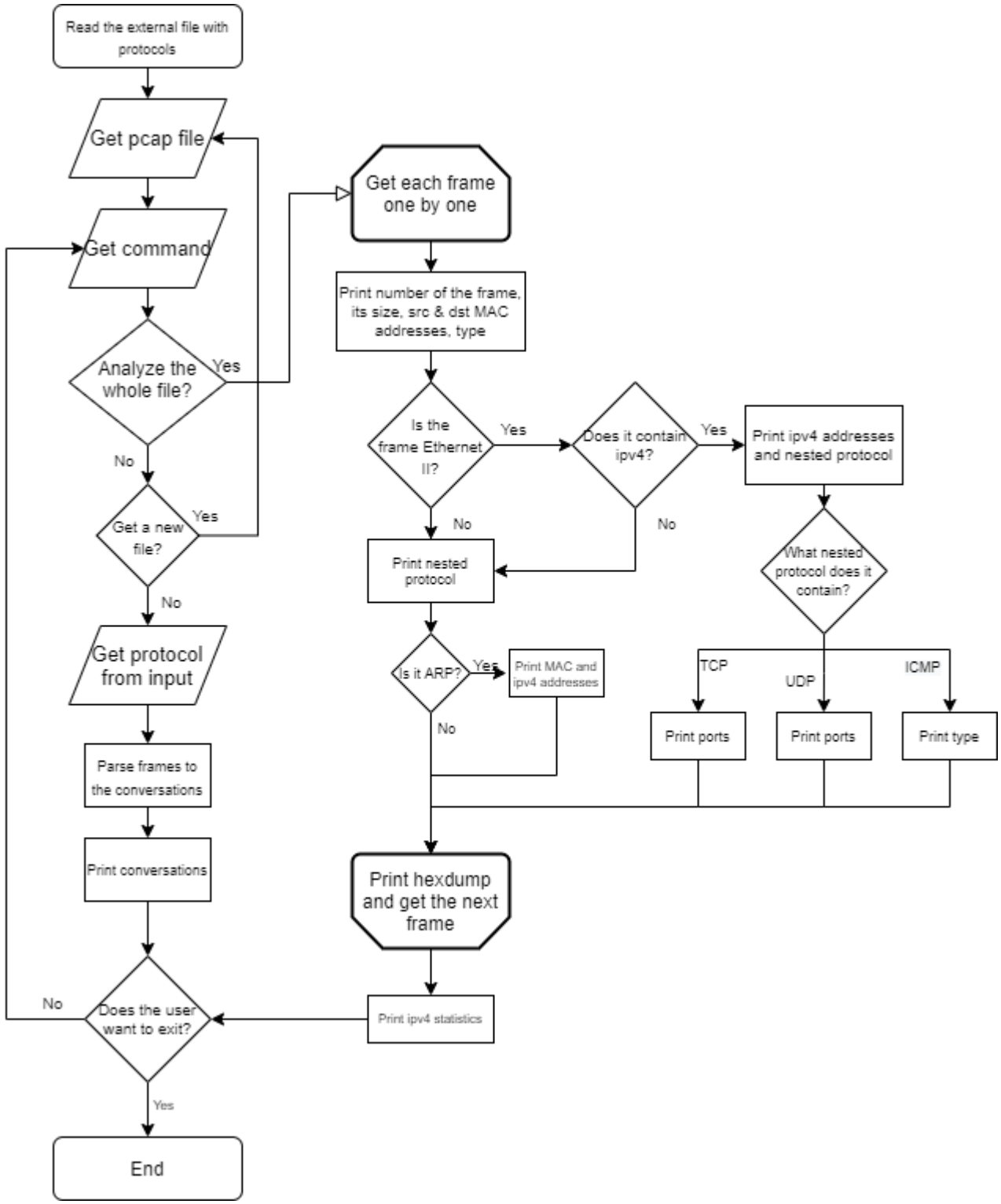
Andrii Rybak

ID: 105840

Akademicky rok: 2021/2022

Andrii Rybak
ID: 105840

Diagram of the solution



Andrii Rybak
ID: 105840

Algorithm

Each frame is analyzed layer by layer printing general information of each protocol (which are specified in the task). The general mechanism is simple – the certain data is located in certain substring of the frame's hex dump, so the only task is to print it.

In case of the conversations there are needed more sophisticated algorithms. General idea is to separate each conversation to the individual list. The new conversation List is created every time the connection was successfully established with three-way handshake (SYN, SYN ACK, ACK), then the frames with the same ip addresses and ports are added to this list. If the conversation ends with two FIN from each side or with RST(at least from one side), it is meant to be complete, otherwise if the conversation starts only with connection establishment, it is considered to be incomplete. The first complete and incomplete conversations are printed in the console.

The approach above perfectly works for HTTP, HTTPS, TELNET, SSH, FTP (data and control).

The function in the code is:

```
def convWithConnect(pcap, protocol):
```

Where pcap is a .pcap file, protocol is one of the the protocols mentioned above (HTTP, HTTPS, TELNET, SSH, FTP (data and control))

For ICMP and TFTP is used similar approach with minor changes.

For example, in TFTP conversation the program creates the new list (which means the new conversation), when the port 69 was detected. To the list are added ip addresses and ports (one of which is not 69, but the TFTP port that server has chosen to use for communication). The next frames are added to the certain conversation with corresponding data.

Function to analyze the TFTP conversations:

```
def convTFTP(pcap):
```

In case of ICMP, one conversation is recognized by the same ip addresses (src and dst).

Function to analyze the ICMP conversations:

```
def conversationICMP(pcap):
```

To pair ARP frames, all ARP frames are divided into requests and replies. Then, for each request is found a reply. If there are several identical requests, they will be printed together.

Function to analyze the ARP conversations:

```
def convARP(pcap):
```

First of all, after execution program reads the file with protocol numbers and names, and puts them to the hash tables, where index is the protocol number. Each protocol has its own hash table

Andrii Rybak
ID: 105840

Example of external file

```
#Ethertypes
0x0800 IPv4
0x0806 ARP
0x86dd IPv6
0x9000 CTP
--
#LSAPs
0x42 STP
0xaa SNAP
0xe0 IPX
--
#IP Protocol numbers
0x01 ICMP
0x06 TCP
0x11 UDP
--
#TCP ports
0x0016 SSH
0x0050 HTTP
0x01BB HTTPS
0x0017 TELNET
0x0015 FTP control
0x0014 FTP data
--
#UDP ports
0x0035 DNS
0x0045 TFTP
--
#ICMP types
0x00 Echo reply
0x08 Echo request
0x0b Time exceeded
0x03 Destination Unreachable
--
```

As can be seen, every list starts with "`#[name]`" and ends with "—"

Andrii Rybak
ID: 105840

User interface

The communication with the program is performed via console. First of all, the program requires a name of the file with protocols and .pcap file from the user, then the command can be chosen. There are 4 commands to choose:

- **0**: end
- **1**: analyze and print all frames with IPv4 statistics
- **2**: analyze and print conversations of the given protocol
- **3**: change file

If the command **2** was chosen, the wanted protocol name must be specified. It can be:

- HTTP
- HTTPS
- TELNET
- SSH
- FTP data
- FTP control
- ICMP
- ARP
- TFTP

In case of HTTP, HTTPS, TELNET, SSH, ICMP, FTP (data or control), TFTP, if the number of frames in the conversation is more than 20, first 10 and last 10 will be printed in the console.

In case of HTTP, HTTPS, TELNET, SSH, FTP (data or control), only first complete and first incomplete conversations will be printed.

In case of ARP all conversations and frames will be printed.

Implementation environment

This solution is implemented in Python 3.9, using only two external libraries `scapy` and `os.path`. The used functions from library `scapy` are `rdpcap`(to read the pcap file) and `raw`(to assemble the file).

Library `os.path` was used to check if the given file exists.

Every frame is analyzed manually by bytes.

IDE – PyCharm Community Edition 2021.2.2

POČÍTAČOVÉ A KOMUNIKAČNÉ SIETE

cvičenia

ak. rok 2021/22, zimný semester

Zadanie 1: **Analyzátor sieťovej komunikácie**

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

   IP adresy a počet odoslaných / prijatých 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 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 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.

<u>Súčasťou riešenia je aj dokumentácia, ktorá musí obsahovať najmä:</u>

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.

<u>Hodnotenie</u>

Celé riešenie - max. 15 bodov (min. 6), z toho:

- max. 3 body za riešenie úlohy v bode 1); riešenie musí byť prezentované na 4. cvičení;
- max. 1 body za doplnenú funkčnosť (doimplementáciu) priamo na cvičení v požadovanom termíne podľa harmonogramu cvičení; V prípade, ak študent nesplní úlohu zadanú priamo na cvičeniach, nehodnotí sa riešenie úlohy podľa bodu 3);
- max. 11 bodov za výsledné riešenie.

Dokumentáciu a zdrojový kód implementácie študent odovzdáva v elektronickom tvare do AISu v určenom termíne.

Bodovací kľúč:

Úloha	Body
Doplnená funkčnosť (doimplementácia) priamo na cvičení. *1 bod získa študent, ktorý doimplementuje úlohu v jej plnom rozsahu a predvedie jej funkčnosť bez toho, aby program padal alebo vyhadzoval akékoľvek chybové hlášky súvisiace s touto úlohou.*	0-1
Výpis rámcov z .pcap súboru s poradovým číslom a analýza L2 - Úlohy v bodoch 1 - 3) a výpis IP adries všetkých vysielajúcich uzlov a IP adresy uzla, ktorý odoslal najviac paketov. *5 bodov získa študent, ktorého výpis bude bez pochybností spĺňať všetky body (a – d) a bude správne naformátovaný (viď vzor na konci dokumentu) a výpis IP adries všetkých odosielajúcich uzlov a uzla, ktorý odoslal najviac paketov.* *1 bod získa študent za otvorenie .pcap súboru a vypísanie všetkých rámcov so správnym poradovým číslom.*	0-5
Uvedenie protokolu na 2-4. vrstve, zdrojové aj cieľové IP adresy a porty. *1 bod získa študent, ktorého výpis bude obsahovať všetky protokoly, adresy a porty týkajúce sa 2-4 vrstvy, ktoré sa v danom rámci nachádzajú.*	0-1
Výpis rámcov patriacich do komunikácií protokolov nad TCP podľa zadania. *2 body získa študent, ktorý vo výpise identifikuje všetky z nasledujúcich protokolov nad TCP: HTTP, HTTPS, TELNET, SSH, FTP (riadiace aj dátové) a bude ich mať vo výpise pozoskupované po jednotlivých komunikáciach. Všetky vzťahujúce sa požiadavky bodu 4 musia byť splnené.*	0-2
Výpis rámcov patriacich do komunikácií protokolu TFTP podľa zadania. *2 body získa študent, ktorý identifikuje všetky rámce jednej TFTP komunikácie a bude vedieť vo výpise prehľadne ukázať, ktoré rámce patria do ktorej komunikácie. Všetky vzťahujúce sa požiadavky bodu 4 musia byť splnené.*	0-2
Výpis rámcov patriacich do komunikácií protokolu ICMP podľa zadania. *1 bod získa študent, ktorý identifikuje všetky rámce jednej ICMP komunikácie a bude vedieť vo výpise prehľadne ukázať, ktoré rámce patria do ktorej komunikácie. Všetky vzťahujúce sa požiadavky bodu 4 musia byť splnené.*	0-1
Výpis rámcov patriacich do komunikácií protokolu ARP podľa zadania. *1 bod získa študent, ktorý identifikuje všetky rámce jednej ARP komunikácie (úplnej aj neúplnej) a bude vedieť vo výpise prehľadne ukázať, ktoré rámce patria do ktorej komunikácie. Všetky vzťahujúce sa požiadavky bodu 4 musia byť splnené.*	0-1
Finálna dokumentácia a kvalita spracovania *Hodnotí sa prehľadnosť a zrozumiteľnosť odovzdanej dokumentácie ako aj kvalita spracovania celového riešenia. 2 body získa študent, ktorý má v dokumentácií uvedené všetky podstatné informácie o fungovaní jeho programu vrátane blokového diagramu spracovávania súborov, popis jednotlivých častí zdrojového kódu (knižnice, triedy, metódy, ...). Ďalej je podmienkou získania týchto bodov predvedenie plne funkčného riešenia (splnené všetko z textu zadania) na prvý pokus, bez nutnosti reštartovať program, robiť úpravy v kóde, atď...*	0-2

Zadanie, ktoré nespĺňa ktorúkoľvek z minimálnych požiadaviek nebude prevzaté a bodované cvičiacim a bude ohodnotené 0 bodmi.

Ukážky výstupu riešenia

V ukážkach ide iba o zobrazenie požadovaného výstupu, obsah rámcov nezodpovedá reálnej komunikácii. Podobne, uvedené IP adresy v desiatkovo-bodkovej notácii nezodpovedajú reálnym hodnotám v rámci.

Výpis celej komunikácie – k bodu 1)

rámec 1
dĺžka rámca poskytnutá pcap API – 68 B
dĺžka rámca prenášaného po médiu – 72 B
Ethernet II
Zdrojová MAC adresa: 00 00 C0 D7 80 C2
Cieľová MAC adresa: 00 04 76 A4 E4 8C
IPv4
zdrojová IP adresa: 192.168.1.33
cieľová IP adresa: 147.175.1.55
TCP

```
00 04 76 A4 E4 8C 00 00   C0 D7 80 C2 08 00 45 00
00 28 0C 36 40 00 80 06   2B 5A 93 AF 62 EE 45 38
87 6A 04 70 00 50 7E 6C   06 32 56 7D 30 A8 50 10
44 70 97 A0 00 00 80 C2   08 0C 36 40 30 A3 23 35
A2 D5 27 81
```

rámec 2
dĺžka rámca poskytnutá pcap API – 494 B
dĺžka rámca prenášaného po médiu – 498 B
IEEE 802.3 – Raw
Zdrojová MAC adresa: 00 04 76 A4 E4 8C
Cieľová MAC adresa: FF FF FF FF FF FF

```
FF FF FF FF FF FF 00 04   76 A4 E4 8C 01 E0 FF FF
01 E0 00 A1 40 00 80 06   05 B0 93 AF 62 EE 93 AF
63 2A 04 4C 00 50 73 78   .......
```

rámec 3
dĺžka rámca poskytnutá pcap API – 62 B
dĺžka rámca prenášaného po médiu – 66 B
Ethernet II
Zdrojová MAC adresa: 00 00 C0 D7 80 C2
Cieľová MAC adresa: 00 04 76 A4 E4 8C
IPv4
zdrojová IP adresa: 192.168.1.33
cieľová IP adresa: 147.175.1.55
TCP

```
00 04 76 A4 E4 8C 00 00   C0 D7 80 C2 08 00 45 00
00 30 07 A1 40 00 80 06   05 B0 93 AF 62 EE 93 AF
```

```
63 2A 04 4C 00 50 73 78   17 88 00 00 00 00 70 02
40 00 C6 0B 00 00 02 04   05 B4 01 01 04 02
```

rámec 4
dĺžka rámca z poskytnutá pcap API – 60 B
dĺžka rámca prenášaného po médiu – 64 B
Ethernet II
Zdrojová MAC adresa: 00 04 76 A4 E4 8C
Cieľová MAC adresa: 00 00 C0 D7 80 C2
IPv4
zdrojová IP adresa: 192.168.1.33
cieľová IP adresa: 147.175.1.55
UDP

```
00 00 C0 D7 80 C2 00 04   76 A4 E4 8C 08 00 45 00
00 2C F0 EA 00 00 3F 11   9D 6A 93 AF 63 2A 93 AF
62 EE 00 50 04 4C 41 59   C9 42 73 78 17 89 60 12
40 00 D0 65 00 00 02 04   05 B4 00 00
```

rámec 5
dĺžka rámca poskytnutá pcap API – 54 B
dĺžka rámca prenášaného po médiu – 64 B
Ethernet II
Zdrojová MAC adresa: 00 00 C0 D7 80 C2
Cieľová MAC adresa: 00 04 76 A4 E4 8C
IPv4
zdrojová IP adresa: 192.168.1.33
cieľová IP adresa: 147.175.1.55
TCP

```
00 04 76 A4 E4 8C 00 00   C0 D7 80 C2 08 00 45 00
00 28 0C 36 40 00 80 06   2B 5A 93 AF 62 EE 45 38
87 6A 04 70 00 50 7E 6C   06 32 56 7D 30 A8 50 10
44 70 97 A0 00 00
```

IP adresy vysielajúcich uzlov:
147.175.10.3
193.45.10.10
……
27.30.44.12
210.20.66.8

Adresa uzla s najväčším počtom odoslaných paketov:
193.45.10.10 94 paketov

Výpis HTTP komunikácie – k bodu 3a)

rámec 5

dĺžka rámca poskytnutá pcap API – 62 B
dĺžka rámca prenášaného po médiu – 66 B
Ethernet II
Zdrojová MAC adresa: 00 14 38 06 E0 93
Cieľová MAC adresa: 00 02 CF AB A2 4C
IPv4
zdrojová IP adresa: 192.168.1.33
cieľová IP adresa: 147.175.1.55
TCP
HTTP
zdrojový port: 1376
cieľový port: 80

```
00 02 CF AB A2 4C 00 14   38 06 E0 93 08 00 45 00
00 30 8D 68 40 00 80 06   16 B0 C0 A8 01 21 93 AF
01 37 05 60 00 50 0A 16   B1 1B 00 00 00 00 70 02
FF FF 6B 8E 00 00 02 04   05 B4 01 01 04 02
```

rámec 8
dĺžka rámca poskytnutá pcap API – 697 B
dĺžka rámca prenášaného po médiu – 701 B
Ethernet II
Zdrojová MAC adresa: 00 14 38 06 E0 93
Cieľová MAC adresa: 00 02 CF AB A2 4C
IPv4
zdrojová IP adresa: 192.168.1.33
cieľová IP adresa: 147.175.1.55
TCP
HTTP
zdrojový port: 1376
cieľový port: 80

```
00 02 CF AB A2 4C 00 14   38 06 E0 93 08 00 45 00
02 AB 8D 6A 40 00 80 06   14 33 C0 A8 01 21 93 AF
01 37 05 60 00 50 0A 16   B1 1C FC 0E FC 3B 50 18
FF FF DF 73 00 00 47 45   54 20 2F 62 75 78 75 73
2F 67 65 6E 65 72 61 74   65 5F 70 61 67 65 2E 70
68 70 3F 70 61 67 65 5F   69 64 3D 31 20 ....
```

rámec 15
dĺžka rámca poskytnutá pcap API – 60 B
dĺžka rámca prenášaného po médiu – 64 B
Ethernet II
Zdrojová MAC adresa: 00 02 CF AB A2 4C
Cieľová MAC adresa: 00 14 38 06 E0 93
IPv4
zdrojová IP adresa: 147.175.1.55

cieľová IP adresa: 192.168.1.33
TCP
HTTP
zdrojový port: 80
cieľový port: 1376

```
00 14 38 06 E0 93 00 02   CF AB A2 4C 08 00 45 00
00 28 E6 16 40 00 3A 06   04 0A 93 AF 01 37 C0 A8
01 21 00 50 05 60 FC 0E   FD 2F 0A 16 B3 A0 50 10
1B A1 80 DE 00 00 00 00   00 00 00 00
```

Výpis ARP dvojíc – k bodu 3i)

Komunikácia č. 1
ARP-Request, IP adresa: 147.175.98.232, MAC adresa: ???
Zdrojová IP: 147.175.98.238, Cieľová IP: 147.175.98.232
rámec 5
dĺžka rámca poskytnutá pcap API – 42 B
dĺžka rámca prenášaného po médiu – 64 B
Ethernet II
ARP
Zdrojová MAC adresa: 00 00 c0 d7 80 c2
Cieľová MAC adresa: ff ff ff ff ff ff

```
ff ff ff ff ff ff 00 00   c0 d7 80 c2 08 06 00 01
08 00 06 04 00 01 00 00   c0 d7 80 c2 93 af 62 ee
00 00 00 00 00 00 93 af   62 e8
```

rámec 6
dĺžka rámca poskytnutá pcap API – 42 B
dĺžka rámca prenášaného po médiu – 64 B
Ethernet II
ARP
Zdrojová MAC adresa: 00 00 c0 d7 80 c2
Cieľová MAC adresa: ff ff ff ff ff ff

```
ff ff ff ff ff ff 00 00   c0 d7 80 c2 08 06 00 01
08 00 06 04 00 01 00 00   c0 d7 80 c2 93 af 62 ee
00 00 00 00 00 00 93 af   62 e8
```

ARP-Reply, IP adresa: 147.175.98.232, MAC adresa: 00 04 76 13 97 df
Zdrojová IP: 147.175.98.232, Cieľová IP: 147.175.98.238
rámec 9
dĺžka rámca poskytnutá pcap API – 60 B
dĺžka rámca prenášaného po médiu – 64 B
Ethernet II
ARP
Zdrojová MAC adresa: 00 04 76 13 97 df

Cieľová MAC adresa: 00 00 c0 d7 80 c2

```
00 00 c0 d7 80 c2 00 04    76 13 97 df 08 06 00 01
08 00 06 04 00 02 00 04    76 13 97 df 93 af 62 e8
00 00 c0 d7 80 c2 93 af    62 ee 00 00 00 00 00 00
00 00 00 00 00 00 00 00    00 00 00 00
```

Komunikácia č. 2

ARP-Request, IP adresa: 147.175.98.238, MAC adresa: ???

Zdrojová IP: 147.175.98.231, Cieľová IP: 147.175.98.238

rámec 20

dĺžka rámca poskytnutá pcap API – 60 B

dĺžka rámca prenášaného po médiu – 64 B

Ethernet II

ARP

Zdrojová MAC adresa: 00 00 c0 d7 80 c2

Cieľová MAC adresa: ff ff ff ff ff ff

```
ff ff ff ff ff ff 00 00   c0 d7 80 c2 ...
```

ARP-Request, IP adresa: 147.175.98.238, MAC adresa: ???

Zdrojová IP: 147.175.98.231, Cieľová IP: 147.175.98.238

rámec 21

dĺžka rámca poskytnutá pcap API – 60 B

dĺžka rámca prenášaného po médiu – 64 B

Ethernet II

ARP

Zdrojová MAC adresa: 00 00 c0 d7 80 c2

Cieľová MAC adresa: ff ff ff ff ff ff

```
ff ff ff ff ff ff 00 00   c0 d7 80 c2 ...
```

ARP-Reply IP adresa: 147.175.98.238, MAC adresa: 00 04 76 23 ab ef

Zdrojová IP: 147.175.98.238, Cieľová IP: 147.175.98.231

rámec 24

dĺžka rámca poskytnutá pcap API – 60 B

dĺžka rámca prenášaného po médiu – 64 B

Ethernet II

ARP

Zdrojová MAC adresa: 00 04 76 23 ab ef

Cieľová MAC adresa: 00 00 c0 d7 80 c2

```
00 00 c0 d7 80 c2 00 04  76 23 ab ef ...
```

Príklad možného formátovania externých súborov

#Ethertypes
0x0800 IPv4
0x0806 ARP
0x86dd IPv6
#LSAPs
0x42 STP
0xaa SNAP
0xe0 IPX
#IP Protocol numbers
0x01 1 ICMP
0x06 6 TCP
0x11 17 UDP
#TCP ports
0x0015 22 SSH
0x0050 80 HTTP
#UDP ports
0x0035 53 DNS
0x0045 69 TFTP

Literatúra (Internet zdroje; Dokumentový server AIS):

https://www.tcpdump.org/
https://www.winpcap.org
https://scapy.net/
https://www.wireshark.org/
RFC dokumenty (pre analyzované protokoly)