
Dokumentácia k druhému projektu predmetu  Počítačové a komunikačne siete

Matej Slivka

xslivk03

24.4.2022

# Obsah

# Informácie z naštudovanej literatúry

Projekt ma za úlohu vyhľadávať a vypisovať packety ktoré sú poslané na určitom rozhraní. Projekt podporuje len hľadanie na Ethernet protokole. Zo sieťovej vrstvy sú to konkrétne protokoly IP,ARP ICMP . Ďalej ak sa jedna o IP protokol tak sa vypisujú pakety UDP a TCP protokolu ktoré sa nachádzajú na prenosovej vrstve. Teda projekt pracuje postupne od nižšej vrstvy až po vyššiu.

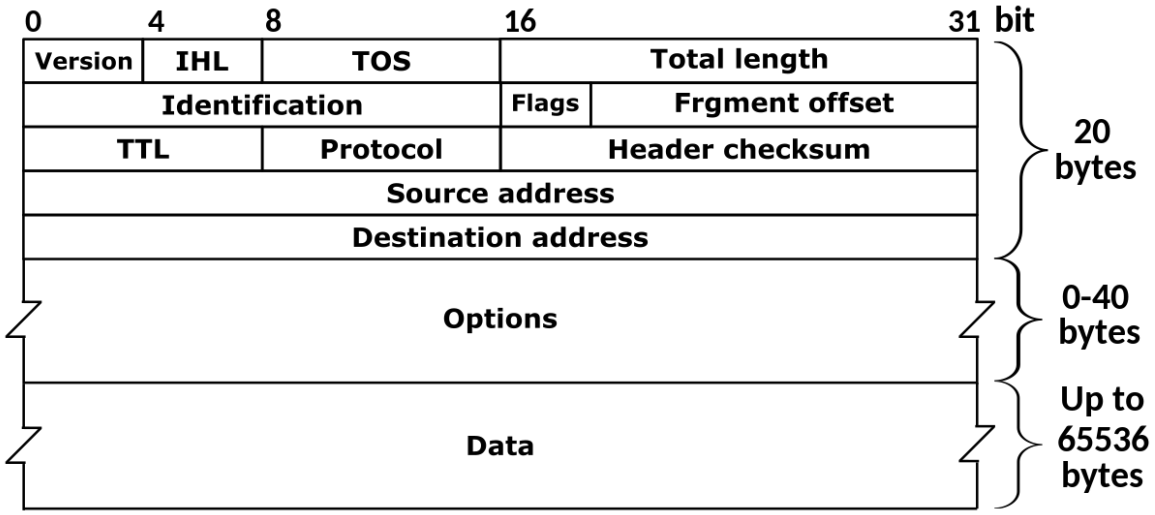
| OSI-Layer | TCP/IP-Layer | Example |
|---|---|---|
| Application (7) | Application | HTTP, UDS, FTP, SMTP, POP, Telnet, DHCP, OPC UA |
| Presentation (6) | | |
| Session (5) | | |
| | | SOCKS |
| Transport (4) | Transport | TCP, UDP, SCTP |
| Network (3) | Internet | IP (IPv4, IPv6), ICMP (über IP) |
| Data Link (2) | Link Layer (Network Interface) | Ethernet, Token Bus, Token Ring, FDDI |
| Physical (1) | | |

## Protokoly

Pri posielaní dát medzi zdrojom a cieľom sú data enkapsulované s množstvom meta dát. Bity rôznych protokolov popisujúce rovnaký parameter sa nachádzajú na rôznych miestach hlavičiek, teda pre extrahovanie týchto dát je vždy potrebné namapovať daný packet na inú štruktúru.

### IPv4 - Internet protokol v 4 [1]

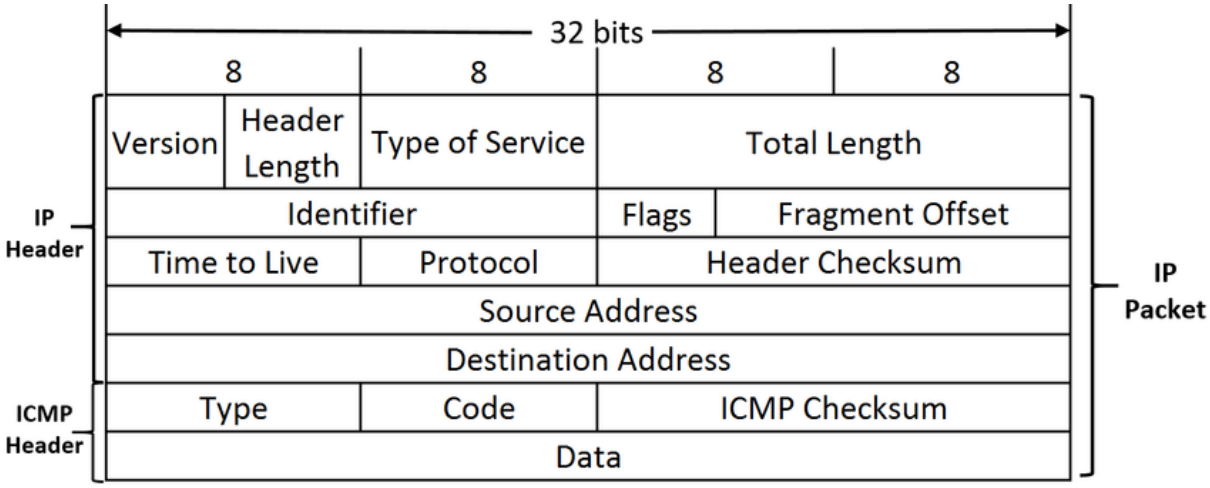
Tu si môžeme všimnúť podobnosť medzi IPv4 a ICMP protokolom.



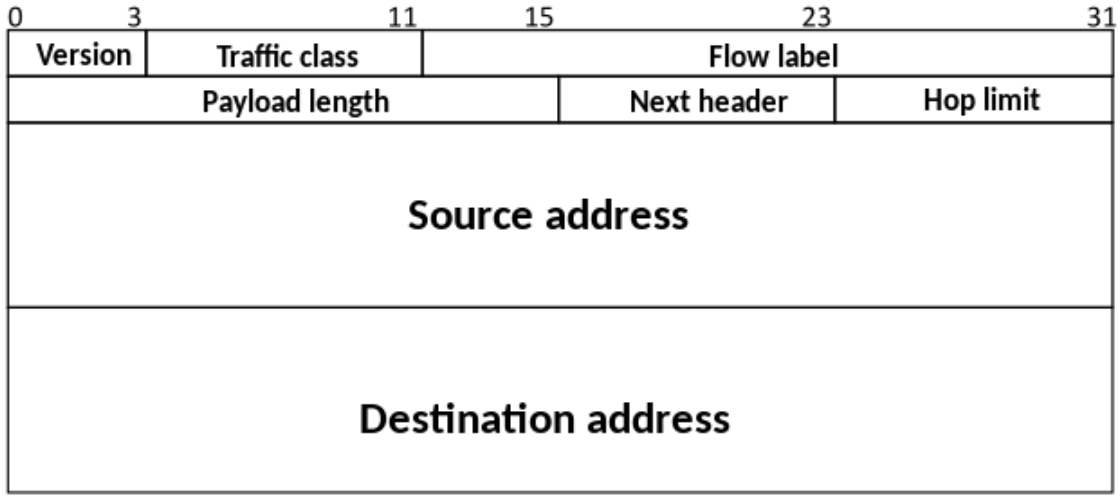

[1] https://wis.fit.vutbr.cz/FIT/st/cfs.php.cs?file=%2Fcourse%2FIPK-IT%2Flectures%2FIPK2021-04-IPv4.pdf&cid=14678

## ICMP – Internet control message protocol

Tu si môžeme všimnúť podobnosť medzi IPv4 a ICMP protokolom.

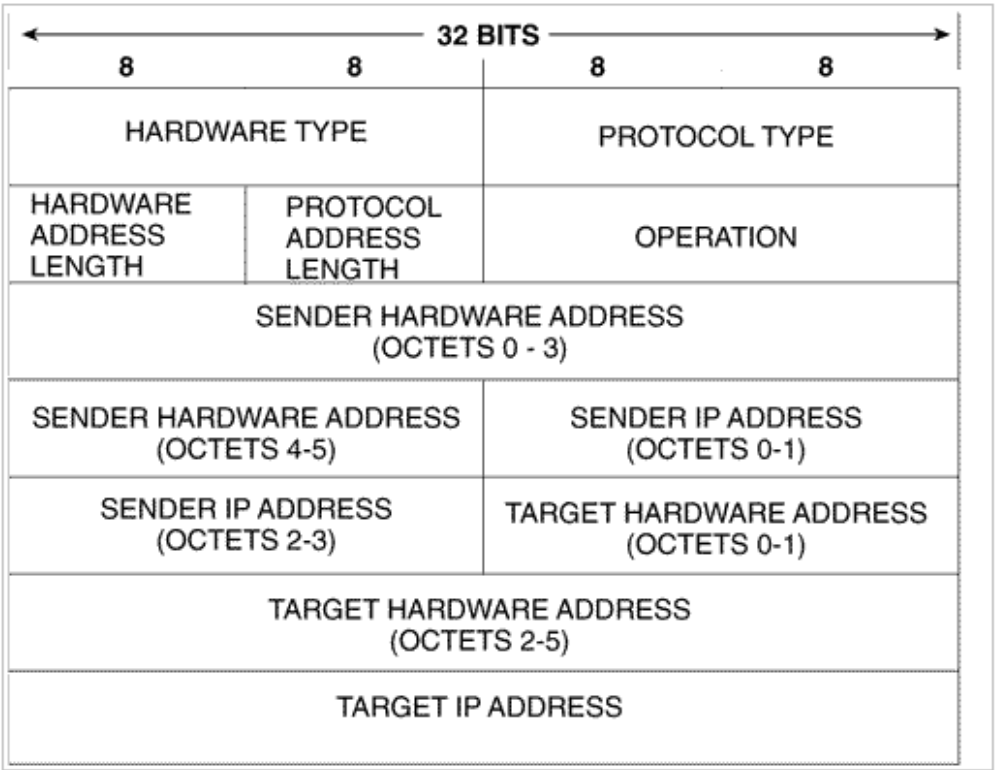



## IPv6 – Internet protokol v 6 [2]




[2] https://wis.fit.vutbr.cz/FIT/st/cfs.php.cs?file=%2Fcourse%2FIPK-IT%2Flectures%2FIPK2021-06-IPv6.pdf&cid=14678

ARP – Adress resolution protocol



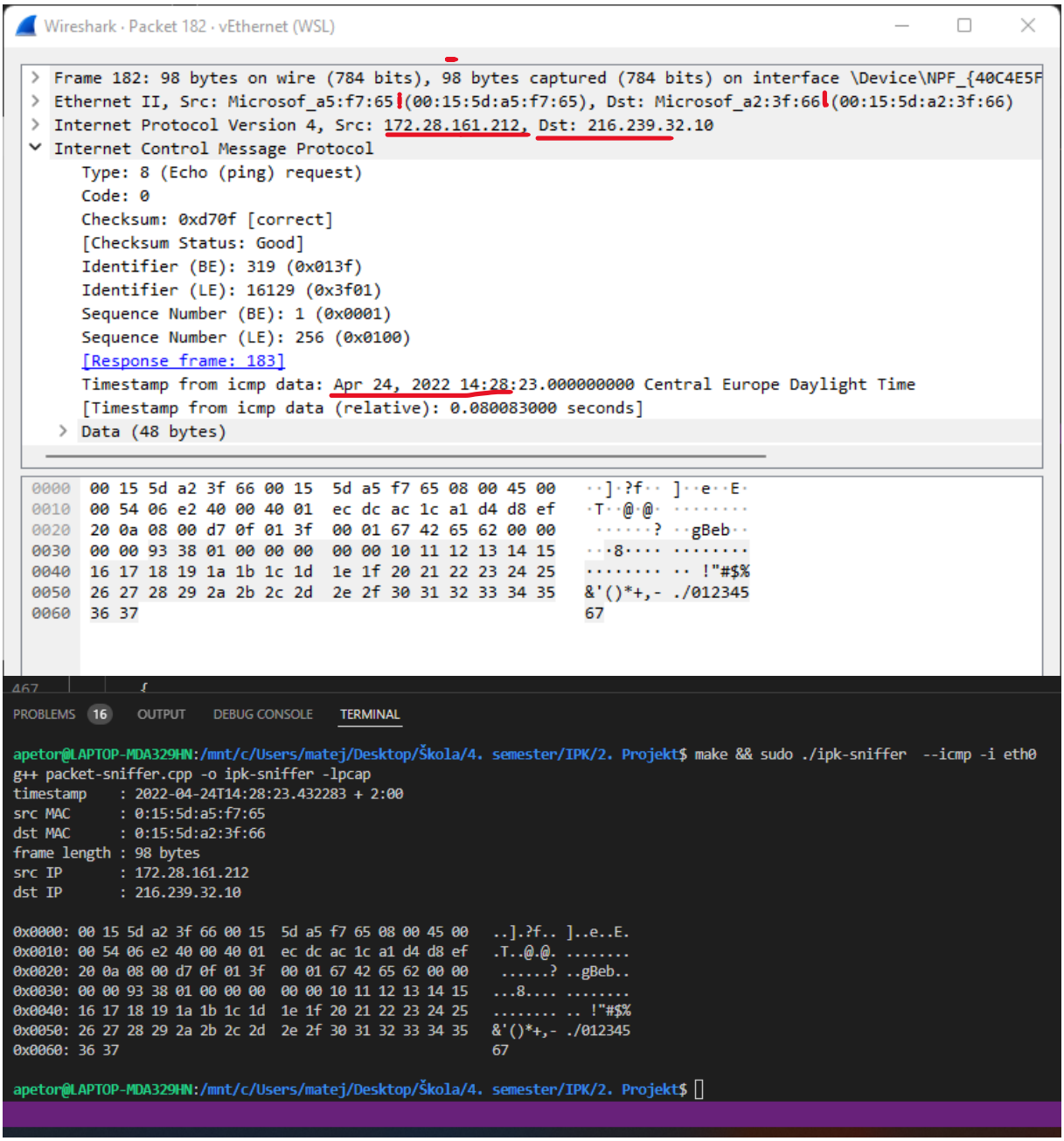

# Popis implementácie

Implementácia bola vytvorená pomocou stránky tcpdump.org[3] kde bola popísaná implementácia sniffer packet. Program sa pripojí na dané rozhranie z ktorého si vyfiltruje potrebné pakety. Tieto packet sú potom namapované na predom vytvorené štruktúry. V týchto štruktúrach sú už vytvorené premenné ktoré máju dĺžku jednotlivých argumentov packet header (ako napr. IP adresa odosielateľa, číslo port príjemcu). Týmto spôsobom sa extrahujú dáta z packetov. Data sú následne vypísané v hexadecimalnom tvare, pričom za každým riadkom sú tieto symboly aj ASCII podobe (ak sú tlačitelné).

# Testovanie

Pri vytváraní som mal problém spojazdniť referenčný image a teda môj projekt som implementoval pomocou WSL[4] a pomocou Wiresharku[5]. Príklad packetu ktorý zachytil Wireshark a ktorý zachytil môj program:

[3] https://www.tcpdump.org/pcap.html
[4] https://ubuntu.com/wsl
[5] https://www.wireshark.org/


Wireshark · Packet 182 · vEthernet (WSL)
Frame 182: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface \Device\NPF_{40C4E5F
Ethernet II, Src: Microsof_a5:f7:65 (00:15:5d:a5:f7:65), Dst: Microsof_a2:3f:66 (00:15:5d:a2:3f:66)
Internet Protocol Version 4, Src: 172.28.161.212, Dst: 216.239.32.10
Internet Control Message Protocol
Type: 8 (Echo (ping) request)
Code: 0
Checksum: 0xd70f [correct]
[Checksum Status: Good]
Identifier (BE): 319 (0x013f)
Identifier (LE): 16129 (0x3f01)
Sequence Number (BE): 1 (0x0001)
Sequence Number (LE): 256 (0x0100)
[Response frame: 183]
Timestamp from icmp data: Apr 24, 2022 14:28:23.000000000 Central Europe Daylight Time
[Timestamp from icmp data (relative): 0.080083000 seconds]
Data (48 bytes)
0000 00 15 5d a2 3f 66 00 15 5d a5 f7 65 08 00 45 00
0010 00 54 06 e2 40 00 40 01 ec dc ac 1c a1 d4 d8 ef
0020 20 0a 08 00 d7 0f 01 3f 00 01 67 42 65 62 00 00
0030 00 00 93 38 01 00 00 00 00 00 10 11 12 13 14 15
0040 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25
0050 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35
0060 36 37
467
PROBLEMS 16 OUTPUT DEBUG CONSOLE TERMINAL
apetor@LAPTOP-MDA329HN:/mnt/c/Users/matej/Desktop/Škola/4. semester/IPK/2. Projekt$ make && sudo ./ipk-sniffer --icmp -i eth0
g++ packet-sniffer.cpp -o ipk-sniffer -lpcap
timestamp : 2022-04-24T14:28:23.432283 + 2:00
src MAC : 0:15:5d:a5:f7:65
dst MAC : 0:15:5d:a2:3f:66
frame length : 98 bytes
src IP : 172.28.161.212
dst IP : 216.239.32.10
0x0000: 00 15 5d a2 3f 66 00 15 5d a5 f7 65 08 00 45 00
0x0010: 00 54 06 e2 40 00 40 01 ec dc ac 1c a1 d4 d8 ef
0x0020: 20 0a 08 00 d7 0f 01 3f 00 01 67 42 65 62 00 00
0x0030: 00 00 93 38 01 00 00 00 00 00 10 11 12 13 14 15
0x0040: 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25
0x0050: 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35
0x0060: 36 37
apetor@LAPTOP-MDA329HN:/mnt/c/Users/matej/Desktop/Škola/4. semester/IPK/2. Projekt$

# Bibliografia

KUROSE ROSS Computer Networking, A top approach featuring internet . Dostupné na internete: https://eclass.teicrete.gr/modules/document/file.php/TP326/%CE%98%CE%B5%CF%89%CF%81%CE%AF%CE%B1%20(Lectures)/Computer_Networking_A_Top-Down_Approach.pdf Addison-wesley , 2003

Veselý, 4. Prednáška https://wis.fit.vutbr.cz/FIT/st/cfs.php.cs?file=%2Fcourse%2FIPK-IT%2Flectures%2FIPK2021-04-IPv4.pdf&cid=14678 2021

Veselý 6. Prednáška https://wis.fit.vutbr.cz/FIT/st/cfs.php.cs?file=%2Fcourse%2FIPK-IT%2Flectures%2FIPK2021-06-IPv6.pdf&cid=14678 2021

Tim Carstens Programming with pcap https://www.tcpdump.org/pcap.html 2002 2002