# ISA-TOP

## 1. Úvod

Autor: Jakub Hamadej, xhamad03

Datum: 11-11-2024

## 2. Obsah

1. Úvod....1
2. Obsah....1
3. Uvedení do problematiky....2
4. Návrh aplikace....2
    4.1 Celková architektura....2
    4.2 Popis modulů a jejich vztahů....2
    4.3 Diagramy....3
5. Popis implementace....3
    5.1 Struktura projektu....3
    5.2 Použité technologie....3
    5.3 Klíčové části kódu....3
6. Základní informace o programu....4
7. Návod na použití....4
    7.1 Instalace a spuštění....4
    7.2 Popis uživatelského rozhraní....4
    7.3 Typické scénáře použití....4
8. Popis testování aplikace....4
9. Výsledky testů....5
10. Přehled nastudovaných informací z literatury....5
    10.1 Ethernet II Frame [1]....5
    10.2 IPv4 packet [2]....5
    10.3 IPv6 packet [3]....6
    10.4 UDP protokol [4]....7
    10.4 TCP protokol [5]....7
    10.4 ICMP protokol [6]....7
11. Literatura....8

# 3. Uvedení do problematiky

Úkolem je vytvořit program, který zachytává packety na rozhraní (interface), zaznamenává si detaily o IP adresách, jejich velikosti a případně také port, pokud ho daný protokol má. Následně každou vteřinu se zobrazují statistiky deseti nejaktivnějších spojení.

# 4. Návrh aplikace

## 4.1 Celková architektura

Program využívá kromě standardní C/C++ knihovny také knihovny ***libpcap*** a ***ncurses***.
K zachytávání packetů je využíváno funkcí z knihovny ***libpcap***, kdy tato data následně předá metodě ***addRecord()*** ze třídy ***IsaTop***.

## 4.2 Popis modulů a jejich vztahů

Aplikace je rozdělena do zdrojových souborů ***main.cpp***, tří zdrojových souborů (**isa_top.cpp**, **packet_handler.cpp**, **record.cpp**) a tří hlavičkových souborů (**isa_top.hpp**, **packet_handler.hpp**, **record.hpp**).

**Main** čerpá z **isa_top.hpp** a **packet_handler.hpp**.

**Packet_handler** z **isa_top.hpp** a **record.hpp**. Zpracovává příchozí packet pro pozdější uložení.

**Isa_top** je přímo napojená na **record.hpp**. Je odpovědná zaz manipulaci s celým záznamem komunikací, přidáváním, řazením a mazáním, případně zobrazením na výstupu.

**Record** třída odpovědná za manipulaci s jednotlivými položkami a případně jejich porovnání.

## 4.3 Diagramy

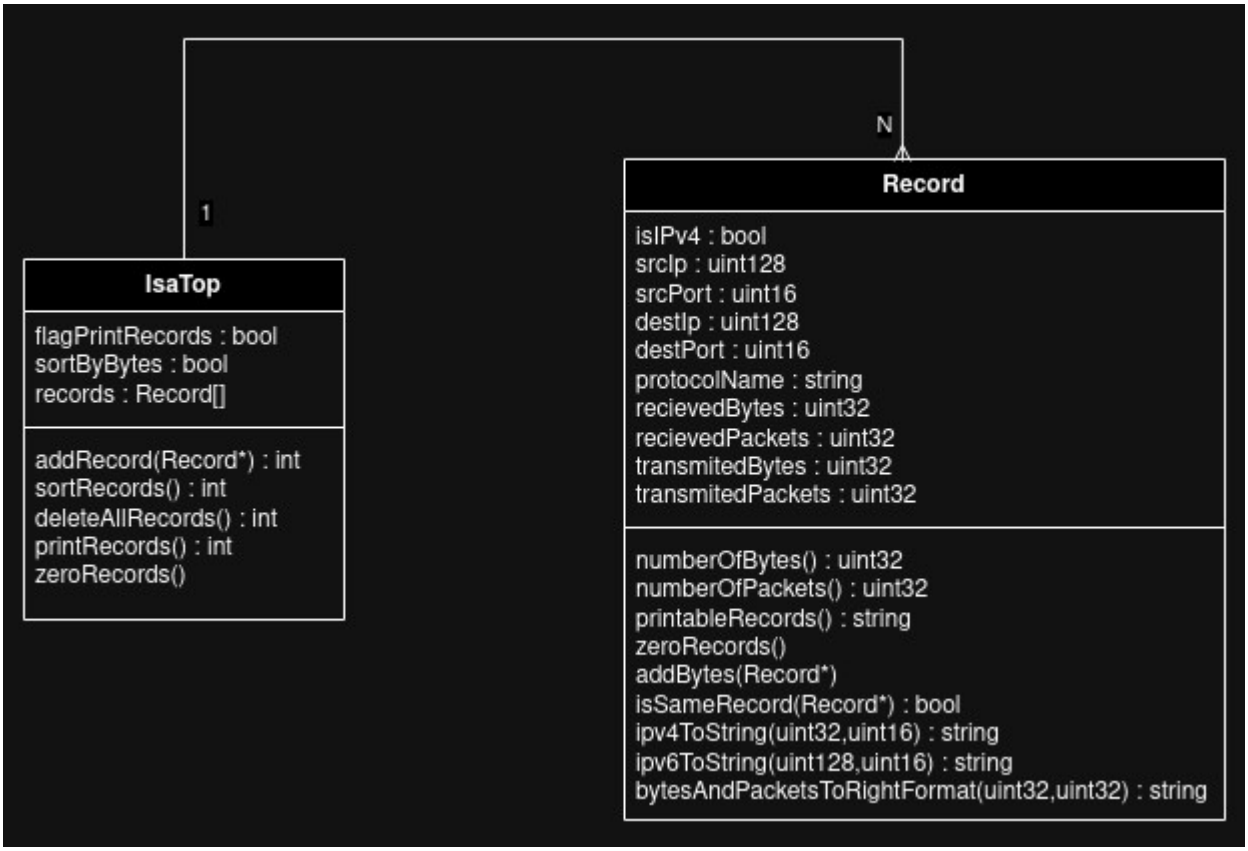



Diagram popisuje vztah mezi třídou IsaTop a třídou Record.

# 5. Popis implementace

## 5.1 Struktura projektu

Všechny zdrojové kódy včetně hlavičkových souborů, Makefile, této dokumentace a README, které je v tomto případě uloženo pod jménem isa-top.1 jako manuálový soubor, jsou ukoženy v jednom adresáři nestrukturovaně

## 5.2 Použité technologie

Program je psán v C/C++ a překládán pomocí g++. Jedná se o čistě konzolovou aplikaci bez použití frameworku nebo něčeho podobného.

## 5.3 Klíčové části kódu

Klíčovými částmi kódu jsou funkce **packet_handler()**, který obstarává pro funkci **pcap_loop()** zpracování packetu, kdy přijmutý packet pomoci ukazatelů rozebere a získá z nich potřebné informace

pro vytvoření objektu třídy **Record** a následně zavolá metoru **addRecord()** od třídy **IsaTop**, aby ji přidala do seznamu záznamů.

Dalšími důležitými funkcemi jsou dvě přerušení, jedno je časové, **next_second_alarm()**, které se zavolá každou sekundu, načež se vytiskne aktualizovaný seznam pro uživatele, vynulují se stávající hodnoty a nastavý se flag na vymazání záznamů, při nejbližší bezpečné příležitosti.

Druhá funkce, **terminal_alarm()**, pak zařizuje vypnutí celého programu pomocí CTRL+C tak, aby se korektně uklidila paměť. Toho dosahuje tím, že vyvolá funkci **pcap_breakloop()** a celé naslouchání se ukončí a program dojde do svého korektního ukončení.

# 6. Základní informace o programu

Jedná se o konzolovou aplikaci bez ovládacích prvků. Program zachytává packety na zvoleném rozhraní, ze kterých vytváří záznam. Každou vteřinu 10 nejaktivnějších komunikací za uplynulou vteřinu vypíše.

# 7. Návod na použití

## 7.1 Instalace a spuštění

Program zkompilujete příkazem ***make***.

Následné spuštění poté provedete příkazem ./***isa-top***. Mějte na paměti, že ***-i interface*** je povinný argument.

Pro další nápovědu přímo v programu zadejte ***./isa-top -h*** nebo ***./isa-top --help***

## 7.2 Popis uživatelského rozhraní

Jedná se o konzolovou aplikaci bez ovládání. K ukončení aplikace stiskněte CTRL + C.

## 7.3 Typické scénáře použití

Aplikaci typicky použijete ve chvíli, kdy chcete monitorovat, aktuální prvoz na síti, respektive chcete zjistit, jestli a co na zadaném interface probíhá.

# 8. Popis testování aplikace

Testy probíhaly na operačním systému Fedora 40, s kompilátorem g++ (GCC) 14.2.1 20240912 (Red Hat 14.2.1-3)

Testování proběhlo manuálně, kdy zdrojový kód byl upraven tak, aby zobrazil druhou sekundu výstupu předem nahraného záznamu packetů.

Další testy poté proběhly za běžného používání prohlížeče, kdy bylo testováno, jestli nedochází k pádu systému, nebo únikům paměti.

# 9. Výsledky testů

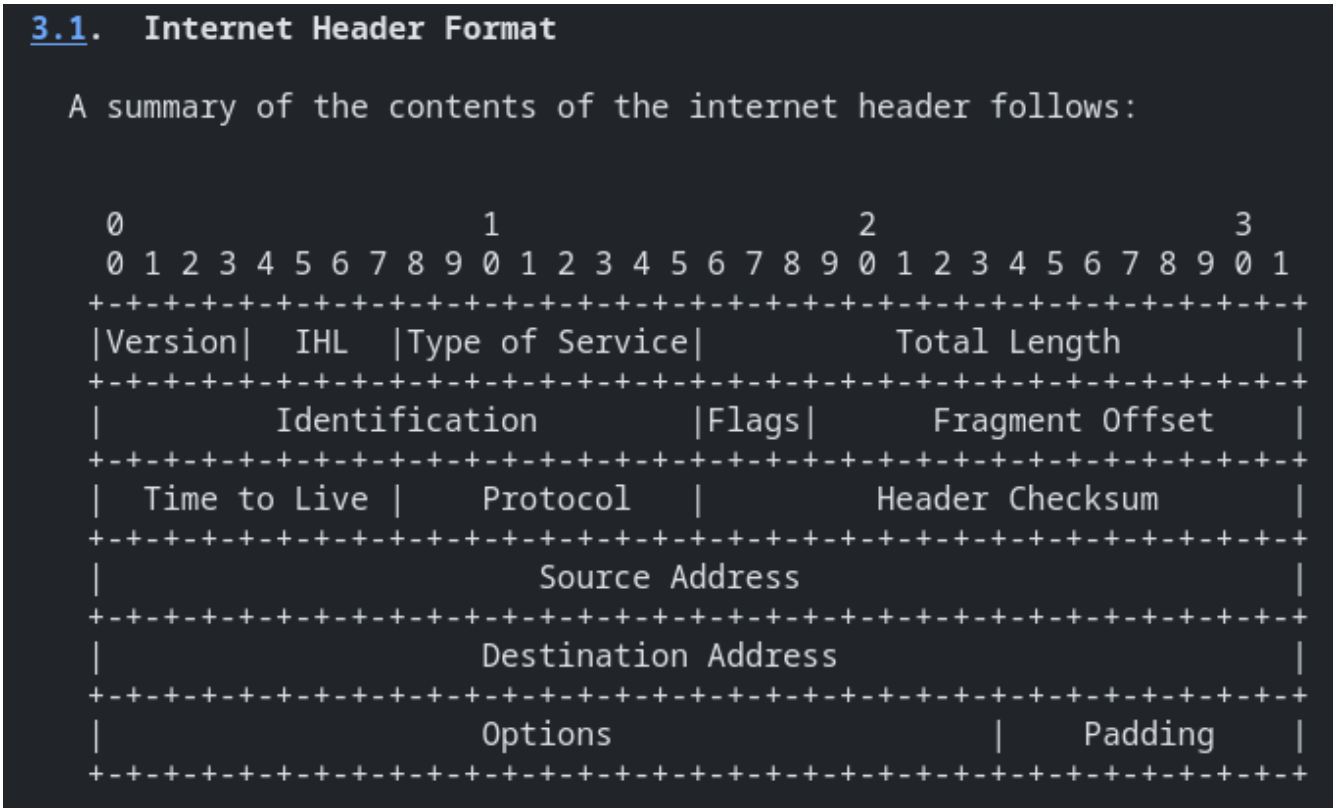
Testy dopadli dle očekávání. Jediné úniky paměti, které byli zaznamenány pocházejí přímo od knihovny ***ncurses***, se kterými však nelze nic udělat.

# 10. Přehled nastudovaných informací z literatury

## 10.1 Ethernet II Frame [1]

Velikost hlavičky je 14 bajtů. O tuto hodnotu je potřeba provést offset na paketu, abychom se dostali na síťovou část/vrstvu paketu.

## 10.2 IPv4 packet [2]




Síťová část IPv4 paketu má promněnou velikost 20 až 60 bajtů. Je rozdělen do slov (words), kdy jedno slovo má 4 bajty.

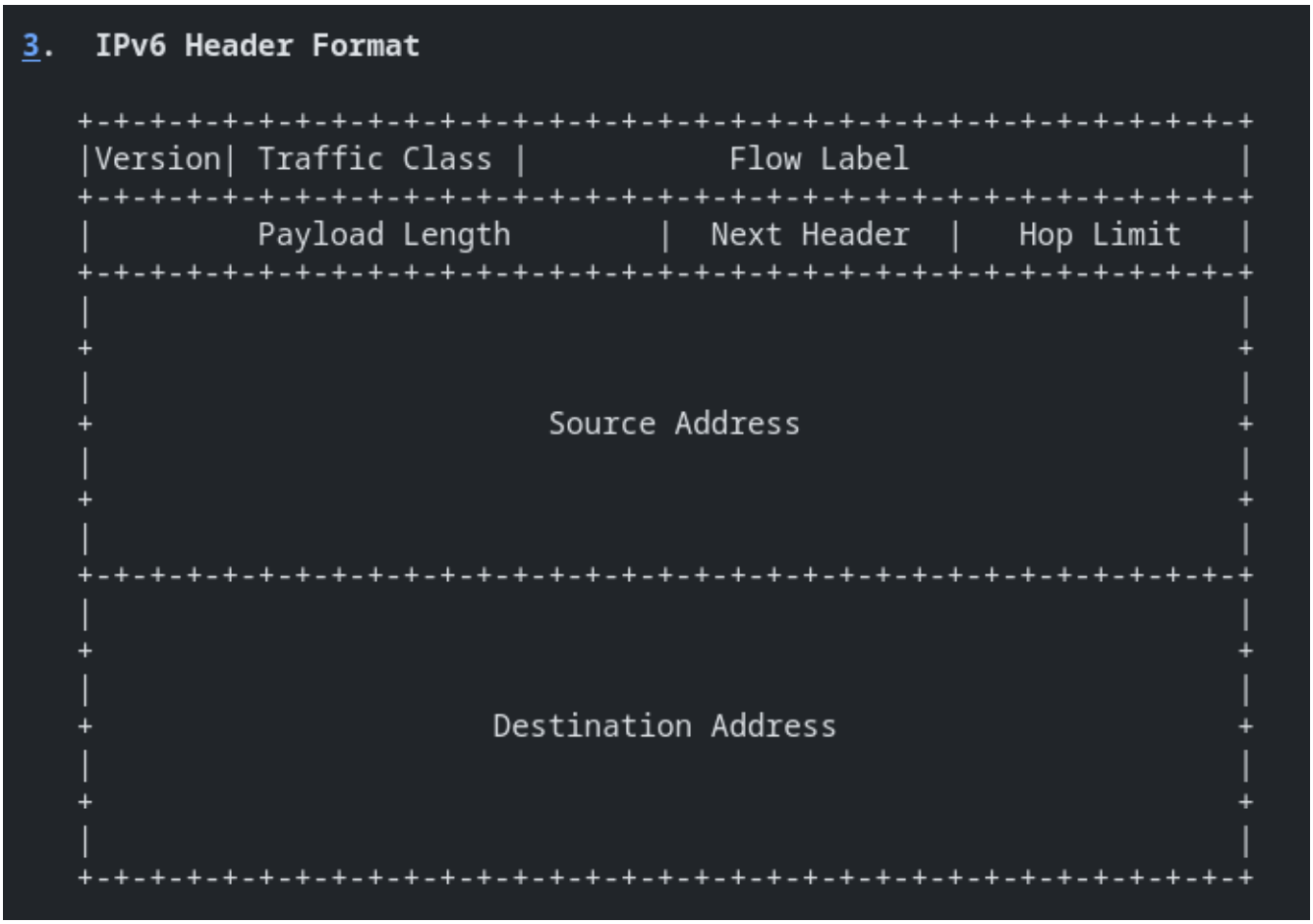
Version část musí mít pro IPv4 hodnotu 4.

V IHL části je uvedeno, kolik slov hlavička obsahuje, nejmenší hodnota je 5, nejvyšší 15 (0xF).

Na offsetu 12 nalezneme poté zdrojovou adresu a na offsetu 16 cílovou adresu.

Hodnoty jsou uloženy jako Big Endien.

Na offsetu 9 je uveden protokol(typ zprávy), kterou daný paket přenáší.




## 10.3 IPv6 packet [3]

Síťová část IPv4 paketu má pevnou velikost 40 bajtů. Je rozdělen do slov (words), kdy jedno slovo má 4 bajty.

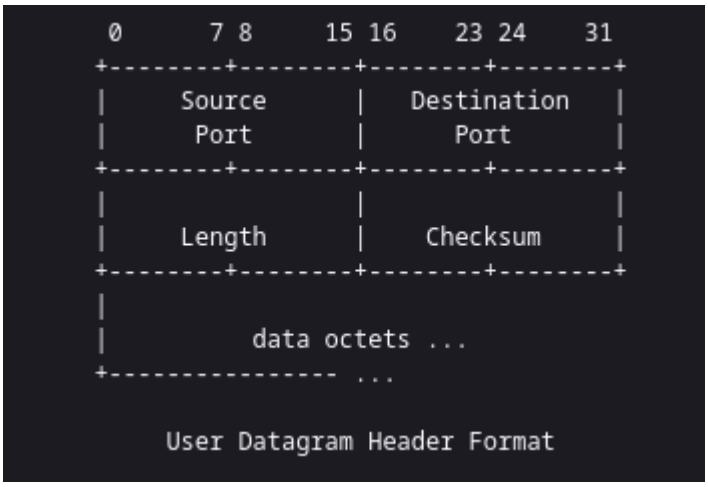
Version část musí mít pro IPv6 hodnotu 6.

Na offsetu á nalezneme zdrojovou adresu a na offsetu 24 cílovou adresu.

Hodnoty jsou uloženy jako Big Endien.

Na offsetu 6 je uveden protokol(typ zprávy), kterou daný paket přenáší. Na obrázku je uveden pod označením Next Header.

## 10.4 UDP protokol [4]




Hodnota pro protokol je 17.

Na transportní vrstvě potřebujeme od UDP pouze zdrojový a cílový port. Zdrojový port nalezneme na začátku části transportní vrstvy paketu a 2 bajty dál nalezneme cílový port. Hodnoty jsou uloženy jako Big Endien.

## 10.4 TCP protokol [5]

Hodnota pro protokol je 6.

Na transportní vrstvě potřebujeme od UDP pouze zdrojový a cílový port. Zdrojový port nalezneme na začátku části transportní vrstvy paketu a 2 bajty dál nalezneme cílový port. Hodnoty jsou uloženy jako Big Endien.

## 10.4 ICMP protokol [6]

Hodnota pro protokol je 1.

Na transportní vrstvě u tohoto protokolu nepotřebujeme nic.

# 11. Literatura

[1] HUAWEI. *Ethernet II Frame*. Online. Dostupné z: https://support.huawei.com/enterprise/en/doc/EDOC1100174721/ea0a043c/ethernet-ii-frame. [cit. 2024-11-16].
[2] DARPA. *RFC 791*. Online. 1981. Dostupné z: https://datatracker.ietf.org/doc/html/rfc791#section-3.1. [cit. 2024-11-16].
[3] DEERING, S. a HINDEN, R. Internet Protocol, Version 6 (IPv6) Specification. Online. 2017. ISSN 2070-1721. Dostupné z: https://datatracker.ietf.org/doc/html/rfc8200. [cit. 2024-11-16].
[4] POSTEL, J. User Datagram Protocol. Online. 1980. Dostupné z: https://www.ietf.org/rfc/rfc768.txt. [cit. 2024-11-16].
[5] TRANSMISSION CONTROL PROTOCOL. Online. 1981. Dostupné z: https://www.ietf.org/rfc/rfc793.txt. [cit. 2024-11-16].
[6] POSTEL, J. INTERNET CONTROL MESSAGE PROTOCOL. Online. 1981. Dostupné z: https://datatracker.ietf.org/doc/html/rfc792. [cit. 2024-11-16].
[7] *Pcap - Packet Capture library*. Online. Dostupné z: https://www.tcpdump.org/manpages/pcap.3pcap.html#lbAS. [cit. 2024-11-17].
[8] *Ncurses - CRT screen handling and optimization package*. Online. Dostupné z: https://man7.org/linux/man-pages/man3/ncurses.3x.html. [cit. 2024-11-17].
[9] *C++ reference*. Online. Dostupné z: https://en.cppreference.com/w/. [cit. 2024-11-17].