VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

ISA-TOP
APLIKACE PRO ZÍSKÁNÍ STATISTIK O SÍŤOVÉM PROVOZU

SEMESTRÁLNÍ PROJEKT - SÍŤOVÉ APLIKACE A SPRÁVA SÍTÍ

AUTOR PRÁCE **JAN PÁNEK**

VEDOUCÍ ZADÁNÍ **Ing. MATĚJ GRÉGR, Ph.D.**

BRNO
Akademický rok 2024/2025

Obsah

Kapitola 1

Popis nástroje isa-top

Nástroj isa-top umožňuje zachytávání síťového provozu na zvoleném rozhraní a zobrazení aktuálních přenosových rychlostí pro jednotlivé detekované toky mezi dvěma koncovými zařízeními. Podporuje monitorování IPv4 i IPv6 komunikace přenášené v rámci ethernetových rámců (EtherType 0x0800 pro IPv4 a 0x86DD pro IPv6). Nástroj je určen pro sledování datových toků TCP/UDP a ICMP/ICMPv6. Pracuje v promiskuitním režimu, což umožňuje zachytávat veškerý síťový provoz na daném síťovém segmentu. Pro výpočet přenosových rychlostí se využívá celková délka IP paketu.

1.1 Konzolové rozhraní

Nástroj je implementován jako konzolová aplikace a zobrazuje deset nejaktivnějších toků dat mezi koncovými zařízeními podle aktuální přenosové rychlosti, přičemž pro každý záznam poskytuje aktuální přenosovou rychlost pro odchozí i příchozí provoz mezi koncovými zařízeními a počty odeslaných i přijatých paketů. Záznamy je možné třídit podle přenosových rychlostí nebo podle počtu přenesených paketů. Statistické údaje jsou ve výchozím nastavení aktualizovány každou sekundu, nicméně lze nastavit i delší celočíselný interval. Přenosové rychlosti jsou zobrazeny v jednotkách bps (bits per second), Kbps, Mbps, Gbps, ... a zaokrouhleny na jedno desetinné místo. Přenosové počty přenesených paketů za sekundu jsou zobrazeny v jednotkách pps (packets per second), Kbps, Mbps, Gbps, ... a zaokrouhleny na jedno desetinné místo.

Konrétní forma výstupu je zobrazena na 1.1. Každý záznam obsahuje identifikaci komunikace (sloupce `Src IP:port` a `Dst IP: port` a protokol `Proto`) a přenosové rychlosti a počty přenesených paketů (`Rx` pro od cíle ke zdroje, `Tx` pro od zdroje k cíli).

Src IP:port	Dst IP:port	Proto	Rx		Tx	
			b/s	p/s	b/s	p/s
[2a00:1450:4014:80b::200e]:443	[2a02:8308:b08d:5100::e3d1]:37712	tcp	73.8K	127.0	2.7M	239.0
[2a02:8308:b08d:5100::e3d1]:55206	[2a00:11c0:4:350::3]:443	tcp	17.7K	16.0	22.6K	19.0
[fe80::b708:cea2:df84:12f2]:5353	[ff02::fb]:5353	udp	0	0	1.3K	2.0
192.168.0.122:5353	224.0.0.251:5353	udp	0	0	984.0	2.0

Obrázek 1.1: Zobrazené přenosové rychlosti a počty přenesených paketů po 1 sekundě.

1.2 Použití aplikace

Aplikace podporuje několik přepínačů, a to pro určení síťového rozhraní, na kterém mají být přenosové rychlosti měřeny, stanovení velikosti periody, se kterou se statistiky aktualizují, a přepínače pro řazení záznamů. Dále poskytuje přepínač pro ukládání statistik do zvoleného adresáře. Synopse je
`isa-top -i int [-s b|p] [-d dir] [-t period]`, kde

- `-i int` je povinný parametr určující rozhraní, na kterém bude provoz sledován,
- `-s b|p` je volitelný parametr určující řazení podle bytů (b), či podle paketů (p),
- `-d dir` je volitelný parametr určující adresář pro textové výstupy,
- `-t period` je volitenlý parametr pro určení periody pro aktualizaci statistik.

Po spuštění jsou se zvolenou periodou zobrazovány aktuální přenosové rychlosti. Záznamy jsou imlicitně řazeny podle počtu přenesených bajtů a perioda pro aktualizaci je nastavena na 1 sekundu. Aplikace je ukončena stisknutím kombinace `CTRL + C`.

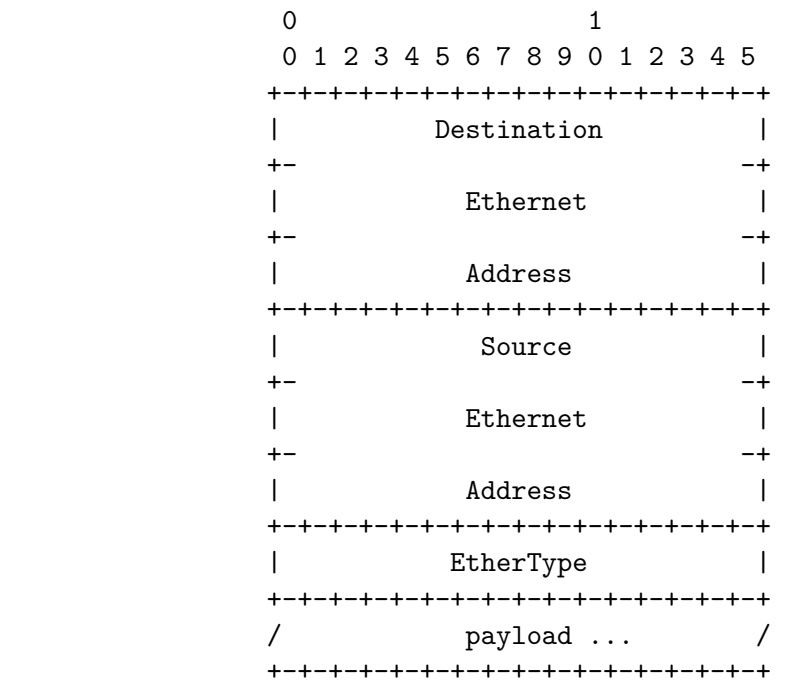
- Spuštění aplikace pro sledování aktuálních přenosových rychlostí na rozhraní eth0 a řezení podle počtu bajtů a aktualizaci po 1 sekundě vypadá následovně:
  `isa-top -i eth0`.
- Spuštění aplikace pro sledování aktuálních přenosových rychlostí na rozhraní wlo1 a řezení podle počtu přenesených paketů a aktualizaci po 2 sekundách vypadá následovně:
  `isa-top -i wlo1 -s p -t 2`.

1.3 Teoretický základ

Klíčovou částí při implementaci nástroje je identifikace jednotlivých toků dat (flows). Tok je definován jako soubor paketů nebo rámců procházejících pozorovacím bodem v síti během určitého časového intervalu. Všechny pakety patřící do určitého toku mají řadu společných vlastností. Klíčem toku (flow key) je potom tradičně pětice (zdrojová ip adresa, zdrojový port, cílová ip adresa, cílový port, protokol transportní vrstvy). Pro protokoly síťové vrstvy je pro identifikaci použita pouze trojice (zdrojová ip adresa, cílová ip adresa, protokol síťové vrstvy). Záznam o toku obsahuje informace o konkrétním toku, který byl pozorován v pozorovacím bodě. V tomto případě nástroje isa-top zaznamenává celkový počet bajtů pro všechny pakety toku a počet paketů toku za danou periodu s rozlišením odesílatele a příjemce, a to společně s identifikací toku. [6, Section 2].

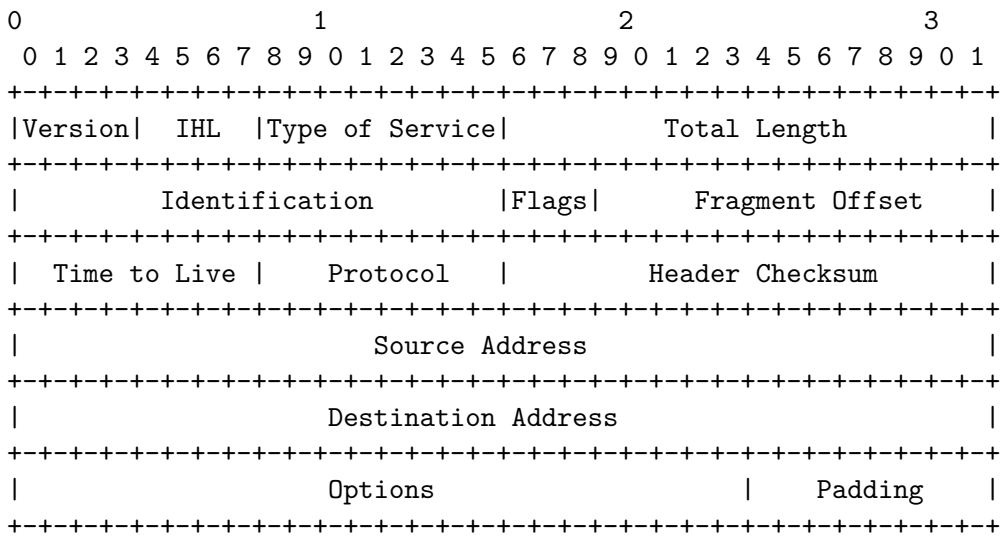
K dosažení údajů pro identifikaci toku a získání informací o něm je nutné u zachyceného paketu postupovat od hlavičky ethernetového rámce k IP či IPv6 datagramu a následně k zprávě ICMP nebo hlavičce paketu TCP,UDP.

V rámci nástroje isa-top nás zajímají pouze data nesená v ethernetovém rámci 1.2 s EtherType 0x0800 pro IPv4 a 0x86DD pro IPv6. IP datagramy jsou základní jednotka přenosu dat v síti, která obsahuje informace o zdrojové a cílové adrese a přenáší data mezi zařízeními na síťové vrstvě. Velikost hlavičky rámce je 14 bytů a následují data rámce (zde IP datagram). [5] [7]



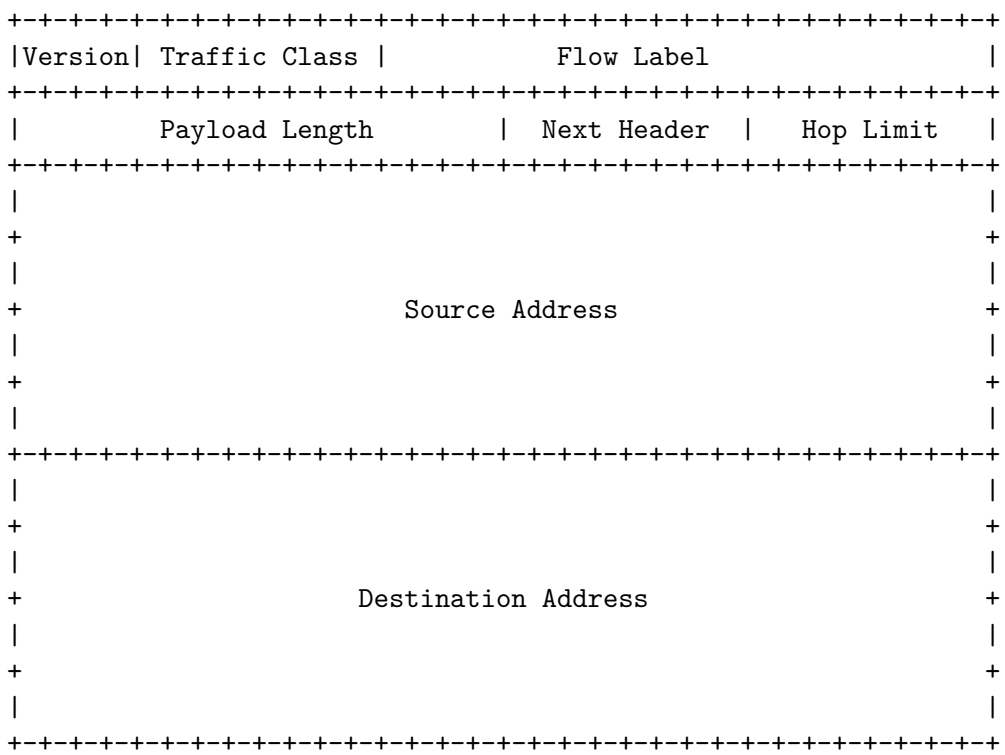
Obrázek 1.2: Podoba ethernetového rámce. [7]

Při zpracování hlavičky protokolu IPv4 1.3 jsou středem zájmu pole zdrojová adresa, cílová adresa, protokol a celková velikost, která označuje celkovou velikost IP datagramu včetně hlavičky. Tato hodnota je použita pro výpočet přenosových rychlostí. Hlavička IPv4 má základní délku 20 bytů bez uvažování voleb. Pro dosažení začátku nesených dat je nutné využít obsah pole velikost hlavičky, která udává veliksot hlavičky v 32 bitových slovech. Typ nesených dat je udán polem protokol. [3]



Obrázek 1.3: Podoba hlavičky IP datagramu. [3]

Při zpracování hlavičky protokolu IPv6 1.4 jsou středem zájmu pole zdrojová adresa, cílová adresa, další hlavička a délka dat, která obsahuje délku dat nesených v datagramu. V rámci aplikace je pro výpočet přenosových rychlostí využita celková veliksot IP datagramu, tedy pro IPv6 je potřeba ještě přičíst velikost hlavičky 40 bytů. Typ nesených dat je udán polem další hlavička. [9]



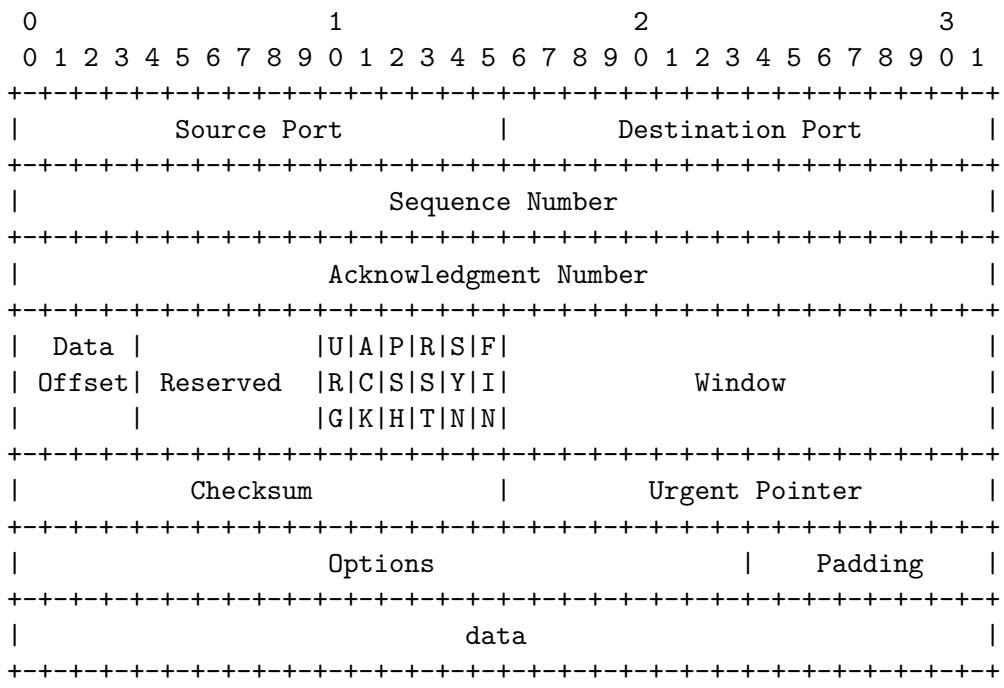
Obrázek 1.4: Podoba hlavičky ipv6 datagramu. [9]

V rámci nástroje isa-top jsou zpracovávány pouze IP datagramy obsahující TCP paket (protokol/další hlavička = 0x06), UDP paket (protokol/další hlavička = 0x11), ICMP zprávu (protokol = 0x01), ICMP6 zprávu (další hlavička = 0x3A) [10].

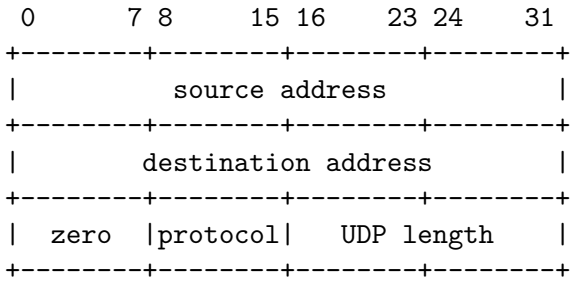
Obsah zpráv ICMP již není z hlediska identifikace toků dat relevantní. Tedy není třeba jej více rozebírat. ICMP (Internet Control Message Protocol) a ICMPv6 slouží k diagnostice a správě síťového provozu tím, že poskytují zpětnou vazbu o stavu síťového spojení, chybách v přenosu a dalších síťových problémech. ICMP je používán v IPv4, zatímco ICMPv6 je jeho ekvivalent pro IPv6, s rozšířenými funkcemi pro správu sousedství a detekci problémů v IPv6 síti. [2] [8]

TCP 1.5 a UDP 1.6 jsou oba transportní protokoly, které slouží k přenosu dat mezi aplikacemi na různých zařízeních v síti. Z hlediska indetifikace spojení jsou podstatná pole zdrojový port a cílový port, které identifikují procesy na koncových zařízeních, se kterými je tok dat spjat. [4] [1]

Aktuální přenosová rychlosti DTR$[b \cdot s^{-1}]$ pro daný tok dat při přenosu Bytes$[B]$ bytů za dobu $T[s]$ je potom dána vztahem $\mathrm{DTR} = \frac{\mathrm{Bytes} \cdot 8}{T}$.



Obrázek 1.5: Podoba TCP paketu. [4]



Obrázek 1.6: Podoba UDP paketu. [1]

Kapitola 2

Technické řešení a testování

Nástroj je implementován v jazyce C++ (verze C++11 a výše). Pro zachytávání a anaylýzu síťového provozu je využita knihovna libpcap [1] (verze 1.10.5-2). Terminálové rozhraní je tvořeno s využitím knihovny ncurses [2] (verze 6.5-3).

2.1 Návrh aplikace

Architektura aplikace odpovídá architektonickému vzoru Model-View-Controller. View zahrnuje zobrazení statistik, které jsou formátovány do tabulky a ta je zobrazena v konzoli. Této části odpovídá soubor `ncurses_terminal_view.cpp`. Controller je reprezentován funkcí `main` v souboru `main.cpp` a odpovídá za zpracování uživatelského vstupu (`CTRL + C`) a aktualizaci view jednou za stanovenou periodu. Po zpracování konfigurace a iniciualizace view a modelu je ve funkci `main` spuštěno monitorování komunikací na separátním vlákně. Až do ukončení aplikace jsou po uplynutí periody pro aktualizaci v cyklu získány naměřené statistiky z modelu a následně je aktualizováno view. Tato logika je zaznemenána v 2.1.

```
// Separate thread for capturing
std::thread monitor_thread(&FlowMonitor::start, &monitor);
while (running)
{
    view_data = monitor.getData();  // get data and clear statistics
    updateView(view_data);
    std::this_thread::sleep_for(std::chrono::seconds(refresh_period));
}
```

Obrázek 2.1: Zobrazené přenosové rychlosti a počty přenesených paketů po 1 sekundě.

Klíčovou roli potom plní model, který je reprezentován třídou `FlowMonitor`. Ten dále zavisí na třídě `FlowTable`, která zapouzdřuje dvě datové struktury. První je hashovací tabulka určená k uložení a rychlemů přístupu k již zaznamenaným statistikám k danému toku dat. Exaktní podoba páru (klíč, hodnota) v tabulce je následující:
`({SrcIP, SrcPort, DstIP, DstPort, Protokol},`
`{počet zaslaných bytů z Dst do Src, počet zaslaných paketů z Dst do Src, počet zaslaných bytů z Src do Dst, počet zaslaných paketů z Src do Dst} ).`

[1] `https://www.tcpdump.org/`
[2] `https://invisible-island.net/ncurses/ncurses.html`

Druhou je seřazený seznam deseti komunikací s nejvyšším počtem přenesených bajtů či paketů (v závisloti na konfiguraci). Třída `FlowTable` potom poskytuje rozhraní pro přidání, či aktualizaci tabulky při automatické aktualizaci seznamu. Přístup k datovým strukturám je patřičně ošetřen, aby bylo možné instanci třídy používat napříč více vláknky.

Třída `FlowMonitor` potom poskytuje rozhraní pro zahájení monitorování v rámci kterého je aktualizována instance třídy `FlowTable` a pro získání seznamu deseti nejakativnějších komunikací. Po získání dat je obsah zmíněných datových struktur vymazán a sběr statistik je zahájen znovu. Pro extrahování potřebných dat ze zachyceného provozu využívá třída třída `FlowMonitor` funkce ze souboru `capturing_utils.cpp`.

2.2 Testování

Pro účely testování byly použity existující nástroje Wireshark[3] pro zachycení síťového provozu, tcpreplay[4] pro přehrání síťového provozu z .pcap souboru a nástroje ping [5] a ncat [6] pro snadné generování síťového provozu. Funkčnost implementace byla validována s nástrojem iftop [7]. Dále byl využit skript v jazyce python, který s využitím knihovny pytshark vypočítá sledované statistik pro zachycené komunikace v souboru .pcap. Skripty pro spuštění testů společně se zachycenými komunikacemi jsou přiloženy v adresáři tests.

Způsob testování je popsán v následujících bodech.

1. Bylo vytvořeno několik záznamů síťového provozu pomocí programu Wireshark.
   - TCP pakety přenášené v protokolem IPv4 byly generovány pomocí nástroje ncat.
     Server: `ncat -4 -l 127.0.0.1 4567`
     Klient: `ncat -4 127.0.0.1 4567`
   - TCP pakety přenášené v protokolem IPv6 byly generovány pomocí nástroje ncat.
     Server: `ncat -6 -l ::1 4567`
     Klient: `ncat -6 ::1 4567`
   - UDP pakety přenášené v protokolem IPv4 byly generovány pomocí nástroje ncat.
     Server: `ncat -4 -u -l 127.0.0.1 4567`
     Klient: `ncat -4 -u 127.0.0.1 4567`
   - UDP pakety přenášené v protokolem IPv6 byly generovány pomocí nástroje ncat.
     Server: `ncat -6 -u -l ::1 4567`
     Klient: `ncat -6 -u ::1 4567`
   - ICMP zprávy byly generovány pomocí nástroje ping.
     `ping -4 127.0.0.1`
   - ICMPv6 zprávy byly generovány pomocí nástroje ping.
     `ping -6 ::1`
   - Náhodn7 reálný provoz byl zachycen na 2 síťových rozraních: `eth0`, `wlo1`.
2. .pcap soubory byly zpracovány skriptem v pythonu pro zjištění požadovaných statistik.
   `python capture_test.py pcap period`

[3] `https://www.wireshark.org/`
[4] `https://tcpreplay.appneta.com/`
[5] `https://github.com/iputils/iputils`
[6] `https://nmap.org/ncat/`
[7] `https://pdw.ex-parrot.com/iftop/`

3. Zaznamenané komunikace v .pcap byly přehrány pomocí nástroje tcpreplay, přičemž byl spuštěn nástroj isa-top s periodou aktualizace 1 sekunda a zapnutým textovým výstupem.
   ```
   ./isa-top -i lo -s b -t 1 -d .
   tcpreplay -intf1=lo
   ```

4. Zaznamenané komunikace v .pcap byly přehrány pomocí nástroje tcpreplay, přičemž byl spuštěn nástroj isa-top a současně nástroj iftop, oba s periodou aktualizace 2 sekundy a zapnutým textovým výstupem.
   ```
   ./isa-top -i lo -s b -t 2 -d .
   iftop -i lo -n -N -p -P -o 2s -t
   tcpreplay -intf1=lo
   ```

Vzhledem k tomu, že nejde zajistit exaktní synchronizaci jednotlivých procesů, tj. aby přehrání provozu a jeho analýza pomocí isa-top a iftop začala ve stejný časový okamžik, byl u výsledků porovnáván vývoj aktuálních přenosových rychlostí a počtu paketů za sekundu. Dále bylo kontorlováno, zda byly správně zachyceny všechny komunikace včetně správných hodnot jednotlivých údajů záznamů (adresy, porty, protokol). Dále bylo kontrolováno, zda sedí celkový přenesený počet paketů. Příklady jendotlivých výstupů, které byly manuálně porovnány, jsou 2.2, 2.3 a 2.4.

Funkčnost přepínačů pro řazení byla otestována kontrolou pořadí záznamů v zobrazené tabulce.

Src IP:port	Dst IP:port	Proto	Rx		Tx	
			b/s	p/s	b/s	p/s
192.168.0.129:42970	34.107.221.82:80	tcp	2.6K	2.0	3.8K	3.0
192.168.0.199:46220	142.251.36.78:443	tcp	988.0	2.0	1.7K	3.5
192.168.0.199:53594	142.251.36.106:443	tcp	1.4K	3.0	1.7K	3.5
192.168.0.199:54006	142.251.36.131:443	tcp	1.2K	2.5	1.6K	3.0
192.168.0.199:60836	142.251.36.106:443	tcp	1.2K	2.5	1.6K	3.0
192.168.0.199:54014	142.251.36.150:443	tcp	1.2K	2.5	1.6K	3.0
192.168.0.199:57314	142.251.36.118:443	tcp	1.2K	2.5	1.6K	3.0
140.82.112.25:443	192.168.0.199:47084	tcp	1.1K	2.0	1.0K	2.0
192.168.0.199:39114	34.107.243.93:443	tcp	528.0	1.0	544.0	1.0
127.0.0.1:34066	127.0.0.1:44605	tcp	416.0	1.0	416.0	1.0

Obrázek 2.2: Statistiky pro interval od 0 po 2 sekundu náhodně zachyceného provozu na rozhraní `wlo1` naměřené pomocí nástroje isa-top.

```
  1 192.168.0.129:42970                     =>     3,73Kb     3,73Kb     3,73Kb      956B
    34.107.221.82:80                        <=     2,53Kb     2,53Kb     2,53Kb      648B
  2 192.168.0.199:53594                     =>     1,68Kb     1,68Kb     1,68Kb      430B
    142.251.36.106:443                      <=     1,37Kb     1,37Kb     1,37Kb      351B
  3 192.168.0.199:54014                     =>     1,52Kb     1,52Kb     1,52Kb      390B
    142.251.36.150:443                      <=     1,17Kb     1,17Kb     1,17Kb      299B
  4 192.168.0.199:60836                     =>     1,52Kb     1,52Kb     1,52Kb      390B
    142.251.36.106:443                      <=     1,17Kb     1,17Kb     1,17Kb      299B
  5 192.168.0.199:57314                     =>     1,52Kb     1,52Kb     1,52Kb      390B
    142.251.36.118:443                      <=     1,17Kb     1,17Kb     1,17Kb      299B
  6 192.168.0.199:54006                     =>     1,52Kb     1,52Kb     1,52Kb      390B
    142.251.36.131:443                      <=     1,17Kb     1,17Kb     1,17Kb      299B
  7 192.168.0.199:46220                     =>     1,68Kb     1,68Kb     1,68Kb      430B
    142.251.36.78:443                       <=      988b       988b       988b      247B
  8 192.168.0.199:47084                     =>     1,05Kb     1,05Kb     1,05Kb      269B
    140.82.112.25:443                       <=     1,02Kb     1,02Kb     1,02Kb      261B
  9 192.168.0.199:39114                     =>      544b       544b       544b      136B
    34.107.243.93:443                       <=      528b       528b       528b      132B
 10 127.0.0.1:34066                         =>      208b       208b       208b       52B
    127.0.0.1:44605                         <=        0b         0b         0b        0B
```

Obrázek 2.3: Statistiky pro interval od 0 po 2 sekundu náhodně zachyceného provozu na rozhraní `wlo1` naměřené pomocí nástroje iftop.

```
[140.82.112.25]:443 [192.168.0.199]:47084 TCP rbps='1.1K' tbps='1.0K' rpps='2.0' tpbs='0.5'
[192.168.0.199]:46220 [142.251.36.78]:443 TCP rbps='988.0' tbps='1.7K' rpps='2.0' tpbs='0.5'
[192.168.0.199]:54006 [142.251.36.131]:443 TCP rbps='1.2K' tbps='1.6K' rpps='2.5' tpbs='0.5'
[192.168.0.199]:54014 [142.251.36.150]:443 TCP rbps='1.2K' tbps='1.6K' rpps='2.5' tpbs='0.5'
[192.168.0.199]:60836 [142.251.36.106]:443 TCP rbps='1.2K' tbps='1.6K' rpps='2.5' tpbs='0.5'
[192.168.0.199]:53594 [142.251.36.106]:443 TCP rbps='1.4K' tbps='1.7K' rpps='3.0' tpbs='0.5'
[192.168.0.199]:57314 [142.251.36.118]:443 TCP rbps='1.2K' tbps='1.6K' rpps='2.5' tpbs='0.5'
[192.168.0.129]:42970 [34.107.221.82]:80 TCP rbps='2.6K' tbps='3.8K' rpps='2.0' tpbs='0.5'
[192.168.0.199]:39114 [34.107.243.93]:443 TCP rbps='528.0' tbps='544.0' rpps='1.0' tpbs='0.5'
[fe80::7a6a:1fff:fe3e:c64b]:0 [ff02::1]:0 ICMPv6 rbps='0.0' tbps='736.0' rpps='0.0' tpbs='0.5'
```

Obrázek 2.4: Statistiky pro interval od 0 po 2 sekundu náhodně zachyceného provozu na rozhraní `wlo1` získané python skriptem z .pcap.

Literatura

[1] *User Datagram Protocol* RFC 768. RFC Editor, srpen 1980. Dostupné z: https://doi.org/10.17487/RFC0768. Accessed: 2024-11-10.

[2] *Internet Control Message Protocol* RFC 792. RFC Editor, září 1981. Dostupné z: https://doi.org/10.17487/RFC0792. Accessed: 2024-11-10.

[3] *Internet Protocol* RFC 791. RFC Editor, září 1981. Dostupné z: https://doi.org/10.17487/RFC0791. Accessed: 2024-11-10.

[4] *Transmission Control Protocol* RFC 793. RFC Editor, září 1981. Dostupné z: https://doi.org/10.17487/RFC0793. Accessed: 2024-11-10.

[5] *A Standard for the Transmission of IP Datagrams over Ethernet Networks* RFC 894. RFC Editor, 1. duben 1984. Dostupné z: https://doi.org/10.17487/RFC0894. Accessed: 2024-11-10.

[6] Aitken, P.; Claise, B. a Trammell, B. *Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information* RFC 7011. RFC Editor, září 2013. Dostupné z: https://doi.org/10.17487/RFC7011. Accessed: 2024-11-10.

[7] Crawford, D. M. a Hinden, B. *Transmission of IPv6 Packets over Ethernet Networks*. Internet-Draft draft-hinden-6man-rfc2464bis-02. Internet Engineering Task Force, březen 2017. Dostupné z: https://datatracker.ietf.org/doc/draft-hinden-6man-rfc2464bis/02/. Work in Progress.

[8] Gupta, M. a Conta, A. *Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification* RFC 4443. RFC Editor, březen 2006. Dostupné z: https://doi.org/10.17487/RFC4443. Accessed: 2024-11-10.

[9] Hinden, B. a Deering, D. S. E. *Internet Protocol, Version 6 (IPv6) Specification* RFC 2460. RFC Editor, prosinec 1998. Dostupné z: https://doi.org/10.17487/RFC2460. Accessed: 2024-11-10.

[10] Reynolds, J. K. a Postel, D. J. *Assigned Numbers* RFC 1700. RFC Editor, říjen 1994. Dostupné z: https://doi.org/10.17487/RFC1700. Accessed: 2024-11-10.