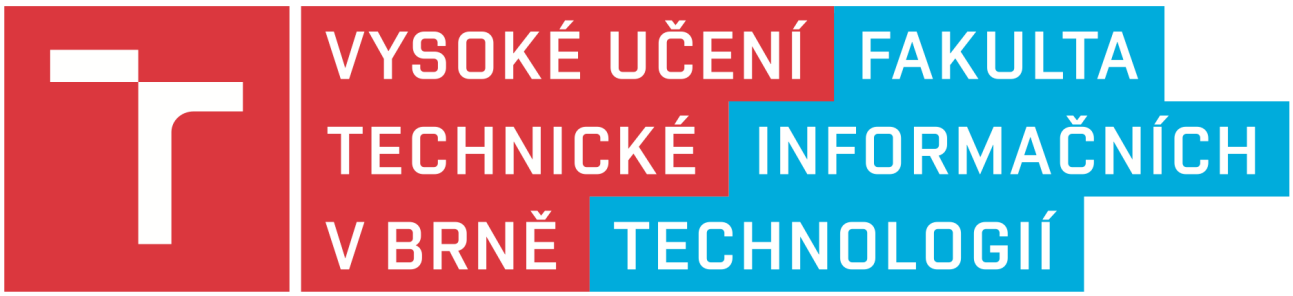


Technická správa

Aplikace pro získání statistik o síťovém provozu

14. novembra 2024

Ján Findra, xfindr01

Obsah

1 Úvod

V tomto projekte som implementoval program `isa-top`, ktorý zobrazuje 10 najviac komunikujúcich spojení na vybranom sieťovom rozhraní. Cieľom programu je analyzovať sieťovú komunikáciu v reálnom čase a zobrazovať štatistiky o rýchlosti prenosu dát v bitoch alebo paketoch za sekundu. Program je vytvorený pre operačný systém Linux, testovanie je ale špecifické pre Ubuntu.

2 Návod na použitie

2.1 Závisloti

Pre preklad sú potrebné knižnice pre preklad c++, pri bežných distribúciach sú tieto knižnice už nainštalované. Ďalej je potreba stiahnuť knižnicu *libpcap* a knižnicu *libncurses*.

Príkazy pre inštaláciu na Ubuntu 22.04 (pre rôzne verzie sa môžu líšiť):

```
sudo apt update
sudo apt upgrade
sudo apt install libpcap-dev
sudo apt install libncurses5-dev
```

2.2 Príprava

Na preklad programu je vytvorený `Makefile`, ktorý zabezpečuje správne preloženie projektu. Program je možné preložiť jednoducho zadaním príkazu `make` v príkazovom riadku.

2.3 Spustenie

Po úspešnom preložení vznikne spustiteľný program `isa-top`. Popis spustenia programu je nasledovný: **isa-top -t interface [-s b/p] [-t period]**

- Argument `interface` určuje sieťové rozhranie, na ktorom bude `isa-top` odpočúvať komunikáciu. Tento argument je povinný.
- Argument `-s b/p` určuje kritérium, podľa ktorého bude zoznam zoradený. Pri zadaní `b` bude zoznam zoradený zostupne podľa rýchlosti v *bitoch za sekundu*. Pri zadaní `p` bude zoznam zoradený zostupne podľa rýchlosti v *paketoch za sekundu*. Tento argument je nepovinný a v prípade jeho neuvedenia sa použije predvolené radenie podľa bitov za sekundu.
- Argument `-t period` definuje periódu v sekundách, za ktorú sa počíta rýchlosť prenosu dát. Perióda určuje časový interval, medzi ktorého obnovami sa zbierajú štatistiky. Ak tento argument nie je zadaný, použije sa predvolená hodnota 1 sekunda.

2.4 Čistenie

Pre vyčistenie preložených súborov a ostatných generovaných súborov je možné použiť príkaz `make clean`.

2.5 Testovanie

Spustenie testov na kontrolu spracovania argumentov je príkaz `make test`. Po spustení môže byť požadované heslo pre super používateľa.

2.6 Zabalenie

Príkaz `make pack` slúži na zabalenie obsahu projektu do archívu typu `.tar`. Tento príkaz vyčistí priečinok, obsah zabalí a pripravený archív uloží, čo umožňuje jeho jednoduché odovzdanie.

3 Návrh aplikácie

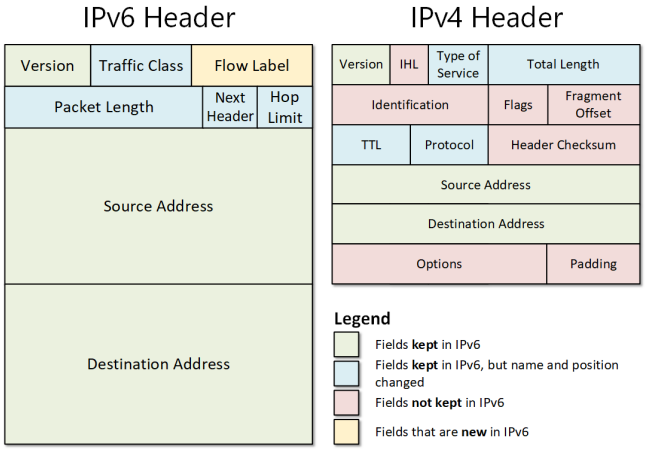
Program načúva komunikáciu na druhej vrstve referenčného modelu ISO/OSI[3]. Tento model popisuje všeobecné princípy sedemvrstvovej sieťovej architektúry.

3.1 Linková vrstva (L2)

Linková vrstva pridáva fyzickú MAC adresu k údajom prenášaným od vyššej vrstvy rámca ISO/OSI a predáva ju nižšej, fyzickej vrstve[2]. Vďaka tejto štruktúre môžeme zo zachytených dát získať informácie aj z vyšších vrstiev, najmä zo sieťovej vrstvy (L3). Z hlavičky každého zachyteného paketu dokážeme zistiť veľkosť prenesených dát v bytoch.

3.2 Sieťová vrstva (L3)

Sieťová vrstva využíva internetový protokol (IP)[4], ktorý existuje vo verziách IPv4 a IPv6. V mojom riešení tiež rozlišujem typ ARP, ktorý identifikuje knižnica ako ďalšiu možnosť. Pre toto riešenie je dôležitý len protokol IPv4, keďže IP hlavičky obsahujú údaje potrebné na spracovanie v našom programe. Na obrázku 1 je znázornený rozdiel medzi hlavičkami protokolov IPv4 a IPv6.



Obr. 1: Porovnanie IPv4 a IPv6 hlavičky[1].

Z IP hlavičky dokážeme získať zdrojovú a cieľovú IP adresu každého prenášaného paketu.

3.3 Transportná vrstva (L4)

Transportná vrstva určuje protokol, ktorý bude použitý na prenos dát[5]. V tomto riešení rozlišujem protokoly ICMP, ICMPv6, IGMP, TCP, UDP a kategóriu `other`, ktorá obsahuje ostatné protokoly. Ak ide o TCP alebo UDP protokol, z hlavičky dokážeme extrahovať čísla zdrojového a cieľového portu. Pri ostatných protokoloch tieto čísla portov nie sú dostupné.

3.4 Zhrnutie

Komunikáciu zachytávame na linkovej vrstve (L2), pričom vďaka štruktúre modelu ISO/OSI máme prístup k informáciám aj z vyšších vrstiev. Takto dokážeme získať všetky údaje potrebné pre fungovanie programu z rôznych vrstiev ISO/OSI modelu.

4 Implementácia

4.1 Knižnice

- `stdio.h` - základná knižnica pre vstup a výstup
- `iostream` - vstup a výstup pre C++
- `string` - manipulácia s reťazcami
- `vector` - práca s dynamickými poliami (vektory)
- `map` - implementácia mapy na nahradenie hashtable
- `thread` - manipulácia s vláknami
- `pcap.h` - knižnica na zachytávanie komunikácie
- `ctime` - práca s časovými údajmi
- `netinet/ip.h|ip6.h|tcp.h|udp.h` - knižnice definujúce štruktúry IP, IP6, TCP a UDP
- `cctype` - kontrola dátových typov
- `ncurses.h` - zobrazenie údajov v termináli

4.2 Prehľad organizácie kódu

```
.
├── LICENSE
├── Makefile
├── args.cpp ..... spracovanie vstupných argumentov
├── args.h
├── isa-top.1 ..... manuálová stránka
├── main.cpp ..... hlavná časť programu
├── main.h
├── messages.cpp ..... pomocné spracovanie chýb
├── messages.h
├── packetHandler.cpp ..... spracovanie zachytených paketov
├── packetHandler.h
├── screen.cpp ..... zobrazenie údajov
├── screen.h
└── test
    ├── ipClient.py ..... klienti pre rôzne protokoly
    ├── ipServer.py ..... servery pre rôzne protokoly
    └── test.sh ..... testy zadávania argumentov
```

4.3 Moja implementácia

Pre ukončenie programu je nastavená klávesová skratka `Ctrl + C`, ktorá spúšťa funkciu `signalHandler`. Táto funkcia signál zachytí a ukončí program korektným spôsobom.

Spracovanie vstupných argumentov zabezpečuje funkcia `parseArgs`, ktorá vracia štruktúru s načítanými hodnotami, prípadne predvolenými hodnotami pri nezadaných voliteľných argumentoch.

Na spustenie zachytávania komunikácie sú použité funkcie z knižnice `pcap.h`. Funkcia `packetHandler` spracováva jednotlivé zachytené pakety, pričom z hlavičky paketu získava potrebné údaje. Tieto údaje sú uložené v pomocnej štruktúre `packetData`. Záznamy o komunikácii sú uložené v globálnej premennej `connections`, ktorá je typu `map`. Operácie nad touto mapou vykonávajú funkcie `addConnection`, `newConnection` a `sortConnections`. Funkcia `addConnection` je volaná z `packetHandler` a stará sa o úpravu záznamov v `connections`. Každé spojenie je identifikované dvojicou kľúč - hodnota, kde kľúč obsahuje kombináciu zdrojovej adresy, zdrojového portu, cieľovej adresy a cieľového portu, a hodnota obsahuje štruktúru `connection` s údajmi pre zobrazenie v termináli.

Funkcia `screenHandler` beží na samostatnom vlákne a spravuje zobrazovanie údajov v termináli. Volá funkciu `sortConnections`, ktorá aktualizuje rýchlosti, odstraňuje staré záznamy a vracia zoradený zoznam top 10 komunikácií. Pre zobrazenie je použitá knižnica `ncurses` s doplnkovými pomocnými funkciami.

4.4 Špecifiká

Rýchlosť prenosu dát je počítaná len z komunikácií, ktoré sa odohrali počas zvolenej periódy. Tento proces môžeme prirovnať k časovému oknu o veľkosti periódy (určenej argumentom `-t`), v ktorom zachytávame komunikácie. Ak je napríklad perióda nastavená na dve sekundy, rýchlosť v bitoch za sekundu bude rovná veľkosti prenesených dát za túto periódu delenou dvomi.

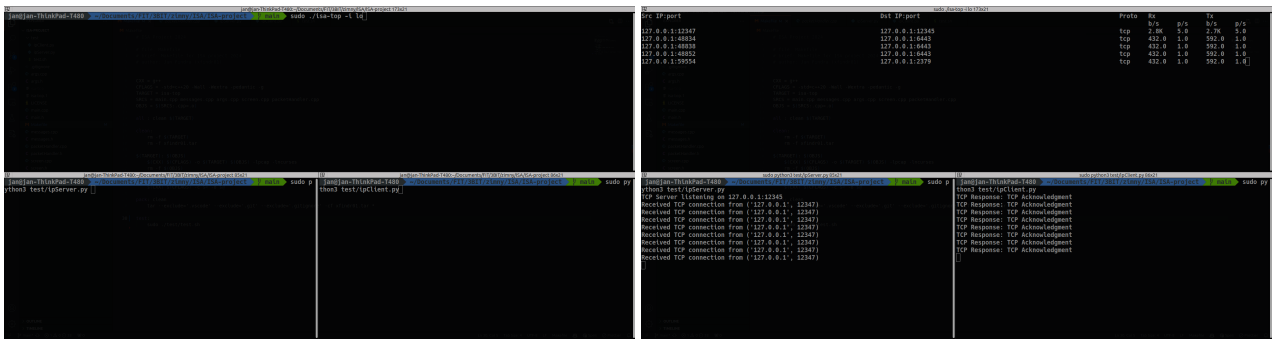
5 Testovanie

Testovanie prebiehalo na *MacOS* (v limitovanej podobe) a hlavne na *Ubuntu 24.04 LTS* a vo virtálnom prostredí *Ubuntu 22.04 LTS*

5.1 Experimentovanie

Väčšina overovania správnej funkčnosti programu prebiehala expertovaním, keďže povaha programu do značnej miery obmedzuje možnosti automatizovaného testovania. Pre porovnanie výstupu som používal program *iftop*, spúčťaný s argumentami `-i interface -n -P -N -l`. Využíval som aj program *Wireshark*, kde som zisťoval veľkosti paketov, adresy, ... Pre umelé zasielanie paketov typov *icmp* a *icmp6* som využil príkaz `curl`.

Pre zjednodušenie a istejšie testovanie som implementoval jednoduchých klientov a servery v jazyku Python (súbory `ipServer.py` a `ipClient.py`. Implementoval som vytváranie komunikácie pomocou protokolov TCP, UDP, ICMP a ICMPv6. Funkčnosť a spoľahlivosť týchto skriptov je nedostatočná a som si vedomý chýb, ktoré pri ich používaní môžu nastať. Ich úlohou je vytvoriť umelý prenos, ktorý môžem sledovať v prehľade komunikácií. Zachytené komunikácie sa nachádzajú v rozhraní *localhost (lo)*. Vytvorenie automatizovaných testov je skomplikované faktom, že na tomto rozhraní prebieha aj iná komunikácia, ktorú nie som schopný vylúčiť. Ide skôr len overenie, že komunikácie, o ktorých viem, že prebiehajú, sa správne zobrazujú vo výstupe testovaného programu *isa-top*.

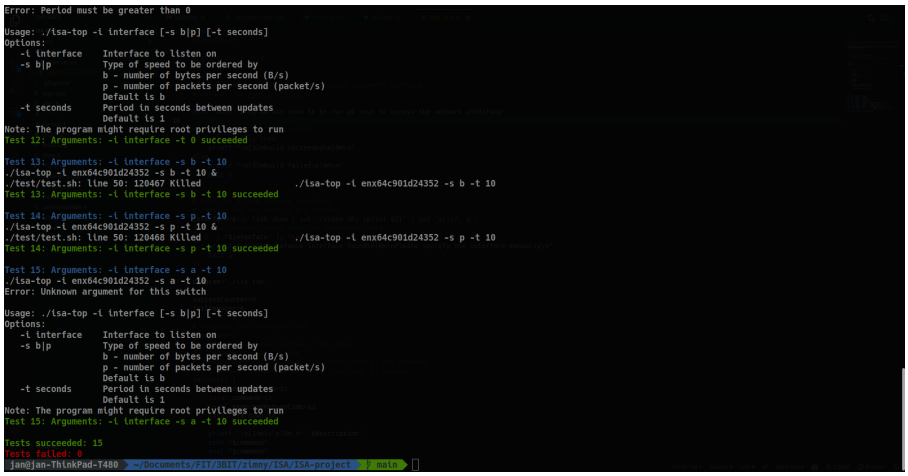


Obr. 2: Ukážka použitia severov a klientov na simuláciu komunikácie a ich zachytávanie pomocou *isa-top* programu

5.2 Automatizované testy

Pre testovanie kontroly a spracovania argumentov je vytvorený shell skrip s názvom `test.sh`. Tento test je do implementovaný pre operačný systém Ubuntu, ale jeho úprava aj pre iné distribúcie Linuxu nie je náročná. Skript overuje správnu reakciu programu na rôzne možnosti zadaných argumentov. Nie sú pokryté všetky možnosti, ale snažil som sa pokryť čo najviac rôznych možností podľa typového rozdelenia možností. Pod typovým rozdelím uvažujem zoskupenia možností podľa základnej črty argumentov alebo ich zoradenia. Napr. zadávanie zlých argumentov, zadávanie v rôznom poradí, zadavaní dvojíc, trojíc a pod.

Pre zavolanie testu je možné použiť `make test` alebo v koreňovom adresári zavolať `sudo ./test/test.sh`.



Obr. 3: Ukážka použitia automatizovaných testov

Literatúra

[1] NetworkAcademy.io: IPv4 vs IPv6 - Understanding the differences. online, 2024. URL `https://www.networkacademy.io/ccna/ipv6/ipv4-vs-ipv6`

[2] Wikipedie: Linková vrstva — Wikipedie: Otevřená encyklopedie. 2023, [Online; navštíveno 10. 11. 2024]. URL `https://cs.wikipedia.org/w/index.php?title=Linkov%C3%A1_vrstva&oldid=23037405`

[3] Wikipedie: Referenční model ISO/OSI — Wikipedie: Otevřená encyklopedie. 2023, [Online; navštíveno 10. 11. 2024].

URL https://cs.wikipedia.org/w/index.php?title=Referen%C4%8Dn%C3%AD_model_ISO/OSI&oldid=23031181

[4] Wikipedie: Síťová vrstva — Wikipedie: Otevřená encyklopedie. 2023, [Online; navštíveno 10. 11. 2024].
URL https://cs.wikipedia.org/w/index.php?title=S%C3%AD%C5%A5ov%C3%A1_vrstva&oldid=23327028

[5] Wikipedie: Transportní vrstva — Wikipedie: Otevřená encyklopedie. 2024, [Online; navštíveno 10. 11. 2024].
URL https://cs.wikipedia.org/w/index.php?title=Transportn%C3%AD_vrstva&oldid=23998412