VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

APLIKÁCIA PRE ZÍSKANIE ŠTATISTÍK O SIEŤOVEJ PREVÁDZKE

AUTOR PRÁCE **AUREL STRIGÁČ**

BRNO 2024

Obsah

Kapitola 1

Úvod do problematiky

V súčasnosti existuje množstvo nástrojov na analýzu sieťovej prevádzky na našich zariadeniach. Príkladom je program `iftop`[10], ktorý umožňuje sledovať aktuálne sieťové toky. Monitorovanie siete je pre užívateľov zaujímavé, pretože umožňuje potencionálne identifikovať zdroje vysokého sieťového zaťaženia, poprípade sledovať neobvyklú sieťovú aktivitu, ktorá môže naznačovať bezpečnostné hrozby alebo aj optimalizovať výkon siete vďaka poskytnutiu údajov potrebných na efektívnu správu a konfiguráciu sieťových nastavení. Cieľom mojeho programu `isa-top` je poskytunúť podobnú funkcionalitu ako práve, vyššie spomenutý, program iftop.

Pre efektívne monitorovanie a analýzu sieťovej prevádzky je kľúčové správne získanie IP adries, čísla portov, transportných protokolov, počtu a veľkosti paketov. Pre neznalého užívateľa tieto pojmy avšak nemusia nič znamenať, a preto by som si ich dovolil v krátkosti vysvetliť:

- **IP adresa**[12] je jedinečná adresa priradená zariadeniu v sieti, ktorá umožňuje komunikáciu s inými zariadeniami v sieti. IP adresy umožňujú smerovanie paketov[1] od zdroja k cieľu. IP adresa je dlhá 32 bitov v prípade IPv4, a 128 bitov v prípade IPv6.
- **Port**[2] je číselné označenie koncového bodu kde sa spojenie začína a končí. Umožňuje aplikáciám identifikovať a smerovať dáta na konkrétnu službu alebo aplikáciu v sieti. V rámci jedného zariadenia môže byť otvorených niekoľko portov, každý pre inú aplikáciu alebo službu. Portové čísla sa pohybujú od 0 do 65535.
- **Transportný protokol** určuje spôsob, akým sú dáta prenášané medzi zariadeniami na transportnej vrstve[6]. Najčastejšie používané protokoly sú TCP (Transmission Control Protocol) a UDP (User Datagram Protocol). Pri IPv4 hlavičke paketu vieme tento údaj vyčítať z poľa `Protocol`[7], a pri IPv6 hlavičke tento údaj dokážeme vyčítať `Next Header`[7].
- **Veľkosť prenesených dát** je dôležitá informácia, ktorá udáva, koľko dát bolo prenesených v rámci daného rámca. Pri IPv4 hlavičke paketu vieme tento údaj vyčítať z poľa `Total Length`[7], a pri IPv6 hlavičke paketu tento údaj dokážeme vyčítať `Payload Length`[7]. Avšak ani jedno z týchto polí nepočíta s veľkosťou ethernetového rámca. Jeho veľkosť je 18 bytov, a táto hodnota sa pre výpočet celkovej veľkosti prenesených dát nesmie zabudnúť. Ak avšak chceme získať iba velkosť preneseného paketu, tak túto hodnotu môžeme zanedbať.

Kapitola 2

Implementácia

Implementácia programu isa-top v jazyku C++ využíva knižnicu pcap[9] na zachytávanie sieťových paketov. Program otvára špecifikované sieťové rozhranie a zachytáva všetky prechádzajúce pakety. Pre každý zachytený paket sa analyzujú jeho sieťové a transportné hlavičky[6], čo umožňuje identifikovať zdrojovú a cieľovú IP adresu, porty a používaný protokol.

Na ukladanie a spracovanie informácií o sieťových tokoch program využíva štruktúry `FlowID` a `FlowStats` spolu s kontajnerom std::map[3], ktorý uchováva štatistiky pre každý tok. Porovnávanie tokov je zabezpečené pomocou vlastného komparátora `FlowIDComparator`, ktorý umožňuje triedenie tokov podľa kombinácie protokolu, IP adries a portov.

Výstup programu je zobrazovaný pomocou knižnice `ncurses`[4], ktorá poskytuje dynamické a prehľadné užívateľské rozhranie v termináli. Program umožňuje triediť zobrazené toky podľa počtu prenesených bajtov alebo paketov a nastavovať interval obnovovania štatistík pomocou príkazových argumentov.

Spracovanie príkazových argumentov je realizované pomocou `getopt`[5], čo umožňuje jednoduchú konfiguráciu nástroja podľa potrieb užívateľa. Program tiež obsahuje obsluhu signálov, konkrétne `SIGINT`, čo umožňuje správne ukončenie programu a uvoľnenie zdrojov pri prerušení.

2.1 Popis návrhu

V tejto sekcii je detailne popísaná štruktúra a rozdelenie jednotlivých súborov v programe `isa-top`. Rozdelenie programu do viacerých súborov umožňuje lepšiu modularitu, udržiavateľnosť kódu a aj potencionálnu škálovateľnosť.

2.1.1 isa-top.cpp & isa-top.h

`isa-top.cpp` je hlavný súbor aplikácie `isa-top`. Tento súbor obsahuje funkciu `main`, ktorá sa stará o:

- Spracovanie príkazových argumentov pomocou funkcie `parse_args`.
- Inicializáciu knižnice `pcap`[9] pre zachytávanie sieťových paketov.
- Nastavenie non-blocking režimu pre pcap handle.
- Validáciu podpory Ethernetových hlavičiek na zvolenom sieťovom rozhraní.

- Inicializáciu užívateľského rozhrania pomocou `ncurses`[4].
- Hlavnú slučku programu, ktorá spracováva pakety a obnovuje štatistiky podľa nastaveného intervalu.
- Obsluhu signálu `SIGINT` pre korektné ukončenie programu.

2.1.2 utils.cpp & utils.h

Súbory `utils.cpp` a `utils.h` obsahujú pomocné funkcie a nástroje, ktoré sú využívané v rôznych častiach programu. Medzi hlavné funkcionality patria:

- Formátovanie bitových a paketových rýchlostí (`format_bits`, `format_packets`).
- Validácia a spracovanie príkazových argumentov (`parse_args`, `check_sort_order`, `check_refresh_interval`, `check_interface_set`, `handle_invalid_argument`).
- Zobrazenie pomocnej nápovedy pre užívateľov (`print_help`).
- Kontrola podpory Ethernetových hlavičiek (`check_ethernet_support`).

2.1.3 flow.cpp & flow.h

Súbory `flow.cpp` a `flow.h` spravujú dáta týkajúce sa sieťových tokov. Obsahujú definície a implementácie nasledujúcich štruktúr a funkcií:

- **FlowID**: Štruktúra identifikujúca jedinečný sieťový tok na základe kombinácie IP adries, portov a protokolu.
- **FlowStats**: Štruktúra uchovávajúca štatistiky pre každý tok, (prenesené bajty a pakety).
- **FlowIDComparator**: Komparátor pre triedenie tokov v `std::map`.
- Funkcie na aktualizáciu a spracovanie štatistík tokov (`update_flow_statistics`, `flow_not_active`, `reset_flow_statistics`, `trim_flows`).

2.1.4 capture.cpp & capture.h

Súbory `capture.cpp` a `capture.h` sa starajú o zachytávanie a parsovanie sieťových paketov. Ich hlavné úlohy zahŕňajú:

- Inicializácia a konfigurácia pcap handle na zachytávanie paketov.
- Implementácia funkcie `packet_handler`, ktorá spracováva každý zachytený paket.
- Parsovanie vrstiev L3 (IP) a L4 (TCP/UDP/ICMP) paketov (`parse_L3_ipv4`, `parse_L3_ipv6`, `parse_L4`).
- Aktualizácia štatistík sieťových tokov na základe analyzovaných paketov.

2.1.5 display.cpp & display.h

Súbory `display.cpp` a `display.h` sú zodpovedné za zobrazovanie informácií o sieťovej prevádzke pomocou knižnice `ncurses`[4]. Ich hlavné funkcionality zahŕňajú:

- Zobrazenie hlavičky tabuľky so sieťovými tokmi (`display_header`).
- Formátovanie a výpis štatistík pre jednotlivé toky (`display_flow`).
- Spracovanie a zobrazenie kolekcie sieťových tokov (`display_statistics`).
- Inicializácia a správa užívateľského rozhrania (`display_startup`).

2.2 Návod na použitie

Táto sekcia poskytuje návod na spustenie programu `isa-top`, vrátane popisu dostupných možností parametrov a príkladov ich použitia.

2.2.1 Prvotné spustenie

Pred spustením programu je potrebné nainštalovať všetky potrebné knižnice:

```
sudo apt install build-essential libpcap-dev libncurses5-dev libncursesw5-dev
```

2.2.2 Kompilácia programu

Program obsahuje Makefile, ktorý môžete použiť pre preklad programu.

1. Prejdite do adresára so zdrojovým kódom:

   ```
   cd */isa-top
   ```

2. Preložte program:

   ```
   make
   ```

   Po úspešnej kompilácii by mal byť vytvorený spustiteľný súbor `isa-top`.

3. (Voliteľne) Vymazanie súborov vytvorených pro preklade:

   ```
   make clean
   ```

2.2.3 Spustenie programu

Na spustenie programu je potrebné použiť nasledujúci príkaz:

```
./isa-top -i <interface-id> [-s b|p] [-t <seconds>] [-h|--help]
```

Vysvetlenie parametrov

- `-i` *`interface-id`*: Určuje sieťové rozhranie, na ktorom bude program zachytávať pakety. Tento parameter je ako jediný povinný.
- `-s` *`b/p`*: Určuje spôsob triedenia zobrazovaných sieťových tokov.
  - `b`: Triediť podľa celkového počtu prenesených bitov (predvolené nastavenie).
  - `p`: Triediť podľa celkového počtu prenesených paketov.
- `-t` *`seconds`*: Nastavuje interval obnovovania štatistík v sekundách. Musí byť väčší ako 0. Predvolené nastavenie je 1 sekunda.
- `-h, -help`: Zobrazuje nápovedu s informáciami o použití programu.

2.2.4 Príklady použitia

Nasledujúce príklady demonštrujú príklady použitia programu `isa-top`.

Monitorovanie sieťovej prevádzky na rozhraní `wlan0` so štandardným triedením podľa bajtov a nastaveným intervalom 2 sekundy

```
./isa-top -i wlan0 -t 2
```

Tento príkaz spustí program `isa-top` na sieťovom rozhraní `wlan0`, triediť bude výstup podľa bajtov a interval obnovovania štatistík bude nastavený na 2 sekundy.

Monitorovanie sieťovej prevádzky na rozhraní `enp0s3` so štandardným triedením podľa paketov a nastaveným intervalom 5 sekúnd

```
./isa-top -i enp0s3 -s p -t 5
```

Tento príkaz spustí program `isa-top` na sieťovom rozhraní `enp0s3`, triediť bude výstup podľa počtu prenesených paketov a interval obnovovania štatistík bude nastavený na 5 sekúnd.

Zobrazenie nápovedy programu

```
./isa-top -h
```

alebo

```
./isa-top --help
```

Tieto príkazy zobrazia nápovedu, a následne ukončia beh programu.

Kapitola 3

Testovanie

Program `isa-top` bol testovaný pomocou rôznych scenárov s využitím príkazu `ping`[8] ale aj pomocou porovnania výstupov nášho programu s referenčným programom `iftop`[10]. Na kontrolu správnej reprezentácie veľkosti paketov bol použitý program `Wireshark`[11]. Nasledujúce tri testovacie prípady demonštrujú rôzne prípady fungovania `isa-top`.

3.1 Test 1: Normálny ping

3.1.1 Príkaz ping

```
ping 8.8.8.8
```

3.1.2 Spustenie aplikácie

```
./isa-top -i eth0 -t 10
```

3.1.3 Odôvodnenie testu

Tento test slúži na overenie, či `isa-top` správne zaznamenáva a zobrazuje bežnú sieťovú prevádzku generovanú príkazom `ping`. Normálny ping vysiela ICMP Echo správy, ktoré by mali byť rospoznané a správne zobrazené v aplikácii.

3.1.4 Výsledok

`isa-top` úspešne zaznamenal všetky ICMP Echo správy vyslané príkazom `ping`. Veľkosť paketov za sekundu bola správne reprezentovaná bez započítania veľkosti Ethernetového rámca, čo bolo potvrdené porovnaním s výstupom `Wireshark`. Táto veľkosť mala, podľa našeho programu, hodnotu 672. Zároveň zobrazil správne počet paketov za sekundu, ktorý činil hodnotu 1. Program zobrazil správne aj zdrojové a cieľové IP adresy, porty, a používaný protokol.

3.2 Test 2: Ping so skráteným intervalom

3.2.1 Príkaz ping

```
ping -i 0.2 8.8.8.8
```

3.2.2 Spustenie aplikácie

```
./isa-top -i eth0 -t 10
```

3.2.3 Odôvodnenie testu

Tento test overuje, ako `isa-top` spracováva sieťovú prevádzku s vyšším frekvenciou paketov za jeden interval obnovenia. Skrátený interval vysielania pingov (`-i 0.2`) umožňuje otestovať schopnosť aplikácie aktualizovať štatistiky jednotlivých tokov.

3.2.4 Výsledok

`isa-top` úspešne zaznamenal všetky ICMP Echo správy vyslané príkazom `ping`. Veľkosť paketov za sekundu bola správne reprezentovaná bez započítania veľkosti Ethernetového rámca, čo bolo potvrdené porovnaním s výstupom `Wireshark`. Táto veľkosť mala, podľa našeho programu, hodnotu 3.4K. Toto je 5-násobne vyššia hodnota ako v teste č. 1, čo je na základe intervalu odosielania správne. Zároveň zobrazil správne počet paketov za sekundu, ktorý činil hodnotu 5. Program zobrazil správne aj zdrojové a cieľové IP adresy, porty, a používaný protokol.

3.3 Test 3: Ping na nedostupnú adresu

3.3.1 Príkaz ping

```
ping 192.0.2.1
```

3.3.2 Spustenie aplikácie

```
./isa-top -i eth0
```

3.3.3 Odôvodnenie testu

Tento test slúži na otestovanie, ako `isa-top` reaguje na sieťovú prevádzku smerovanú iba jedným smerom. V takomto prípade by zariadenie ktoré odosiela správy nemalo dostávať žiadne odpovede, takže hodnota všetkých polí Rx by mala byť nulová.

3.3.4 Výsledok

Test prebehol úspešne. Program `isa-top` správne zaznamenal sieťovú prevádzku smerovanú iba jedným smerom. Hodnoty všetkých polí Rx boli nulové. Tento výsledok ukazuje, že `isa-top` efektívne rozlišuje medzi odosielanými a prijatými paketmi.

Literatúra

[1] Cisco. *Introduction to Routing and Packet Forwarding*. Brno, Czech Republic: Masaryk University, 2012. Dostupné z: https://is.muni.cz/el/1433/podzim2012/PV233/um/prednasky/Exploration_Routing_Chapter_1.pdf.

[2] Cloudflare. *What is a Computer Port?* 2024. Dostupné z: https://www.cloudflare.com/learning/network-layer/what-is-a-computer-port/.

[3] cplusplus.com. *Std::map*. 2024. Dostupné z: https://en.cppreference.com/w/cpp/container/map.

[4] Debian. *Ncurses*. 2024. Dostupné z: https://wiki.debian.org/Ncurses.

[5] Kerrisk, M. *Getopt(3) — Linux manual page*. 2024. Dostupné z: https://www.man7.org/linux/man-pages/man3/getopt.3.html.

[6] Kinza Yasar, A. N. *What is TCP/IP?* 2024. Dostupné z: https://www.techtarget.com/searchnetworking/definition/TCP-IP.

[7] NetworkAcademy.io. *IPv4 vs IPv6 - Understanding the differences*. 2024. Dostupné z: https://www.networkacademy.io/ccna/ipv6/ipv4-vs-ipv6.

[8] tie.net. *Ping(8) - Linux man page*. 2024. Dostupné z: https://linux.die.net/man/8/ping.

[9] Tim Carstens, G. H. *Programming with pcap*. 2024. Dostupné z: https://www.tcpdump.org/pcap.html.

[10] Warren, P. *Iftop(8) - Linux man page*. 2024. Dostupné z: https://linux.die.net/man/8/iftop.

[11] Wireshark. 2024. Dostupné z: https://www.wireshark.org/.

[12] Yasar, K. *IP address (Internet Protocol address)*. 2024. Dostupné z: https://www.techtarget.com/whatis/definition/IP-address-Internet-Protocol-Address.