



ISA - Síťové aplikace a správa sítí

Aplikace pro získání statistik o síťovém provozu

Juraj Budai

xbudai02

17.11.2024

Obsah

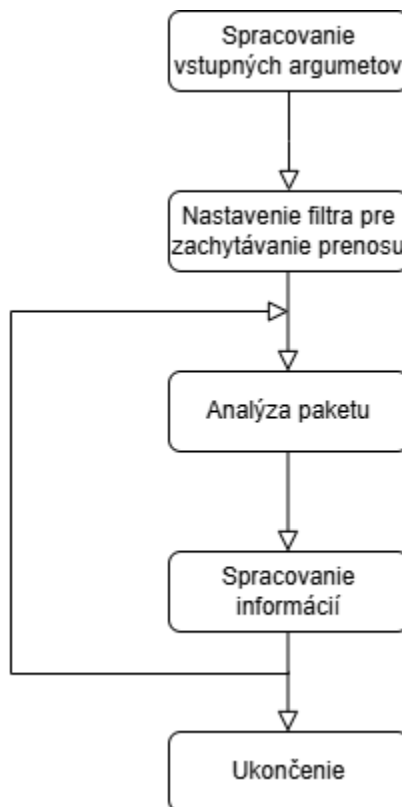
1. Uvedenie do problematiky	3
2. Návrh aplikácie	3
3. Popis implementácie.....	3
4. Informácie o programe	4
a. Účel:	4
b. Použitie	4
c. Výstup.....	4
d. Požiadavky.....	5
e. Obmedzenia	5
5. Popis testovania a výsledky testov.....	5
6. Použitá literatúra	6

1. Uvedenie do problematiky

Cieľom projektu bolo vytvoriť aplikáciu, ktorá zachytáva pakety na sieťovom rozhraní a následne počítať, koľko paketov/bytov bolo poslaných medzi jednotlivými komunikujúcimi adresami a uviesť len 10 najviac komunikujúcich.

2. Návrh aplikácie

K implementácii bol použitý jazyk C s knižnicami pcap a ncurses. Štruktúra programu sa nachádza v jedinom zdrojovom súbore.



Obrázok 1: Návrh aplikácie

3. Popis implementácie

Spustenie programu:

`$make` – pre preloženie programu

`$sudo ./isa-top -i <interface> [-s s|b]`

Pričom:

- `-i` prepínač slúži na určenie sieťového rozhrania kde chceme odpočúvať
- `-s` prepínač, ktorý nie je povinný, určuje zoradenie výstupu podľa množstva prijatých paketov/bajtov

Po spustení programu sa spracujú vstupné argumenty, pričom sa zohľadňujú iba prepínače **-i** a **-s**. Ostatné argumenty nemajú vplyv na funkčnosť aplikácie. Následne sa inicializuje snímanie paketov s filtráciou, využívajúc knižnicu **libpcap**. Tento proces zahŕňa:

- otvorenie zariadenia na snímanie pomocou funkcie **pcap_open_live**,
- kompiláciu filtra prostredníctvom **pcap_compile**,
- aplikáciu filtra s použitím **pcap_setfilter**,
- snímanie paketov pomocou **pcap_loop**.

Funkcia **pcap_loop** volá pomocnú funkciu **got_packet**, ktorej úlohou je identifikovať linkovú vrstvu, aby bolo možné správne určiť polohu IP adres odosielateľa a prijímateľa. Ďalej funkcia **got_packet** rozpoznáva, či ide o adresu **IPv4** alebo **IPv6**. Ak sú aktuálne IP adresy už uložené v zozname komunikujúcich spojení, vyhľadá sa príslušné spojenie, pripočíta sa dĺžka paketu a inkrementuje sa počet paketov. Ak spojenie v zozname neexistuje, vytvorí sa nový záznam. Zoznam spojení je implementovaný pomocou poľa a ukazovateľa na jeho koniec.

Pre správne fungovanie aplikácie sa štatistiky aktualizujú každú sekundu pomocou funkcie **update_display(int sig)**, ktorá generuje prerušenie a volá funkciu **display_stats(int sort_option)**. Tá má za úlohu zoradiť výstup všetkých spojení podľa špecifikovaného kritéria zadaného pomocou prepínača **-s**. Ak prepínač **-s** nie je zadaný, výstup sa predvolene zoradí podľa objemu prenesených bajtov. Okrem zoradenia funkcia **display_stats** konvertuje údaje o počte bajtov a paketov do čitateľnej podoby.

Na zabezpečenie konzistentného a prehľadného výstupu bola použitá knižnica **ncurses** na formátovanie zobrazenia.

4. Informácie o programe

a. Účel:

Program **isa-top** monitoruje sieťovú prevádzku a v reálnom čase zobrazuje štatistiky prenosovej rýchlosti pre aktívne spojenia.

b. Použitie

\$make

Pre preloženie zdrojového súboru

\$sudo ./isa-top -i <interface> [-s s|b]

Kde **-i** určuje sieťové rozhranie (podporované iba pre ethernet) a **-s** spôsob radenia (podľa bajtov alebo paketov).

c. Výstup

Program zobrazuje tabuľku s údajmi o zdrojových a cieľových IP adresách, portoch, použitom protokole a prenosových rýchlostiach.

d. Požiadavky

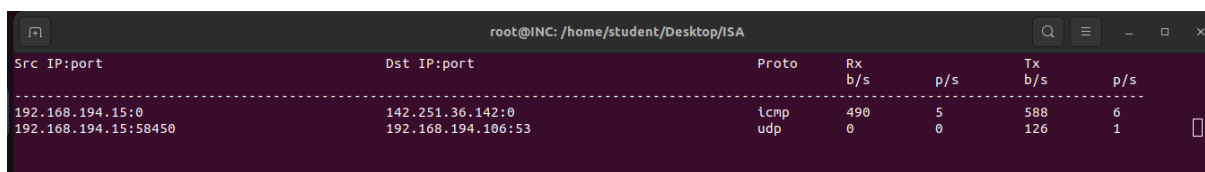
Program vyžaduje oprávnenia na čítanie sieťového rozhrania a prítomnosť knižníc libpcap a ncurses.

e. Obmedzenia

Program podporuje iba ethernetové rozhranie, pri ostatných rozhraniach sa môže správať nepredvídateľne.

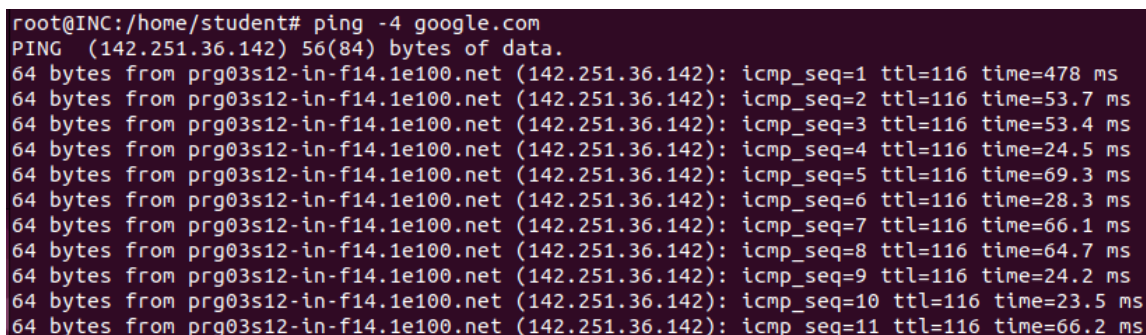
5. Popis testovania a výsledky testov

Aplikácia bola testovaná na virtuálnom stroji s operačným systémom Ubuntu 64, na rozhraní **enp0s3** (ethernet). Po spustení webového prehliadača sa monitorovali komunikujúce strany a prenášané dáta. Na zjednodušenie testovania bola aktualizácia výstupu nastavená na interval 10 sekúnd. Z tohto dôvodu sú reportované hodnoty počtu paketov a bajtov za sekundu desaťnásobne vyššie.



Src IP:port	Dst IP:port	Proto	Rx b/s	p/s	Tx b/s	p/s
192.168.194.15:0	142.251.36.142:0	icmp	490	5	588	6
192.168.194.15:58450	192.168.194.106:53	udp	0	0	126	1

Obrázok 2: Zachytávanie ICMP paketov



```
root@INC:/home/student# ping -4 google.com
PING (142.251.36.142) 56(84) bytes of data:
64 bytes from prg03s12-in-f14.1e100.net (142.251.36.142): icmp_seq=1 ttl=116 time=478 ms
64 bytes from prg03s12-in-f14.1e100.net (142.251.36.142): icmp_seq=2 ttl=116 time=53.7 ms
64 bytes from prg03s12-in-f14.1e100.net (142.251.36.142): icmp_seq=3 ttl=116 time=53.4 ms
64 bytes from prg03s12-in-f14.1e100.net (142.251.36.142): icmp_seq=4 ttl=116 time=24.5 ms
64 bytes from prg03s12-in-f14.1e100.net (142.251.36.142): icmp_seq=5 ttl=116 time=69.3 ms
64 bytes from prg03s12-in-f14.1e100.net (142.251.36.142): icmp_seq=6 ttl=116 time=28.3 ms
64 bytes from prg03s12-in-f14.1e100.net (142.251.36.142): icmp_seq=7 ttl=116 time=66.1 ms
64 bytes from prg03s12-in-f14.1e100.net (142.251.36.142): icmp_seq=8 ttl=116 time=64.7 ms
64 bytes from prg03s12-in-f14.1e100.net (142.251.36.142): icmp_seq=9 ttl=116 time=24.2 ms
64 bytes from prg03s12-in-f14.1e100.net (142.251.36.142): icmp_seq=10 ttl=116 time=23.5 ms
64 bytes from prg03s12-in-f14.1e100.net (142.251.36.142): icmp_seq=11 ttl=116 time=66.2 ms
```

Obrázok 3: Ping - pakety a veľkosť

Na priloženom obrázku (viz Obrázok 2) je viditeľné zachytávanie ICMP paketov pomocou príkazu **ping** (viz Obrázok 3). Aj keď výsledky zobrazujúce počet bajtov môžu na prvý pohľad pôsobiť nesprávne, je to spôsobené tým, že do počtu bajtov sa započítava aj veľkosť hlavičiek paketov, ktoré príkaz **ping** vo svojom výstupe nezobrazuje.

Src IP:port	Dst IP:port	Proto	Rx b/s	p/s	Tx b/s	p/s
2a00:11b1:1050:8eea:8647:1a15:8ba9:1328:43356	2a00:1450:4014:80e::200e:443	tcp	13.5k	124	261.8k	148
192.168.194.15:34688	34.120.208.123:443	tcp	12.8k	44	11.6k	53
2a00:11b1:1050:8eea:8647:1a15:8ba9:1328:43204	2a00:1450:4014:80f::2016:443	tcp	2.5k	15	6.7k	13
2a00:11b1:1050:8eea:8647:1a15:8ba9:1328:58008	2a00:1450:4014:80b::200a:443	tcp	2.4k	14	22.9k	17
2a00:11b1:1050:8eea:8647:1a15:8ba9:1328:59186	2a00:1450:4013:c02::54:443	udp	3.0k	14	16.4k	23
192.168.194.15:34702	34.120.208.123:443	tcp	1.6k	12	5.4k	9
2a00:11b1:1050:8eea:8647:1a15:8ba9:1328:35276	2a00:1450:4014:80a::2003:80	tcp	3.0k	11	4.5k	11
2a00:11b1:1050:8eea:8647:1a15:8ba9:1328:42680	2a00:1450:4013:c02::54:443	tcp	1.7k	10	6.0k	8
2a00:11b1:1050:8eea:8647:1a15:8ba9:1328:55010	2a00:1450:4014:80a::2003:443	udp	923	8	8.6k	11
2a00:11b1:1050:8eea:8647:1a15:8ba9:1328:47653	2a00:1450:4014:80f::2004:443	udp	704	6	9.6k	10

Obrázok 4: Výstup isa-top

6. Použitá literatúra

TIM CARSTENS. *Programming with pcap*. Online. Tcpdump.org. 2002, 2024. Dostupné z: <https://www.tcpdump.org/pcap.html>. [cit. 2024-11-17].

Sniffer program with C. [online]. 2021 [cit. 2024-11-16]. Dostupné z: <https://www.youtube.com/watch?v=03vhsOO5kcs>

Vichargrave.github.io [online]. 9.2012 [cit. 2024-11-16]. Dostupné z: <https://vichargrave.github.io/programming/develop-a-packet-sniffer-with-libpcap/>

Ncurses tutorial [online]. 2017, 2024 [cit. 2024-11-16]. Dostupné z: https://www.youtube.com/watch?v=lv-OPQhPvSM&list=PL2U2TQ__OrQ8jTf0_noNKtHMuYlyxQL4v&index=1