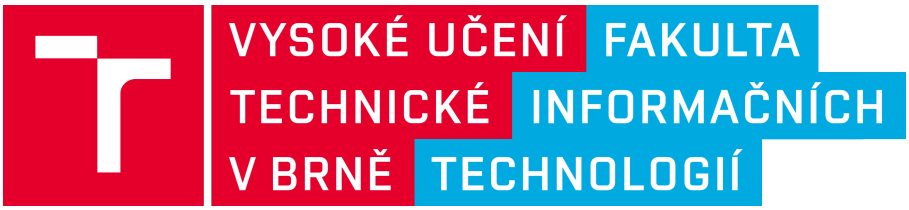


ISA projekt

Aplikace pro získání statistik o síťovém provozu

Norman Babiak (xbabia01)

Brno, 18. listopadu 2024

Obsah

1 Teorie

1.1 Protokoly pro adresaci a směrování

1.1.1 IPv4

IPv4 adresa je logická adresa zařízení v síti IP. Skládá se ze 4 částí zvaných octety, každá část je veliká 8 bitů, a zapisuje se oddělená tečkou. Adresa se většinou zapisuje v dekadické formě, ale pro výpočet je jasnější binární zápis. Teoreticky je tedy adresní rozsah od 0.0.0.0 do 255.255.255.255. Příkladem IP adresy je třeba 68.12.5.10. Hlavička obsahuje obecně tyto pole: cílová adresa, zdrojová adresa a protokol, který rámec nese. Internet protokol byl navrhnut i s tzv. IP options – což měly být pole, které se přidávají hned za hlavičku. Options pole mohou být různých délek a v různém počtu až do velikosti 40 bajtů – IP hlavička tedy má délku 20 – 60 bajtů. Dnes se options již nepoužívá, takže délka je pevná – 20 bajtů.

1.1.2 IPv6

V případě IPv6 je to trochu jiné, ale princip je stejný. Mějme IPv6 adresu 2001:0db8:85a3::8a2e:0370:7334/64. Zde je 2001:0db8:85a3::8a2e síťová část a 0370:7334 je hostitelská část. Číslo za lomítkem (/64) je prefix, který nám říká, kolik bitů adresy patří k síti a které části definují specifické zařízení. Prefix úzce souvisí s tzv. beztřídní adresací. Třídní adresace byla původní metoda pro rozdělení IPv4 adresního prostoru. Systém byl založen na čtyřech třídách adres - třída A, B, C a D. Přestože tento systém byl jednoduchý a snadno pochopitelný, nebyl dostatečně flexibilní a vedl k velkému plýtvání adresním prostorem. Classless Inter-Domain Routing (CIDR) systém nahradil třídní adresaci a umožnil mnohem efektivnější využití adresního prostoru. V CIDR může být délka síťové části jakékoliv délky, nikoli pouze násobky oktetu, jak tomu bylo v třídní adresaci.

1.2 Základní protokoly

1.2.1 TCP

Transmission Control Protocol je přepravní protokol, který se používá nad rámec IP k zajištění spolehlivého přenosu paketů. TCP obsahuje mechanismy pro řešení mnoha problémů, které vznikají při odesílání zpráv ve formě paketů, jako jsou ztracené pakety, pakety ve špatném pořadí, duplicitní pakety a poškozené pakety.
TCP spojení může detekovat ztracené pakety pomocí časového limitu. Po

odeslání paketu spustí odesílatel časovač a umístí paket do fronty pro přeposílání. Pokud časovač vyprší a odesílatel ještě od příjemce ACK neobdržel, odešle paket znovu.
TCP spojení můžou detekovat pakety v nesprávném pořadí použitím pořadových a potvrzovacích čísel. Pokud příjemce vidí vyšší pořadové číslo, než jaké dosud potvrdil, ví, že mu chybí alespoň jeden paket. Ve výše uvedené situaci příjemce vidí pořadové číslo 73, ale očekával pořadové číslo 37. Příjemce dává odesílateli vědět, že něco chybí a odesílá proto paket s potvrzovacím číslem nastaveným na očekávané pořadové číslo.

1.2.2 UDP

UDP (User Datagram Protocol) je základním protokolem transportní vrstvy v architektuře TCP/IP. Je také označován jako tzv. nespojový protokol. Velkou výhodou UDP je malé zatížení sítě a naopak nevýhodou je to, že není nijak zajištěno, jestli sítí přenášené datagramy se ztratí, poškodí či se nezmění pořadí jejich doručení, také může docházet k duplikacím dat. Tyhle nevyhody potom musí řešit aplikace, která UDP pakety příjmá. V případě neřešení chyb, může dojít ke ztrátě poslaných paketů, což vede ke ztrátě informací.

1.2.3 ICMP

Protokol ICMP (Internet Control Message Protocol) je pomocný protokol používaný k diagnostice a monitorování sítě. Používá se především k přenosu zpráv o chybách a dalších výjimečných situacích, ke kterým dochází při přenosu dat po síti. ICMP má také některé servisní funkce. Protokol ICMP je popsán v dokumentu RFC 792. Protokol ICMP je protokol síťové vrstvy modelu OSI. ICMP zprávy mají jednoduchou strukturu, která se skládá z hlavičky a datové části. Hlavička obsahuje informace o typu zprávy a kontrolní součet, který zajišťuje integritu dat. Datová část závisí na typu zprávy a může obsahovat různé informace, jako například IP adresu nebo názvy sítí. ICMP může být cílem útoků, jako je DDoS (Distributed Denial of Service), zejména pokud jsou do sítě zasílány velké množství ICMP zpráv. Pro správnou bezpečnost sítě je proto důležité umožnit pouze nezbytný provoz ICMP a implementovat ochranná opatření proti možným útokům.

1.3 Přenosová rychlost

Přenosová rychlost udává, jaký objem informace se přenese za jednotku času. Základní jednotkou přenosové rychlosti je bit za sekundu. Jednotka

udává, kolik bitů informace je přeneseno za jednu sekundu. Existují dva typy garance přenosové rychlosti: **garantované** a **agregované**.

Garantované jsou nabízeny zejména pro ISP, servery, velké firmy a korporace, náročné zákazníky domácích přípojek. V tomto případě zaručení, že linka neklesne pod danou úroveň. Tímto způsobem je zaručeno, že linka propustí udávanou rychlost, pod níž nesmí za žádných podmínek klesnout. Garantovaná rychlost internetu 10 Mb/s určuje klientovi, že od svého nadřazeného serveru propustí za všech okolností minimálně onu rychlost 10 Mb/s.
Agregované jsou nabízeny zejména pro střední a menší firmy, domácnosti, neplacené hotspoty, mobilní sítě. V tomto případě označuje, že linka shlukuje několik klientů pro dosažení garance. Agregace se udává v poměru x:y přičemž x znamená minimální podíl a y maximální podíl z rychlosti. Agregace internetu 1:10 znamená, že klient mající např. rychlost internetu 10Mb/s v agregaci 1:10 má dynamickou rychlost připojení v rozsahu od 1 Mb/s až 10 Mb/s přičemž reálná rychlost zpravidla nabývá hodnot primárně určené rychlosti tj. 10 Mb/s.

2 Základní informace aplikace

2.1 Popis aplikace

Aplikace ISA-TOP slouží na zobrazení aktuální přenosové rychlosti pro IP adresy, které komunikují v daném čase. Program zachytává přenos na síťovém rozhraní které bylo zpecifikované uživatelem, kde taktéž vypočítá přenosovou rychlost za nějaký pevně daný čas.

2.2 Možnosti různých nastavení

ISA-TOP podporuje různé nastavení, které specifikuje uživatel. Jedním z nastavení je, jak již bylo zmíněno, síťové rozhrtaní na kterém bude zachytávat přenos. Druhé nastavení, které může uživatel nastavit je čas za který se displej znova načítá pro zobrazení. Dalším nastavením je řazení daných spojení v terminálu a to lze buď pomocí základní hodnoty b, jako bity, nebo podle hodnoty p, pro přenesené pakety. U obou variant se to řadí sečtením příchozích a odchozích bitů/paketů.

3 Návod na použití

3.1 Potřebné nástroje pro spouštění programu

- Libpcap: `sudo apt install libpcap-dev`
- Ncurses: `sudo apt install libncurses5-dev libncursesw5-dev`
- C++17: `sudo apt install g++ gcc`
- Arpa, netinet: `sudo apt install build-essential`

3.2 Základní spouštění programu

```
./isa-top -i int [-s [b|p]] [-t time]
```

- -i int - Síťové rozhraní na kterém bude ISA-TOP zobrazovat přenos
- -s [b—p] - Řazení výstupu aplikace podle bitů/paketů
- -t time - Čas, za který displej znovu vypíše přenos za uplynulou dobu

3.3 Příklady spouštění

Aplikace používanmá na síťovém rozhraní eth0, kde výstup se řadí podle počtu paketů. Displej se obnovuje po základní časové délce 1s:

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

Aplikace používaná na síťovém rozhraní eth0, kde se řadí podle základní hodnoty -s, a to podle bitů:

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

Aplikace používaná na síťovém rozhraní eth0, kde se řadí podle bitů, a obnova displeje je každých 6 sekund:

```
./isa-top -i eth0 -s b -t 6
```

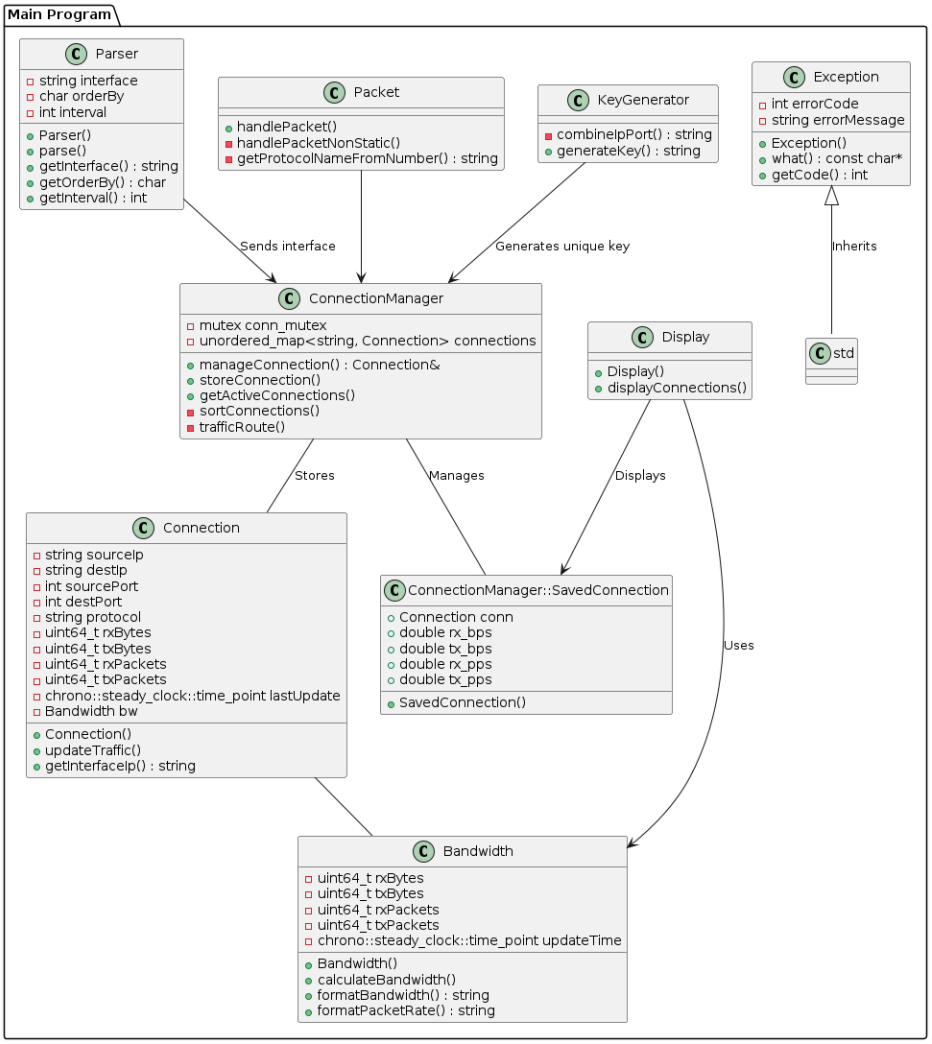
Nesprávné spouštění aplikace, nesmí chybět síťové rozhraní:

```
./isa-top -s p -t 4
```

4 Implementace

Aplikace ISA-TOP byla implementovaná v souladu ze zadáním, kdežto jsou naimplementované funkce navíc. Aplikace podporuje více protokolů jak UDP, TCP a ICMP, těmito protokoly jsou například GRE, ESP... Další funkce bylo implementace přepínače -t, který bude určovat za jaký časový interval se statistiky aktualizují. V základním případě při aktualizaci za sekundu, a taktéž při aktualizaci za x sekund, se pokud nějaké dvě IP adresy přestanou komunikovat, displej je již nezobrazí (v případě že Tx a Rx jsou oba 0). V případě že komunikující se IP adresy neshodují s IP adresou počítače kde je program spuštěn, první komunikace určuje směr, který je vždy odchozí. Taktéž bylo naimplementováno ukončení aplikace pomocí stlačení kláves CTRL + C.
Program je implementovaný pomocí vícevláknového procesu, kde jedno vlákno se zabývá aktualizaci výstupu programu pomocí instancí **Display** a **ConnectionManager**, přičemž druhé vlákno zpracovává příchozí pakety v instanci **pcap_t**.

4.1 UML Diagram



4.2 Použité knihovny

4.2.1 Libpcap

Knihovna libpcap je určená pro zachytávání síťového provozu na úrovni paketů. Umožňuje programům přistupovat k datům přenášeným přes síťová rozhraní a zachytávat síťovou komunikaci pro účely analýzy, monitorování, diagnostiky a bezpečnostních kontrol.

4.2.2 Ncurses

Ncurses je knihovna určená pro tvorbu textových uživatelských rozhraní v terminálu. Umožňuje vývojářům vytvářet komplexní rozhraní s použitím různých prvků, jako jsou okna, tlačítka, formuláře, posuvníky a další, bez nutnosti práce s grafickým rozhraním. Knihovna je napsána v jazyce C.

4.3 Třídy

4.3.1 Parser

Táto třída slouží na parsování argumentů, které uživatel zadá při spuštění aplikace. Nastavuje základní hodnoty pro přepinače -s a -t, přičemž funkce **parser** potom pracuje s jejich argumenty. V případě chyb využívá třídu Expcetion pro správný výpis.

4.3.2 Packet

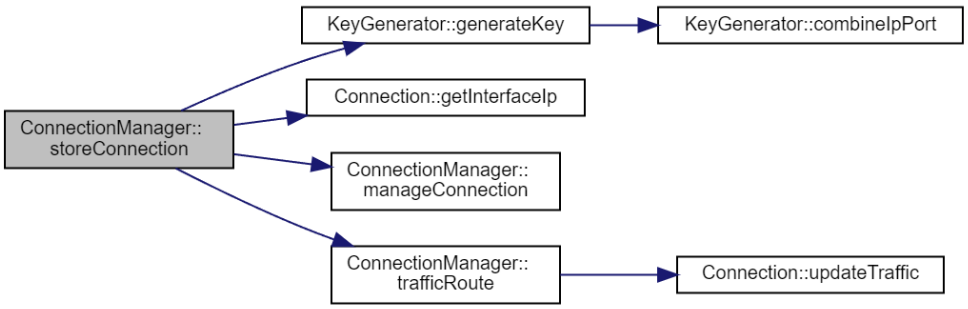
Packet je třída, která využívá knihovny Libpcap pro zpracování příchozích paketů a získání důležitých informací z nich (IP, port, protokol). Získaná spojení jsou poté uložené a dále zpracované pomocí třídy **ConnectionManager**.

4.3.3 Bandwidth

Třída sloužící pro výpočet a formátování přenesených paketů a bitů. Pro vlastní funkčnost taktéž ukládá čas ve kterým byla daná komunikace aktualizovaná (**updateTime**).

4.3.4 ConnectionManager

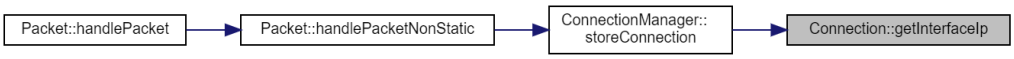
ConnectionManager je vpodstatě hlavní třídou programu po **main**, jelikož spravuje jednotlivá spojení. V případě že spojení ještě neexistovalo předtím, uloží ho a postupně aktualizuje pokud se objevuje vícekrát.
Příklad funkce **storeConnection** a jeho komunikace s ostatními třidami a funkcemi:



Pokud po čase spojení zanikne, a nebudou přenášená žádná data, už je neuloží a neposílá pro **Display** pro zobrazení. Taktéž rozpoznává jestli jsou data odesílaná nebo příjimaná, přičemž základní nastavení je takové, že pokud cílová IP adresa není naša, data se automaticky berou jako odesílané při prvním spojení.

4.3.5 Connection

Táto třída slouží na uchovávání informací pro jednotlivé spojení. Sleduje statistiky přenosu dat a aktualizuje počet přenesených paketů/bajtů pomocí funkce **updateTraffic**. Třída taktéž obsahuje funkci pro získání IP adresy rozhraní na kterém aplikace poslouchá, aby se zjistilo jestli jsou směřována na počítač na kterém běží aplikace, nebo z ní:



4.3.6 KeyGenerator

Třída slouží na generování jediněčného klíče pro každé spojení na základě IP adres, portů a protokolů. Využívá se v **ConnectionManager** pro generování **forward** a **reverse** klíče pro identifikování spojení.

4.3.7 Display

Táto třida slouží pro zobrazení aktivních síťových spojení a jejich rychlost. Používá knihovnu ncurses a pravidelně aktualizuje výstup.

4.3.8 Exception

Třída sloužící pro snadnější výpis chyby a návratového kódu k němu.

5 Testování

Testování probíhalo ve většíně na virtualním počítači s **Ubuntu 22.10**, na **Ubuntu WSL**. Přeložení programu se taktéž testoval na servery **merlin.fit.vutbr.cz**.

Pro testování IPv4 a IPv6 byl použitý python skript (odevzdán s projektem), který pomocí knihovny **scapy** posílál pakety různých transportních protokolů.

5.1 Testování IPv4

Použitý skript: ipv4_tests.py
Potřebné instalace: python 3, scapy

Spouštění programu isa-top:

```
./isa-top -i lo -s p
```

Spouštění python skriptu pro test:

```
python3 ./tests/ipv4_tests.py
```

Skriptem se opakovaně posílají protokoly: TCP, UDP, ICMP, ESP, GRE.
Výstup programu:

```
ISA-TOP
Src IP:Port             | Dst IP:Port             | Protocol | Rx Rate   | Tx Rate   | Rx P/s   | Tx P/s
127.0.0.1:42282         | 127.0.0.1:34359         | tcp      | 13.1 Kbps | 0.0 bps   | 6.0 pps  | 0.0 pps
127.0.0.1:34359         | 127.0.0.1:42280         | tcp      | 6.8 Kbps  | 0.0 bps   | 4.0 pps  | 0.0 pps
127.0.0.1:0             | 127.0.0.1:0             | gre      | 423.9 bps | 0.0 bps   | 1.0 pps  | 0.0 pps
127.0.0.1:0             | 127.0.0.1:0             | esp      | 391.9 bps | 0.0 bps   | 1.0 pps  | 0.0 pps
127.0.0.1:12345         | 127.0.0.1:80            | tcp      | 551.8 bps | 0.0 bps   | 1.0 pps  | 0.0 pps
127.0.0.1:12345         | 127.0.0.1:80            | udp      | 455.8 bps | 0.0 bps   | 1.0 pps  | 0.0 pps
127.0.0.1:0             | 127.0.0.1:0             | icmp     | 463.8 bps | 0.0 bps   | 1.0 pps  | 0.0 pps
```

5.2 Testování IPv6

Použitý skript: ipv6_tests.py
Potřebné instalace: python 3, scapy

Spouštění programu isa-top:

```
./isa-top -i lo -s p
```

Spouštění python skriptu pro test:

```
python3 ./tests/ipv6_tests.py
```

Skriptem se opakovaně posílají protokoly: TCP, UDP, ICMPv6, ESP.
Výstup programu:

Bibliografie

[1] SAMURAJ-CZ.COM. TCP/IP - adresy, masky, subnety a výpočty [online]. [cit. 2024-11-03]. Dostupné z: https://www.samuraj-cz.com/clanek/tcpip-adresy-masky-subnety-a-vypocty/

[2] ITNETWORK.CZ. Sítě: IPv4 a IPv6 - tvorba podsítí [online]. [cit. 2024-11-03]. Dostupné z: https://www.itnetwork.cz/site/zaklady/site-ipv4-a-ipv6-tvorba-podsiti

[3] KHAN ACADEMY. Transmission Control Protocol (TCP) [online]. [cit. 2024-11-03]. Dostupné z: https://cs.khanacademy.org/computing/informatika-pocitace-a-internet/x8887af37e7f1189a:internet/x8887af37e7f1189a:tcp-protokol/a/transmission-control-protocol--tcp

[4] SPRÁVA-SÍTĚ.EU. User Datagram Protocol (UDP) [online]. [cit. 2024-11-03]. Dostupné z: https://www.sprava-site.eu/udp/

[5] ROJE2014.WEBNODE.CZ. Přenosová rychlost [online]. [cit. 2024-11-10]. Dostupné z: https://roje2014.webnode.cz/prenosova-rychlost/

[6] TCPDUMP/LIBPCAP. Libpcap [online]. [cit. 2024-11-10]. Dostupné z: https://www.tcpdump.org/

[7] GNU NCURSES. Ncurses Library [online]. [cit. 2024-11-10]. Dostupné z: https://invisible-island.net/ncurses/