Vysoké učení technické v Brně
Fakulta informačních technologií

Síťové aplikace a správa sítí
Monitorovaní DHCP komunikace

17. novembra 2023 Patrik Potančok

Obsah

1 Uvedenie do problematiky

V súčasnej dobe je internet neoddeliteľnou súčasťou našich životov. Od obyčajných domácností až po veľké podniky, dynamická alokácia IP adries je kľúčová pre plynulé a efektívne fungovanie sietí. Ako sa zdá byť jednoduchý proces pripojenia k internetu pre bežného používateľa, za tým stojí komplexná sústava protokolov a mechanizmov, ktoré to umožňujú. Medzi týmito mechanizmami je DHCP jedným z najdôležitejších, ktorý umožňuje automatickú distribúciu IP adries v sieti. S rastúcim počtom počítačov s možnosťou pripojenia na internet bolo nereálne prideľovať týmto počítačom IP adresy ručne. Preto vznikol protokol Bootstrap. Počítač, ktorý po spustení nepozná svoju IP adresu ani IP adresu serveru, z ktorého si načíta konfiguráciu, vyšle správu typu broadcast. Bootstrap potom pridelí zariadeniu stále tú istú IP adresu. Jeho priamym nástupcom je protokol DHCP. DHCP (Dynamic Host Configuration Protocol) rozširuje možnosti dynamickej konfigurácie tak, že zariadenie nedostane vždy tú istú IP adresu. IP adresy sú prideľované na obmedzený čas, a po ukončení spojenia môže prideliť rovnakú IP adresu inému zariadeniu [2].
V dnešnej dobe je dôležité mať kedykoľvek prehľad o alokácii IP adries. Keď je veľký počet IP adries pridelený, môže to viesť k nedostatku dostupných adries pre nové zariadenia, čo môže spôsobiť sieťové problémy.

2 Teoretický základ

2.1 DHCP

DHCP (Dynamic Host Configuration Protocol) je sieťový protokol, ktorý umožňuje automatickú konfiguráciu IP adries a ďalších sieťových parametrov pre zariadenia pripojené k sieti. DHCP je založený na klient-server architektúre, kde DHCP server poskytuje konfiguračné informácie klientom[1].

2.2 DHCP pakety

Existujú rôzne typy DHCP paketov, ktoré sa používajú pri komunikácii medzi DHCP klientom a DHCP serverom. Tieto pakety zahŕňajú:

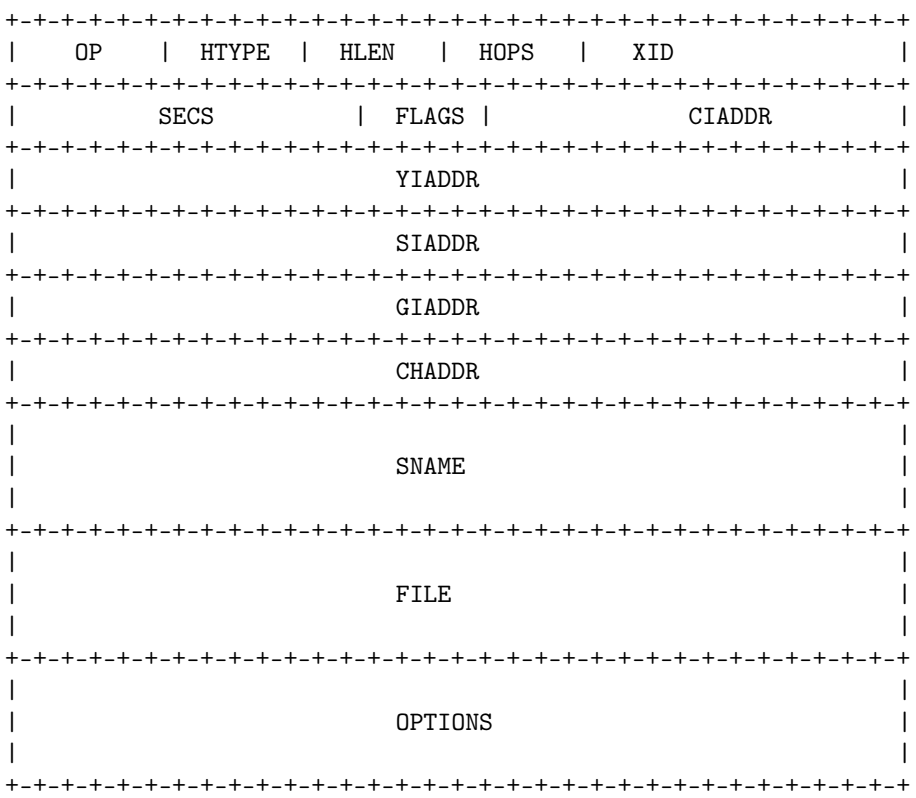
- **DHCP Discover**: Klient odosiela tento paket do siete, aby vyhľadal dostupné DHCP servery. Tento paket obsahuje informácie o možnostiach, ktoré klient preferuje.
- **DHCP Offer**: DHCP server odpovedá na DHCP Discover paket klienta pomocou DHCP Offer paketu. Tento paket obsahuje ponuku IP adresy a ďalších sieťových parametrov pre klienta.
- **DHCP Request**: Klient vyberá jednu z ponúkaných IP adries z DHCP Offer paketu a odosiela DHCP Request paket, v ktorom žiada o túto IP adresu.
- **DHCP Acknowledge (ACK)**: DHCP server potvrdzuje vyžiadanú IP adresu klienta odpoveďou DHCP Acknowledge paketu. Tento paket obsahuje pridelenú IP adresu a ďalšie konfiguračné informácie.
- **DHCP Inform**: Je špecifický typ DHCP správy, ktorý je používaný v situáciách, keď DHCP klient požaduje dodatočné informácie od DHCP servera, ale už má pridelenú IP adresu. Tento typ správy je typicky využívaný v situáciách, kde klient získa svoju IP adresu staticky alebo z iného zdroja a používa DHCP INFORM na získanie ďalších konfiguračných informácií, ako sú napríklad adresy DNS serverov, nastavenie brány a podobne.

Každý DHCP paket obsahuje nasledujúce časti:

- **op (Operation Code)**: Označuje, či ide o DHCP dotaz alebo odpoveď. Toto pole určuje, čo presne je obsiahnuté v DHCP pakete, či ide o požiadavku na pridelenie IP adresy alebo o odpoveď zo servera.
- **htype (Hardware Type)**: Udáva typ hardvérovej adresy, ktorá je použitá pre identifikáciu klienta. Typy môžu zahŕňať Ethernet, WiFi a ďalšie.
- **hlen (Hardware Length)**: Toto pole udáva dĺžku hardvérovej adresy v oktetoch. Rôzne typy hardvérových adries môžu mať rôznu dĺžku.
- **hops**: Počet skokov, ktorými prešiel DHCP paket. Toto pole je dôležité pre DHCP relay agentov, ktoré preposielajú DHCP pakety medzi klientom a serverom.
- **xid (Transaction ID)**: Jedná sa o unikátny identifikátor transakcie, ktorý pomáha spojiť DHCP požiadavky a odpovede. Identifikátor transakcie je náhodne generovaný klientom a je zahrnutý vo všetkých DHCP správach.

- **secs (Seconds)**: Udáva počet sekúnd od začiatku bootovania klienta. Toto pole pomáha serveru určiť, ako dlho je klient aktívny.
- **flags**: Toto pole označuje, či je DHCP paket broadcast alebo unicast. Broadcast pakety sú určené všetkým zariadeniam v sieti, zatiaľ čo unicast pakety sú určené konkrétnemu klientovi.
- **ciaddr (Client IP Address)**: Toto pole obsahuje IP adresu klienta, ktorá sa používa v odpovedi od servera. Je to dôležité, keď klient obnovuje svoju existujúcu IP adresu.
- **yiaddr (Your IP Address)**: Toto pole obsahuje IP adresu, ktorú server pridelil klientovi. Je to adresa, ktorú klient používa po úspešnom získaní IP adresy.
- **siaddr (Server IP Address)**: IP adresa DHCP servera, ktorý komunikuje s klientom. Je to server, ktorý prideluje IP adresy klientom.
- **giaddr (Gateway IP Address)**: IP adresa relay agenta, ak je prítomný. Relay agent je sprostredkovateľ medzi klientom a serverom DHCP.
- **chaddr (Client Hardware Address)**: Toto pole obsahuje hardvérovú adresu (MAC adresu) sieťovej karty klienta. Slúži na jednoznačnú identifikáciu klienta v sieti.
- **sname (Server Name)**: Voliteľné pole, ktoré môže obsahovať meno DHCP servera.
- **file (Boot File Name)**: Voliteľné pole, ktoré môže obsahovať názov obrazového súboru, ktorý klient použije pri bootovaní.
- **options (Options)**: Toto pole obsahuje ďalšie volby a konfiguračné informácie, ktoré môžu byť súčasťou DHCP paketu. Tieto informácie môžu zahŕňať IP adresy DNS serverov, brány, masky siete a ďalšie konfiguračné parametre.

Tu je príklad štruktúry DHCP paketu:



Táto štruktúra reprezentuje jednotlivé polia a ich pozície v DHCP pakete. Každé pole má špecifický účel v procese DHCP komunikácie[1].

2.3 Použité knižnice

- **getopt** - Táto knižnica slúži na analýzu argumentov príkazového riadku, čo umožňuje efektívne spracovanie vstupných parametrov programu.
- **ncurses** - Grafická knižnica pre konzolové aplikácie, ktorá umožňuje vytváranie užívateľsky prívetivého rozhrania v textovom režime.
- **syslog** - Táto knižnica slúži na zaznamenávanie správ do systémového logu, čo umožňuje sledovanie dôležitých udalostí a chýb programu.
- **pcap** - Knižnica pcap je využívaná na zachytávanie paketov a monitorovanie sieťovej komunikácie, čo umožňuje analýzu DHCP prevádzky.

3 Návrh aplikácie

Cieľom aplikácie je monitorovať DHCP komunikáciu v sieti a sledovať štatistiky pridelených IP adries v určených IP prefixoch. Program pracuje s knižnicou `pcap`, aby analyzoval sieťový prevádzku a identifikoval DHCP správy, ktoré signalizujú pridelenie IP adresy. Používatelia môžu špecifikovať rozhranie, na ktorom chcú sledovať prevádzku, alebo môžu zadefinovať súbor pcap na analýzu. K implementácii bol použitý jazyk C++ a knihovňa `libcap`. Program je rozdelený na niekoľko zdrojových kódov a dva hlavičkové súbory.

3.1 Flowchart diagram

3.2 Spracovanie argumentov

Aplikácia spracováva argumenty pomocou knižnice `getopt` a pomocou funkcie `check_args`. Skontroluje sa, či je na vstupe parameter pre čítanie zo súboru alebo naslúchanie priamo na rozhraní. Je možné zadať iba parameter `-i` alebo parameter `-r`. Podľa toho sa potom do premennej uloží buď názov súboru alebo názov rozhrania. Potom sa IP prefixy zo vstupu uložia do vektoru. Nakoniec sa všetkým prefixom skontroluje správnosť syntaxu. Po kontrole syntaxu sa vytvorí pre každý prefix inštancia triedy `Prefix`.

3.3 Čítanie z pcap súboru

Najskôr sa kontroluje, či nie je pcap súbor prázdny. Keďže pracujeme offline použijeme funkciu z knižnice `pcap.h`, `pcap_open_offline()`, potom nastavíme filter na `udp and port 67 or port 68`. Aplikácia potom spracúva DHCP pakety čo bude opísané v ďalšej časti.

3.4 Naslúchanie na rozhraní

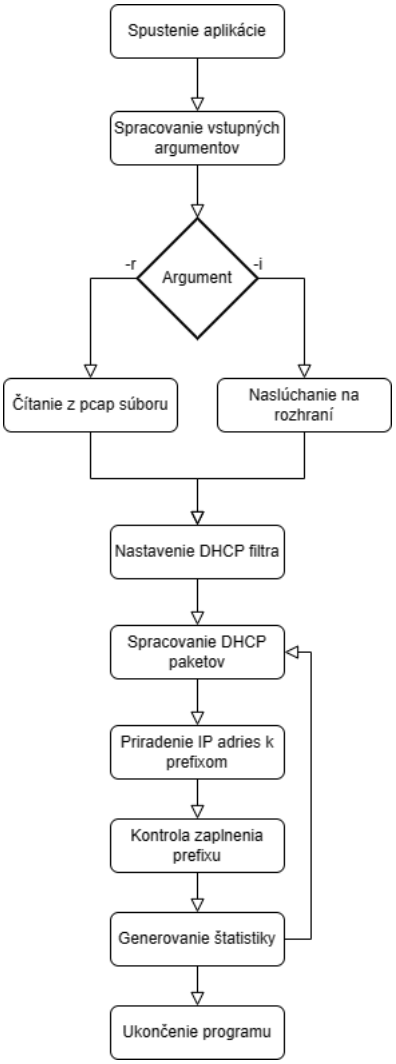
Pri naslúchaní na rozhraní sa najskôr skontroluje, či dané rozhranie existuje. Ak nie je zadané správne rozhranie, aplikácia vypíše dostupné rozhrania. Následne sa kontroluje, či má užívateľ dostatočné oprávnenia pre naslúchanie na rozhraní. Rovnako ako pri čítaní zo súboru sa nastaví rovnaký filter, a potom sa spracúvajú DHCP pakety.

3.5 Spracovanie DHCP

Na začiatku sa nastaví filter na zachytávanie DHCP paketov. Pomocou funkcie
`pcap_loop(descr, 0, callback, NULL)` začne zachytávanie paketov. Funkcia `callback` sa zavolá pre každý zachytený paket. V tejto funkcii sa získajú dáta DHCP paketu a vyfiltrujú sa iba DHCP ACK alebo INFORM pakety. Z paketu sa získa yiaddr/ciaddr adresa a skontroluje sa či už bola videná predtým. Ak nie, tak sa hľadá prefix zo vstupu, ku ktorému daná IP adresa patrí a pripočíta sa počítadlo alokovaných IP adries daného prefixu.

3.6 Generovanie štatistiky

Štatistika sa vypočítava a vypisuje v funkcii `callback` pomocou knižnice `ncurses` a triedy `Prefix`, ktorá obsahuje pomocné funkcie na vypočítanie počet maximálnych alokovaných adries, počet momentálne alokovaných adries a výpočet percenta využitia prefixu. Pomocou knižnice `ncurses` sa potom vypíše štatistika na výstup.



Obr. 1: Flowchart diagram aplikácie

4 Popis implementácie

4.1 Trieda Prefix

Trieda `Prefix` reprezentuje IP prefix a poskytuje metódy pre manipuláciu s ním.

4.1.1 Atribúty

- `ip_address`: IP adresa v reťazcovej forme.
- `prefix_length`: Dĺžka prefixu.
- `current_hosts`: Aktuálny počet hostiteľov v danom prefixe.
- `max_hosts`: Maximálny počet hostiteľov, ktorí môžu byť v danom prefixe.
- `usage_flag`: Príznak označujúci, či prefix prekročil 50% alokovaných IP adries.

4.1.2 Metódy

- Konštruktor a deštruktor: Inicializuje atribúty triedy a v prípade potreby uvoľňuje zdroje.
- `usage()`: Vypočíta a vráti aktuálne využitie prefixu v percentách.
- `ip_to_int()`: Prevádza IP adresu z reťazcovej formy na celočíselnú formu.

- `ip_belongs()`: Kontroluje, či zadaná IP adresa patrí do daného prefixu.
- `increment_host_count()`: Zvyšuje počet aktuálnych hostiteľov v prefixe o jedna.
- Gettery a settery: Poskytujú prístup k hodnotám atribútov a umožňujú ich nastavenie.
- `to_string()`: Vracia textovú reprezentáciu prefixu v tvare "`IP_adresa/prefix_length`".

4.2 Callback funkcia

Funkcia `callback` je spätné volanie, ktoré je vyvolané knižnicou `libpcap` pre každý zachytený paket. Táto funkcia analyzuje DHCP pakety a filtruje z nich iba tie, ktoré sú typu DHCP ACK. Z týchto paketov získava `yiaddr` alebo `ciaddr` adresu, čo je IP adresa pridelená klientovi. Nasleduje kontrola, či bola táto adresa videná predtým. Ak nie, skontroluje sa, do ktorého prefixu táto IP adresa patrí a inkrementuje sa počítadlo pridelených IP adries pre daný prefix.

4.3 Funkcia check_args

Funkcia `check_args` zodpovedá za spracovanie argumentov z príkazového riadka. Vstupné parametre sú kontrolované s použitím knižnice `getopt`. Funkcia skontroluje, či bol zadaný parameter pre čítanie zo súboru (parameter -r) alebo pre naslúchanie na sieťovom rozhraní (parameter -i). V závislosti od zvoleného parametra sa následne uloží buď názov súboru alebo názov rozhrania. Ďalšie parametre predstavujú IP prefixy, ktoré sú uložené do vektora.

- Kontroluje sa, či je počet argumentov väčší ako 1, inak vypíše návod na použitie a ukončí program.
- Definujú sa možné dlhé argumenty pomocou štruktúry `option`.
- Následne sa spracúvajú jednotlivé argumenty pomocou `getopt_long`. Podľa získaných argumentov sa nastaví príznak `i_flag` alebo `r_flag` a príslušné hodnoty súboru alebo rozhrania. Takisto sa pridávajú IP prefixy do vektora `ip_prefixes`.
- Na konci sa skontroluje, či bol zadaný buď argument -r alebo -i, ale nie oba naraz.

4.4 Funkcia valid_prefix

Táto funkcia kontroluje platnosť IP adresy/prefixu podľa regulárneho výrazu. Funkcia vráti true, ak je zadaný reťazec platný IP prefix, inak vráti false. Tu je zobrazený regex, ktorým sa kontroluje správnosť IP prefixu.

```
std::regex ip_prefix{
    "^(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\\."
    "(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\\."
    "(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\\."
    "(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)(/([0-9]|[1-2][0-9]|3[0-2]))?$"};
```

4.5 Funkcia callback

Funkcia `callback` je spätné volanie, ktoré je vyvolané knižnicou `libpcap` pre každý zachytený paket.

- Údaje z DHCP paketu sú spracovávané preskočením Ethernet, IP a UDP hlavičiek.
- Hľadá sa DHCP ACK alebo DHCP INFORM v možnostiach DHCP.
- Ak ide o DHCP ACK alebo INFORM, extrahuje sa `yiaddr` alebo `ciaddr`, ktorá je IP adresa pridelená klientovi.
- Adresa je konvertovaná na reťazec.
- Kontroluje sa, či bola táto IP adresa videná predtým. Ak nie, pridá sa do zoznamu videných IP adries.
- Pre každý prefix v zozname sa skontroluje, či pridaná IP adresa patrí do daného prefixu. Ak áno, inkrementuje sa počítadlo pridelených IP adries pre daný prefix.
- Ak použitie pre daný prefix presiahne 50%, loguje sa chybové hlásenie do syslogu a do `ncurses` okna.
- Na konci sa vygeneruje štatistika

4.6 Hlavná Funkcia (main)

Hlavná funkcia programu má na starosti riadenie monitorovania DHCP komunikácie. Jej úlohou je inicializovať potrebné premenné a rozhrania pre zachytávanie sieťovej prevádzky a následne volať callback funkciu na spracovanie každého zachyteného paketu.

- zavolá s funkcia `check_args` a `valid_prefix`.
- Ďalej prebieha inicializácia inštancií triedy `Prefix` na základe zadaných IP prefixov. Pre každý zadaný prefix sa vytvorí táto inštancia, aby sa mohlo sledovať využitie IP adries v danom prefixe.
- Následne je nastavená obsluha signálu pre prípadné ukončenie programu klávesovou skratkou Ctrl+C.
- Podľa argumentov sa buď program pripraví na naslúchanie na konkrétnom sieťovom rozhraní alebo na čítanie zo súboru formátu PCAP.
- V prípade naslúchania na rozhraní sa vykonajú overenia, či je zadané rozhranie platné a či má používateľ dostatočné práva na naslúchanie. Taktiež sa inicializuje rozhranie pre zachytávanie paketov pomocou knižnice pcap.
- Ak program pracuje s vstupným súborom PCAP, overí sa, že súbor nie je prázdny, a potom sa s ním inicializuje čítanie.
- Nakoniec sa nastaví filter pre zachytávanie len DHCP paketov na portoch 67 a 68, kompiluje sa a aplikuje sa na rozhranie alebo súbor, a spúšťa sa samotné zachytávanie paketov.
- Pre každý zachytený paket sa zavolá callback funkcia na jeho spracovanie.
- Štatistiky sa postupne obnovujú, aplikáciu je možne ukončiť pomocou CTRL+C.

4.7 Zaujímave časti kódu

4.7.1 Funkcia valid_prefix

Tato funkcia pomocou regexu dokáže skontrolovať správnosť IP regexu na vstupe a tak predísť rôznym problémom spojených zo zle zadaným prefixom. Funkcia sama o sebe je jednoduchá, ale dokáže efektívne odstrániť neplatné IP prefixy.

```
bool valid_prefix(std::string ip)
{
    std::regex ip_prefix{
        "^(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\\."
        "(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\\."
        "(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\\."
        "(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)(/([0-9]|[1-2][0-9]|3[0-2]))?$"};

    return std::regex_match(ip, ip_prefix);
}
```

4.7.2 Metóda ip_belongs

Najskôr sa yiaddr/ciaddr IP a IP prefix konvertujú na ich číselnú reprezentáciu. Potom sa získa maska daného prefixu, pomocou bitových posunov `1 << (32 - this->prefix_length)`, 1 sa posunie doľava o určený počet pozícií, čo nám dáva hodnotu s jedničkou na najvyššej pozícii hostiteľskej časti. Potom sa odčíta 1, čo zmení všetky bity napravo od začiatku hostiteľskej časti. Nakoniec sa bity prevrátia pomocou bitovej inverzie a dostaneme masku siete. Napr. pre prefix `192.168.1.0/24`:

- Z prefixu vieme, že `this->prefix_length` je 24. Toto znamená, že prvých 24 bitov IP adresy je vyhradených pre sieť, zatiaľ čo zvyšných 8 bitov je vyhradených pre hostiteľa.
- Vypočítame `(32 - this->prefix_length)`, čo je `(32 - 24) = 8`. Toto znamená, že máme 8 bitov vyhradených pre hostiteľa.
- Teraz, 1 « ... je 1 « 8, čo v binárnom formáte je 100000000.

- Potom, ... - 1 je 100000000 - 1, čo je 011111111.
- Nakoniec, (...) prevráti všetky bity z 011111111 na 100000000.
- Takže výsledok ((1 « (32 - 24)) - 1) je `11111111.11111111.11111111.00000000`, čo je v decimálnom formáte maska siete 255.255.255.0, čo zodpovedá /24 prefixu.

Potom pomocou bitovej operácie `and` medzi prefixom a maskou získame adresu siete a tak isto pomocou operácie `or` medzi prefixom a inverzom masky získame broadcastovú adresu. Adresu siete ani broadcastovú nepočítame, nakoniec porovnáme `and` operácie medzi yiaddr/ciaddr IP s maskou siete a prefix IP s maskou siete. Ak sa rovnajú, yiaddr/ciaddr IP patrí do prefixu.

```
bool Prefix::ip_belongs(const std::string ip)
{
    uint32_t ip_num = Prefix::ip_to_int(ip);
    uint32_t prefix_num = Prefix::ip_to_int(this->ip_address);
    uint32_t mask = ~((1 << (32 - this->prefix_length)) - 1); // Calculate netmask
    uint32_t network_address = prefix_num & mask;
    uint32_t broadcast_address = network_address | ~mask;

    if (ip_num == network_address || ip_num == broadcast_address)
    {
        return false; // The IP is either the network or broadcast address
    }

    return (ip_num & mask) == (prefix_num & mask) ? true : false; // Check if IPs match in the prefix
}
```

5 Testovanie aplikácie

Pri testovaní aplikácie bolo využitých viacero nástrojov na simuláciu DHCP komunikácie a overenie správnosti implementácie.

5.1 Scapy

Scapy je knižnica a v jazyku Python, ktorá umožňuje vytváranie, odosielanie, zachytávanie a analýzu sieťovej prevádzky na nízkej úrovni. Pri testovaní bola použitá knižnica Scapy na generovanie a odosielanie špecifických DHCP paketov na testovanie rôznych scenárov DHCP komunikácie.

5.2 Analýza pcap súborov

Ďalšou dôležitou súčasťou testovania bolo použitie pcap súborov. Vďaka pcap súborom bolo možné otestovať zadanú funkcionalitu aplikácie a to čítanie z pcap súboru. Pcap (Packet Capture) súbory obsahujú dáta zachytené pri monitorovaní sieťovej komunikácie. Pcap súbory boli generované pomocou Wiresharku a knižnice scapy.

5.3 TCP Replay

TCP Replay je nástroj, ktorý umožňuje prehrávanie zachytených sieťových dát. Pomocou tohto nástroja je možné "prehrávať"zachytenú sieťovú komunikáciu z pcap súboru a pozorovať, ako sa aplikácia správa pri spracovaní paketov v reálnom čase.

6 Informácie o programe

Program `dhcp-stats` je nástroj navrhnutý na monitorovanie a analýzu komunikácie DHCP (Dynamic Host Configuration Protocol) v počítačových sieťach. Jeho hlavným cieľom je sledovať pridelené IP adresy, zisťovať ich využitie v rámci definovaných IP prefixov a upozorňovať na prípadné problémy, ako je nadmerné využitie dostupných adries v jednom z prefixov. Je napísaný v jazyku `C++`. Je navrhnutý pre Linux zariadenia.

6.1 Funkcionalita

- Zachytáva DHCP pakety na sieťovom rozhraní alebo analyzuje pcap súbory obsahujúce zachytenú sieťovú komunikáciu.
- Sleduje pridelené IP adresy a ich využitie v rámci definovaných IP prefixov.
- Upozorňuje na nadmerné využitie adries v prefixoch a zaznamenáva chyby v systémovom logu (syslog).
- Poskytuje prehľadné textové rozhranie pomocou knižnice `ncurses`, ktoré zobrazuje aktuálne štatistiky.

7 Návod

7.1 Inštalácia

Program `dhcp-stats` nevyžaduje zložitú inštaláciu. Stačí stiahnuť zdrojový kód a skompilovať ho na svojom systéme s použitím príkazu `make`. Pred spustením si overte, že máte nainštalované potrebné knižnice, ako je `pcap` a `ncurses`. Taktiež na sledovanie živého rozhrania je zapotreby mať príslušné práva.

7.2 Spustenie

Program môžete spustiť s rôznymi argumentmi, ktoré určujú, či bude sledovať sieťové rozhranie naživo, analyzovať pcap súbor alebo vypísať pomoc. Napríklad:

- `./dhcp-stats -i <rozhranie> <ip-prefix>` - Sledovanie živého rozhrania.
- `./dhcp-stats -r <súbor.pcap> <ip-prefix>` - Analýza pcap súboru.
- `./dhcp-stats -h` - výpis pomoci.

Taktiež je možné spustiť manuálovú stránku pomocou:

- `man dhcp-stats` - Výpis manuálu.

7.3 Príklady použitia

Príklad 1: Sledovanie sieťového rozhrania `eth0` s prefixom `192.168.1.0/24`

- `./dhcp-stats -i eth0 192.168.1.0/24`

Výstup 1:

```
IP-Prefix Max-hosts Allocated addresses Utilization
192.168.1.0/24 254 50 19.69%
Press CTRL-C to exit...
```

Príklad 2: Analýza pcap súboru `multiple.pcap` s prefixmi `192.168.1.0/24 192.168.2.0/25 192.168.3.0/30`

- `./dhcp-stats -r pcap_files/multiple.pcap 192.168.1.0/24 192.168.2.0/25 192.168.3.0/30`

Výstup 2:

```
IP-Prefix Max-hosts Allocated addresses Utilization
192.168.1.0/24 254 105 41.34%
192.168.2.0/25 126 80 63.49% exceeded 50% of allocations!
192.168.3.0/30 2 1 50.00%
Press CTRL-C to exit...
```

Bibliografia

[1] DROMS, R.: Dynamic Host Configuration Protocol. online, 1997, [vid. 2023-10-21]. Dostupné z: `https://www.rfc-editor.org/info/rfc2131`

[2] MATOUŠEK, P.: *Síťové aplikace a jejich architektura*, kapitola 1. Brno: Nakladatelství VUTIUM, první vydání, 2014, ISBN 978-80-214-3766-1, s. 41 – 44.