ISA 2023- Monitorování DHCP komunikace

Ondřej Lukášek (xlukas15)

Obsah

Uvedení do problematiky

Dynamic Host Configuration Protocol (DHCP) je klíčový protokol používaný v moderních IP sítích k automatizaci procesu konfigurace síťových zařízení. Jeho hlavním účelem je usnadnit správu IP adres a jiných konfiguračních detailů pro zařízení připojená k síti. Sledováním DHCP komunikace lze tedy předcházet zaplnění adres v síti. Právě k prevenci zaplnění sítě by mohl pomoct tento projekt.

Komunikace se skládá ze čtyř základních fází.

Fáze Discover

Když se klient připojí k síti, nemá typicky přidělenou žádnou IP adresu. Aby získal adresu, začne proces odesíláním *DHCP DISCOVER* zprávy na síť. Tato zpráva je vysílána jako broadcast, což znamená, že ji přijmou všechny zařízení na lokální síti. Cílem je zjistit, zda je v síti přítomen DHCP server, který může přidělit IP adresu.

Fáze Offer

DHCP server, který přijme *DHCP DISCOVER* zprávu, odpoví zprávou *DHCP OFFER*. Tato zpráva obsahuje nabídku pro konfiguraci, včetně IP adresy, masky podsítě, výchozí brány a dalších informací, jako jsou adresy DNS serverů. Nabídka je poslána zpět na síť, přičemž obsahuje MAC adresu klienta, který požádal o konfiguraci.

Fáze Request

Po přijetí *DHCP OFFER* od serveru klient vyhodnotí nabídku z předchozí fáze. Pokud klient akceptuje nabídku, pošle *DHCP REQUEST* zprávu zpět serveru. Tímto krokem klient žádá o možnost použít konfigurační detaily obsažené v nabídce. Tato zpráva je také obvykle vysílána jako broadcast.

Fáze Acknowledgement

Po obdržení *DHCP REQUEST* zprávy, DHCP server odesílá *DHCP ACK* zprávu klientovi. Tato zpráva potvrzuje, že klient má nyní oprávnění používat přidělené parametry, a také obsahuje délku platnosti (lease time), což je doba, po kterou může klient parametry používat. Pokud by server nemohl potvrdit žádost, místo toho by odeslal *DHCP NAK* [1]zprávu, odmítající žádost klienta.

Základní informace o programu

Program *dhcp-stats* sleduje DHCP komunikaci v síti a tvoří si statistiky o počtu připojených zařízení. Ze čtyř zmíněných základních fází jej zajímá zpráva *DHCP ACK*, u které je jisté, že došlo k úspěšnému připojení daného zařízení k síti. Zároveň tato zpráva obsahuje všechny potřebné informace. Těmi jsou jaká adresa se obsadila, na jak dlouho se obsadila neboli také lease time.

Zároveň se ale také zajímám o zatím nezmíněnou zprávu *DHCP RELEASE*, která mi umožňuje zařízení ze statistiky zase vymazat.

Můj projekt tedy monitoruje počet zařízení v síti, které se připojí, od jeho spuštění, ale také ze statistik mazat zařízení, kterým vyprchal jejich lease time (doba, po kterou má zařízení „propůjčenou“ adresu) nebo která odeslala zprávu *DHCP RELEASE*.

[1] S touto zprávou jsem se setkával velmi často, když jsem poslouchal na síti se dvěma DHCP servery. Po odeslání *DHCP REQUEST* žádosti klienta odpověděl jeden server zprávou *DHCP ACK* a druhý zprávou *DHCP NAK*.

Popis implementace

Celý program je naprogramován primárně za pomocí knihovny pcap[2]. Ta sloužila ke splnění jádra funkcionality programu, tedy k poslouchání DHCP komunikace (pomocí funkce `pcap_loop()`) ať už v reálném čase na nějakém rozhraní nebo skrze přidělený soubor jako argument při spuštění.

Čtení ze souboru se vykonává pomocí `pcap_open_offline()`, čtení z rozhraní se provádí pomocí `pcap_open_live()`. Pro uzavření komunikace je využita funkce `pcap_close()`.

Mezi další používané knihovny patří třeba syslog[3], která mi umožňovala zapisovat do syslogu informace o přesažení 50% obsazenosti prefixu. Také byla využita knihovna ncurses[4], pomocí které je udělaný konzolový výpis programu. Díky ní je výstup o něco přehlednější a lépe zpracovaný.

Jako ostatní byly využity knihovny string (práce s řetězci), netinet (pomáhala s IP adresami), time (pro práci s časem pro uvolňování IP adres) a další standardní knihovny.

Program také používá struktury, které ulehčují jeho přehlednost. Mezi tyto struktury patří `dhcp_packet`, který mi sloužil k rozdělení DHCP packetu na jeho podčásti[5]. Mezi další struktury patří `IP_Prefix`, do které si zapisuji, kolik zařízení je k nějakému IP prefixu připojeno. Déle struktura `IP_Prefixes`, do které jsem si ukládal všechny IP prefixy, o kterých jsem potřeboval získávat informace.

Informace o prefixech si ukládám do seznamu, jehož implementaci lze najít v souboru (*listfunc.c*). Obsahuje naprosto standardní funkce, jakými jsou třeba přidání prvku do seznamu, smazání prvku ze seznamu, nalezení prvku v seznamu a podobné.

Průběh programu

Program začíná tím, že se zpracují argumenty příkazové řádky. Mezi povolené argumenty patří pouze ty, které jsou uvedeny v zadání[6]. Argument pro výpis nápovědy (zpravidla -h) tedy podporován není.

Pro případnou nápovědu slouží primárně manuálová stránka, která je součástí odevzdaného archivu. Při vytváření manuálové stránky jsem se snažil postupovat podle zažitých standardů, odrážel jsem se tedy od toho, jak jsou koncipovány manuály jiných knihoven (například knihovny ssh a jiných). Syntaxi manuálové stránky jsem se snažil držet s webovou stránkou[7], která byla doporučená na použití v zadání projektu.

Po zpracování argumentů se zadané IP prefixy vloží do seznamu a následně se postupuje podle toho, jaké byly vloženy argumenty. Mohou nastat 3 platné kombinace argumentů.

1. Interface + pcap soubor,
2. Samotný interface,
3. Samotný pcap soubor.

V případě, že dojde k první kombinaci, dojde nejprve k přečtení dat z pcap souboru a následně bude probíhat poslouchání DHCP komunikace v reálném čase až do přerušení programu pomocí

[2] Dokumentace knihovny je dostupná na https://www.tcpdump.org/manpages/pcap.3pcap.html.
[3] Popis knihovny dostupný na https://www.gnu.org/software/libc/manual/html_node/Syslog.html.
[4] Manuál dostupný na https://tldp.org/HOWTO/NCURSES-Programming-HOWTO/.
[5] Ty jsou vyhovující protokolu RFC 2131. Popis je dostupný například na https://datatracker.ietf.org/doc/html/rfc2131.
[6] Zadání projektu je dostupné na https://www.vut.cz/studis/student.phtml?script_name=zadani_detail&apid=268266&zid=54265.
[7] Manpage manuál dostupný na https://liw.fi/manpages/.

kombinace *Ctrl + C*. Před samotným vypisování statistik na výstup dojde k otevření *ncurses* okna funkcí `initscr()`.

Při druhé kombinaci se začne rovnou poslouchat na zadaném rozhraní DHCP komunikace. Statistiky se budou při přicházejících paketech v reálném čase přepisovat.

Ve třetí kombinaci dojde k přečtení a vygenerování statistik z odkazovaného souboru v argumentu. Čtení ze souboru není nijak zpomalené, takže i při velkém souboru jsou průběžné změny statistik prakticky nepostřehnutelné. Po přečtení celého souboru dojde k uzavření okna funkcí `endwin()` a ukončení běhu programu. Díky zavření okna ncurses a rychlému proběhnutí celého programu není vidět vlastní běh programu. Ten lze zpomalit změnou konstanty `PROGRESS_DELAY` a běh programu tak „rozanimovat“.

O čtení DHCP paketů se stará funkce `packet_handler()`, která zjistí, o jaký paket se jedná (zajímá nás totiž pouze DHCP ACK a DHCP RELEASE paket). Pokud se IP adresu zařízení nepodařilo najít v seznamu, vytvoří se nový prvek. Pokud se ji najít podaří, pouze se aktualizují údaje o něm (lease time). Nakonec funkce aktualizuje počet zařízení (`update_dev_count()`), která spadají do prefixu a změnu vypíše na výstup. Tato funkce také provede kontrolu všech IP adres v seznamu. Pokud nějaké vypršel *lease time*, pak je ze seznamu odstraněna. V případě načítání pcap souboru, je interní čas aktualizován ze zachycených packetů. Odstranit zařízení ze seznamu tak lze na základě jakéhokoli příchozího *ACK* paketu, kdy je kontrolován lease time již přiřazených IP adres, nebo také na základě *DHCP RELEASE* paketu. Poté bude zařízení také odstraněno.

Ve funkci `update_dev_count()` se také provede kontrola, jestli počet zařízení v prefixu nepřesahuje 50% kapacity prefixu. Pokud ano, potom se při každém připojení zařízení, ve kterém bude počet zařízení v prefixu větší než 50 % vypíše zpráva *prefix <prefix> exceeded 50 % of allocations*. Dále se zde provádí kontrola, jestli vůbec IP adresa, která je obsažena v DHCP paketu, spadá do nějakého ze zadaných prefixů. To se vykoná funkcí `is_ip_in_prefix()`. IP adresy mimo zadané prefixy jsou ignorovány

Návod na použití

Prerekvizitní knihovny

- Knihovna *libpcap*
  pro stažení: `sudo apt-get install libpcap-dev`
- Knihovna *ncurses*
  pro stažení: `sudo apt-get install libncurses5-dev libncursesw5-dev`

Kompilace a spuštění

Program lze zkompilovat pomocí *Makefile*, který je součástí odevzdaného archivu, příkazem `make` v příkazové řádce v adresáři, kde se rozbalený obsah archivu nachází.

Přeložený binární soubor lze vymazat pomocí příkazu `make clean`.

Pokud máte zájem o to se podívat na nějaké příklady případně anglický manuál k tomu, jak program používat, můžete využít manpage (*dhcp-stats.1*). Tu lze otevřít příkazem `man ./dhcp-stats.1`. Manpage obsahuje stručný popis aplikace, způsob použití aplikace, příklady, jak lze aplikaci spustit a referenční odkazy. Manuálová stránka je napsána v anglickém jazyce.

Po úspěšném přeložení aplikace ji lze spustit. Aplikaci při čtení ze síťového rozhraní nelze spustit jenom příkazem `./dhcp-stats` ze dvou důvodů. Jedním z nich je ten, že musí být spuštěna

s oprávněním správce (na systému Linux tedy pomocí `sudo`). Je to z důvodu využití *pcap* knihovny, která oprávnění správce vyžaduje.

Druhý důvod je, že program také vyžaduje zadaný alespoň jeden argument, tedy `-i` nebo `-r`, případně i kombinaci obou. Zároveň program pro spuštění vyžaduje alespoň jeden IP prefix zadaný ve formátu, který více specifikuje zadání (odkaz v poznámce pod čarou v předchozí kapitole).

Způsob chování programu při různě zadaných kombinacích argumentu jsou popsány v kapitole „*Průběh programu*“.

Upozornění: Program byl testován na operačním systému Linux - Ubuntu 22.04. Kompilace testována také na serveru Merlin.

Praktické využití programu

Program vyniká svou praktičností zejména v situaci, kdy je potřeba v reálném čase kontrolovat množství zařízení v síti. Nejvíce užitečný by mohl být na velkých firemních sítích, kde by se mohlo snadněji stát, k přeplnění kapacity sítě v důsledku velkého množství připojovaných zařízení.
V prevenci přeplnění sítě pomáhá zapisování do *syslogu*, které probíhá vždy, když se připojí nové zařízení a zároveň je při tom zaplněnost sítě (podle zadaných prefixů) větší než 50 %. V případě, že by uživateli nevyhovovala prahová hodnota 50 %, lze program dále snadno upravit přepsáním dělitele (nyní nastaveno na 2 (50 %)).

Také může sloužit pro lepší vizualizaci toho, kdy je se k síti připojuje nejvíce zařízení. Při nastavení definované konstanty `PROGRESS_DELAY` z aktuální hodnoty 0 na hodnotu například 10000 (zpoždění běhové smyčky v mikrosekundách), dojde ke zpomalení průběhu programu v případě čtení ze souboru a program při svém běhu trochu připomíná animaci.

Další velikou výhodou programu je to, že podporuje i odpojení zařízení na základě vypršení *lease time* nebo na základě zprávy *DHCP RELEASE*. Program tedy může konstantě běžet v reálném čase bez nutnosti ho kdykoliv restartovat.

Zajímavé části kódu

Zpracování DHCP zpráv

```
while (*dhcp_options != 255) {  // End option

    switch (*dhcp_options) {
        case DHCP_OPTION_MESSAGE_TYPE: {
            u_char dhcp_type = *(dhcp_options + 2);
            if (dhcp_type == DHCP_ACK)
                wasAck = 1;
            }
            if (dhcp_type == DHCP_RELEASE) {
                removeElement(dhcp_pkt->ciaddr);
                update_dev_count(head, &ip_prefixes);
                print_ip_ranges(&ip_prefixes);
            }
            break;
        }
        default:
            break;
    }

    dhcp_options += *(dhcp_options + 1) + 2;
}
```

Obrázek 1 - Zpracování DHCP zpráv.

Zde se zabývám zachytáváním typů DHCP zpráv, se kterými následně pracuje.

Pokud přijde zpráva ACK, potom se nastaví příznak, že zpráva, která přišla, byla typu ACK a bude potřeba později volat funkci pro vyhledání nebo případné přidání adresy do seznamu s adresami.

Pokud příchozí zpráva byla typu RELEASE, potom dojde k jejímu odstranění ze seznamu IP adres a díky tomu také musí dojít k obnovení počtu připojených zařízení a opětovnému vypsání statistik na výstup.

Přidávání adres do seznamu

```
if(wasAck){
    if (findElement(dhcp_pkt->yiaddr) != NULL) {
        updateElement(dhcp_pkt->yiaddr, pkthdr->ts.tv_sec + lease_time);
    }
    else {
        addElement(dhcp_pkt->yiaddr, pkthdr->ts.tv_sec + lease_time);
    }
```

Obrázek 2 - Přidávání adres do seznamu.

Tato část kódu je spojená právě s předchozí zmíněnou částí, protože se stará o hledání adres v seznamu nebo jejich případnému přidání, pakliže vyhledání adresy proběhlo neúspěšně. Důležitá je část kvůli tomu, že díky tomu nedochází k neustálému přidávání IP adres do seznamu.

Rozbor struktury paketů

```
struct ip* ip_header = (struct ip*)(packet + 14);
struct udphdr* udp_header = (struct udphdr*)(packet + 14 + (ip_header->ip_hl << 2));
struct dhcp_packet* dhcp_pkt = (struct dhcp_packet*)(packet + 14 + (ip_header->ip_hl << 2) + sizeof(struct udphdr));
```

Obrázek 3 - Rozbor struktury paketů.

Tato část kódu je zaměřena na extrakci hlaviček ze síťových paketů pro další zpracování.

Nejprve se získává IP hlavička, která je umístěna po Ethernetové hlavičce (14 bytů od začátku paketu). Následně se extrahuje UDP hlavička, jejíž pozice je závislá na délce IP hlavičky.

Nakonec se identifikuje a izolují vlastní DHCP informace, které následují po UDP hlavičce.

Zápis do syslogu

```
for (int i = 0; i < ip_prefixes->count; i++) {
    int max_devs = (1 << (32 - ip_prefixes->prefixes[i].prefix)) - 2;
    int dev_count = ip_prefixes->prefixes[i].dev_count;
    char ip_str[INET_ADDRSTRLEN];
    inet_ntop(AF_INET, &ip_prefixes->prefixes[i].ip, ip_str, INET_ADDRSTRLEN);

    if (dev_count > max_devs / 2) {
        char msg[255];
        sprintf(msg, "prefix %s/%d exceeded 50%% of allocations", ip_str, ip_prefixes->prefixes[i].prefix);
        move(ip_prefixes->count+i, 0);
        printw("%s\n", msg);
        refresh();
        syslog(LOG_WARNING, "%s", msg);
    } else {
        move(ip_prefixes->count+i, 0);
        printw("%78s", " ");

    }
}
```

Obrázek 4 - Zápis do syslogu.

Ve funkci `update_dev_count()` dochází k zapisování do syslogu, podle specifikace. Zapisování do syslogu proběhne vždy, když dojde k připojení zařízení, u kterého bude zaplněnost prostoru více než 50 %. To způsobuje velké množství docházejících zpráv, nicméně pak s menší pravděpodobností dojde k tomu, že bude uživatelem zápis do syslogu ignorován.

Zároveň se dá také tato hranice 50 % posunout výše v podmínce, kontrolující obsazenost.

Přehled literatury

- Knihovna LIBPCAP (https://www.tcpdump.org/manpages/pcap.3pcap.html)
- Knihovna Syslog (https://www.gnu.org/software/libc/manual/html_node/Syslog.html)
- Manuál manpages (https://liw.fi/manpages/)
- Manuál ncurses (https://tldp.org/HOWTO/NCURSES-Programming-HOWTO/)
- RFC2131 (https://datatracker.ietf.org/doc/html/rfc2131)