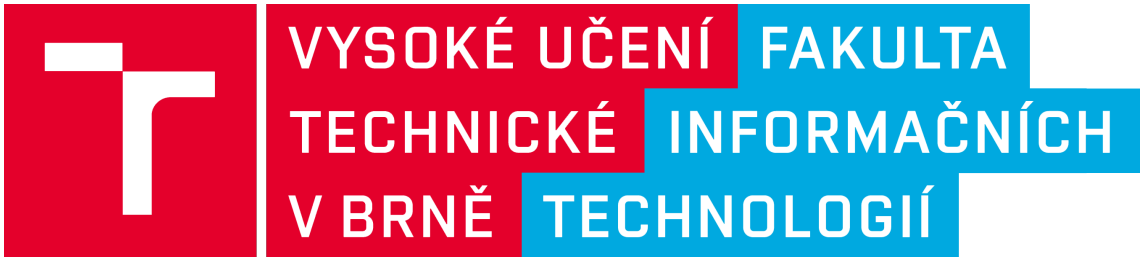


Síťové aplikace a správa sítí

Monitorování DHCP komunikace

Projektová dokumentace

říjen - listopad 2023

Veronika Jirmusová - xjirmu00

Obsah

1 Úvod

Tento program byl vytvořen s cílem poskytnout administrátorovi síťové statistiky o vytížení síťových prefixů z pohledu alokovaných IP adres. Program sleduje DHCP provoz na zadaném rozhraní nebo zpracovává pcap soubory a generuje statistiku vytížení síťových prefixů, které jsou specifikovány v příkazové řádce.

2 Teorie monitorování DHCP

DHCP je zkratka anglického názvu Dynamic Host Configuration Protocol a značí síťový protokol, který slouží k dynamickému přidělování síťových konfiguračních parametrů zařízením v počítačové síti. Mezi tyto parametry patří IP adresa, maska podsítě, brána, DNS servery a další informace potřebné pro správnou komunikaci v síti. Obecně se uvádí čtyři základní fáze, které klient a DHCP server absolvují během procesu získávání IP adresy a dalších informací. Mezi tyto fáze patří: [4]

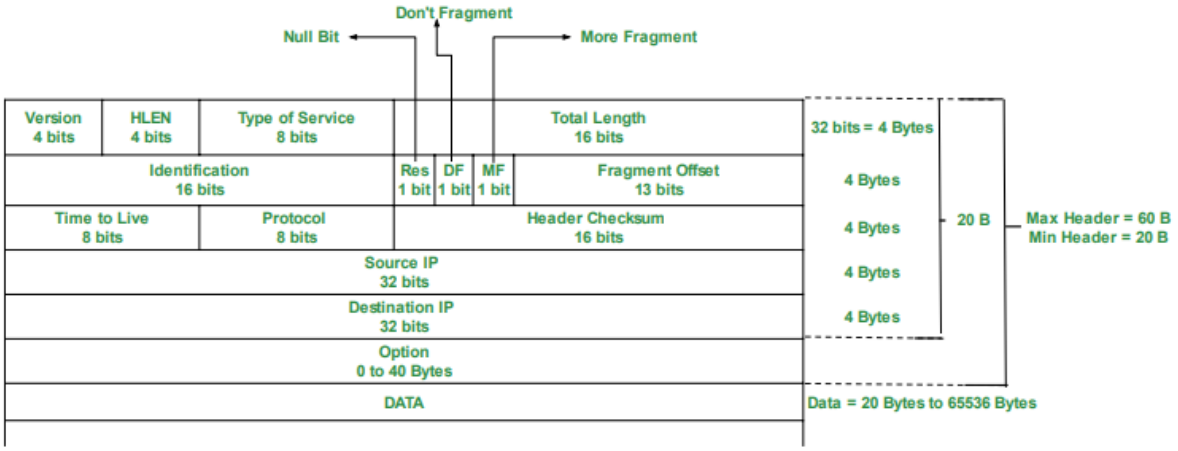
- **Discover**
  Klient v této fázi odesílá do sítě broadcast zprávu s žádostí o konfigurační informace s cílem najít v síti dostupné DHCP servery.
- **Offer**
  DHCP servery, které obdrží broadcast zprávu, reagují nabídkou konfiguračních informací. Každý server může nabídnout klientovi různé konfigurační parametry, včetně IP adresy.
- **Request**
  Klient odesílá zpět požadavek na přidělení konkrétní konfigurace, většinou té s nejnižší nabízenou IP adresou, a informuje ostatní DHCP servery o přijetí žádosti.
- **Acknowledge**
  DHCP server, který obdržel požadavek, potvrdí klientovi přidělení konkrétní konfigurace, včetně IP adresy a dalších informací a klient příjmá přidělenou konfiguraci, kterou může aktivovat pro svoji síťovou komunikaci.

DHCP tedy hraje velice důležitou roli v usnadnění a optimalizaci správy IP adres a síťových konfigurací ve velkých nebo dynamických sítích. DHCP umožňuje dynamické přidělování IP adres, Konfiguraci sítě, správu přenosných zařízení, monitoring a statistiky a řadu dalších důležitých akcí, bez kterých bychom se v dnešní době jen těžko obešli.

2.1 IPv4 datagram

Při zachycování packetů musel program najít požadovanou IP adresu a další potřebné informace, jako třeba adresu sítě, broadcast adresu, prefix a další. Pro takové případy existuje pevný datagram IPv4. [2]

Obrázek 1 *IPv4 datagram,*
dostupné z: https://www.geeksforgeeks.org/introduction-and-ipv4-datagram-header/



Hlavička IPv4 může mít 20 až 60 bytů a zřejmě z ní lze bitovými posuvy získat potřebné informace ke zpracování.

3 Implementace

Program sestává celkem ze 3 základních .cpp souborů *parser.cpp, argcheck.cpp a pcap.cpp* a je pojat objektově. Každý z těchto souborů má k sobě přiřazený hlavičkový soubor. Dále se v programu vyskytuje soubor *main.cpp* a $Makefile$. Program počítá pouze s protokoly UDP.

Program pracuje s příkazovou řádkou a očekává následující vstupy:
$./dhcp - stats[-r < filename >][-i < interface - name >] < ip - prefix > [< ip - prefix > [...]]$,
kde:

- **-r** $< filename >$**:** Statistika bude vytvořena z pcap souborů.
- **-i** $< interface >$**:** Rozhraní, na kterém může program naslouchat.
- $< ip - prefix >$**:** Rozsah sítě pro které se bude generovat statistika.

Soubor main.cpp tedy po spuštění programu zavolá funkci pro ověření a zpracování argumentů. Třída ArgCheck zajišťuje práci s argumenty. Nejprve je zkontrolováno použití flagů -r a -i, přičemž validním vstupem je použití pouze jednoho flagu. Po této kontrole následuje validace zadaných prefixů pomocí regulárního výrazu.

Po zpracování vstupu následuje parsing IP adres. Z nalezených hlaviček program zjišťuje masku, broadcast adresu, maximum adres v subnetu a další a ukládá získané informace pomocí struktury $parser\_t$ do vektoru *prefixes*, z důvodu ulehčení práce s pamětí.

Na základě vstupních parametrů se vyhodnotí další postup programu. Pokud je potřeba, zpracuje se pcap soubor a program pracuje s daty v souboru, jinak začne zachytávat pomocí funkce *pcap_open_live* packety pro následné zpracování a generaci statistiky síťového vytížení.

Program generuje pomocí funkce *mvprintw* výslednou statistiku vytížení získanou ze souboru nebo se dynamicky mění podle počtu zrovna zachycených packetů. Pokud vytížení přesáhne 50 %, zapíše se do systémového logu hláška o této skutečnosti.

Program je psán anglicky a obsahuje anglické komentáře. Krom klasických systémových a řetězových knihoven využívá síťové knihovny a pro práci s terminálem využívá ncurses.h.

4 Zajímavé pasáže

Ačkoliv je to lehce úsměvné, autorka je nejvíce pyšná na vymyšlení funkčního regulárního výrazu, který se nachází v souboru *argcheck.cpp* a kontroluje korektnost vstupních adres.

Dále je zajímavá celková komunikace všech částí programu, protože je program řešen objektově, aby se ke všem informacím získaných z IP parsingu mohly dostat všechny ostatní funkce.

Při testování byla objevena chyba při výpisu. Pokud byl použit pcap soubor, který sestával pouze z *offer* zpráv, tedy žádné *ACK* zprávy, program nevypisoval inicializační hlášku *IP-Prefix Max-hosts Allocated addresses Utilization.* Tato situace byla po dlouhé periodě testování jednotlivých částí a těžkopádném se dobírání k řešení vyřešena přidáním funkce *ConsoleAccess* do třídy *IpParser*, která se stará o výpis mimo veškeré dění programu. Tato funkce problém vyřešila a výpis je proveden i po přijetí nulového počtu ACK zpráv.

V neposlední řadě je (z mého pohledu) elegantně a přehledně řešeno "posouvání se" IPv4 adresou pomocí funkce *PcapGet* ve třídě *PcapParse*. Tato funkce se po zavolání prvního výpisu posouvá skrz daný datagram, získává informace o proměnných délkách zpráv a hledá zprávu ACK.

5 Návod k použití

Jako každý program psaný v jazyce C++ je nutné jej nejprve přeložit, což je možné zadáním příkazu *make* do terminálu. Soubor Makefile zajistí přeložení všech nutných souborů a poté smazání binárních souborů. Po přeložení se program spouští pomocí příkazu *./dhcp-stats* a zadáním potřebných argumentů. Program také umí zpracovat argument $-h$ a odpovídá na něj výpisem korektního vstupu.

6 Studium a literatura

Studium probíhalo výhradně z poskytnutých materiálů, nejvíce jsem pak čerpala z opory předmětu. Pro konkrétní studium DHCP byla využita webová stránka dostupná z:
https://datatracker.ietf.org/doc/html/rfc2131 [4]. Ke studiu UDP protokolu byla využitá stránka dostupná z:
https://datatracker.ietf.org/doc/html/rfc768 [1].
Ke správnému pochopení práce s IP adresami byly využity následující stránky:
https://datatracker.ietf.org/doc/html/rfc791 [2], https://datatracker.ietf.org/doc/html/rfc1042 [3].

Referenční literatura

[1] User Datagram Protocol. RFC 768, Srpen 1980, doi:10.17487/RFC0768. Dostupné z: https://www.rfc-editor.org/info/rfc768

[2] Internet Protocol. RFC 791, Září 1981, doi:10.17487/RFC0791. Dostupné z: https://www.rfc-editor.org/info/rfc791

[3] Standard for the transmission of IP datagrams over IEEE 802 networks. RFC 1042, Únor 1988, doi:10.17487/RFC1042. Dostupné z: https://www.rfc-editor.org/info/rfc1042

[4] Droms, R.: Dynamic Host Configuration Protocol. RFC 2131, Březen 1997, doi:10.17487/RFC2131. Dostupné z: https://www.rfc-editor.org/info/rfc2131