



VYSOKÉ UČENIE TECHNICKÉ V BRNE
FAKULTA INFORMAČNÝCH TECHNOLOGIÍ

Sieťové aplikácie a správa sietí – Dokumentácia k projektu

DHCPv6 relay
s podporou vloženia MAC adresy

Autor:
Peter KRUTÝ *xkruty00*

18.novembra 2019

Obsah

1	Úvod	2
2	Problematika DHCP	2
2.1	Odlišnosti DHCPv6	2
2.2	Princíp komunikácie DHCPv6	2
2.2.1	Stavový DHCPv6	3
2.3	DHCPv6 relay	3
2.4	Formát a typy správ DHCPv6	3
3	Návrh a implementácia programu	5
3.1	Hlavný program	5
3.2	ArgParser	5
3.3	Logger	5
3.4	PacketHandler	6
3.5	Ostatné súbory	6
4	Testovanie programu	6
4.1	Simulácia servera	6
4.2	Simulácia klienta	7
4.3	Testovanie relayu	7
5	Použitie programu	7
5.1	Preklad	7
5.2	Spustenie	7
5.3	Možnosti spustenia	7
5.4	Ukážka spustenia programu a následného výstupu	8
6	Záver	8
	Literatúra	9

1 Úvod

Cieľom projektu bolo vytvoriť DHCPv6 relay v jazyku C++, ktorého chovanie je definované podľa RFC 8145 [3]. Program navyše dokáže vložiť MAC adresu klienta do DHCPv6 správ, tak ako definuje RFC 6939 [1].

Funkčnosť programu spočíva v odchyťovaní DHCPv6 žiadostí o adresu/prefix a následnom preposielaní na používateľom špecifikovaný server. Odchyťovanie je možné realizovať na jednom rozhraní, ktoré je explicitne zadané používateľom, alebo na všetkých, v prípade, že používateľ žiadne nezadal. Relay ďalej umožňuje logovať informácie pomocou protokolu syslog alebo výpisov na štandardný výstup, podľa toho ako špecifikuje používateľ.

2 Problematika DHCP

Dynamic Host Configuration Protocol (DHCP) je protokol z rodiny TCP/IP, ktorý sa využíva pre automatickú konfiguráciu počítačov pripojených do siete. Komunikácia je zpravidla realizovaná medzi klientom a serverom, pričom iniciátorom je vždy klient. DHCP server prideluje klientovi hneď niekoľko konfiguračných nastavení, ako napríklad IP adresu, prefix/masku siete, implicitnú bránu, adresa DNS serveru, ale aj omnoho viac. Záleží podľa konkrétnej špecifikácie/implementácie. Protokol beží na UDP portoch 67 a 68, klient komunikuje na porte 68 a server naslúcha na porte 67.

V dnešnej dobe si existenciu počítačových sietí bez protokolu DHCP vieme asi len ťažko predstaviť. Protokol významným spôsobom zjednodušuje a centralizuje správu počítačovej siete.

2.1 Odlišnosti DHCPv6

Úlohou projektu bolo implementovať relay, ktorý pracuje na sieťovej vrstve IPv6. A tak je teda potrebné aby komunikácia prebiehala podľa pravidiel protokolu DHCPv6. Protokoly DHCP a DHCPv6 majú mnoho vlastností spoločných, avšak je ich aj mnoho odlišných a táto podkapitola je mierená práve na odlišnosti.

Nasledujúci výpis zahŕňa iba tie najhlavnejšie rozdiely:

- DHCPv6 využíva k identifikácii klienta DUID, kým DHCP MAC adresu
- Protokoly využívajú odlišné názvy typov správ
- Pri DHCPv6 klient posiela správy na multicast adresu, kým pri DHCP sa správy posielajú na broadcast adresu
- Čísla portov pre DHCPv6 sú odlišné, klient naslúcha na porte 546, server a relay naslúcha na porte 547
- ...

2.2 Princíp komunikácie DHCPv6

Dynamické pridelenie adresy pomocou protokolu DHCPv6 môže byť realizované viacerými spôsobmi, konkrétne sa jedná o 3 rozdielne metódy a to SLAAC, bezstavový DHCPv6 a stavový

DHCPv6. Bližšie bude popísaný iba stavový DHCPv6, nakoľko sa implementácia relayu týka práve tejto metódy.

2.2.1 Stavový DHCPv6

Kroky v stavovej DHCPv6 komunikácii sú nasledujúce:

1. Klient pošle správu ICMPv6 Router Solicitation, ktorej význam spočíva v tom, že dá lokálnemu routeru vedieť o svojej prítomnosti v sieti.
2. Router odpovie správou ICMPv6 Router Advertisement s príznakom 'M', ktorý značí to, že pôjde o stavový typ DHCPv6 komunikácie s využitím DHCPv6 serveru.
3. Klient pošle DHCPv6 správu Solicit, ktorou lokalizuje server.
4. Server odpovie DHCPv6 správou Advertise, ktorou informuje o tom, že je dostupný.
5. Klient pošle DHCPv6 správu Request, ktorou požiada o IPv6 adresu s ďalšími dodatočnými informáciami.
6. Server odpovie DHCPv6 správou Reply, ktorou klientovi pridelí IPv6 adresu s ďalšími dodatočnými informáciami.

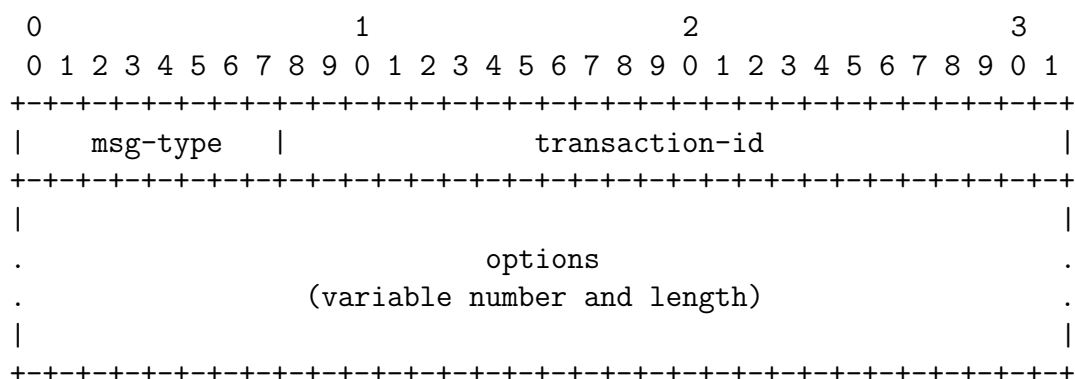
2.3 DHCPv6 relay

DHCP relay je program/služba, ktorej úlohou je preposielanie správ medzi klientom a serverom. V prípade, že sa server nachádza v rovnakej lokálnej sieti ako klient, tak nemá žiadne opodstatnenie. Jeho prítomnosť je požadovaná až v prípade, keď je server umiestnený mimo lokálnu sieť. Prečo je to tak? Cieľová adresa správ, ktoré posielajú klient je ff02::1:2, čo znamená adresa multicast skupiny DHCPv6. Router však takýto typ správ nemôže posielajú mimo lokálnu sieť. Je teda potrebné aby sa v lokálnej sieti nachádzal prvok, ktorý odchytlí multicast správu a pošle ju na server ako unicast. Týmto prvkom je práve DHCPv6 relay. Jeho podrobnejšia funkčnosť a práca nad preposielanými správami je popísaná v RFC 8145 [3].

2.4 Formát a typy správ DHCPv6

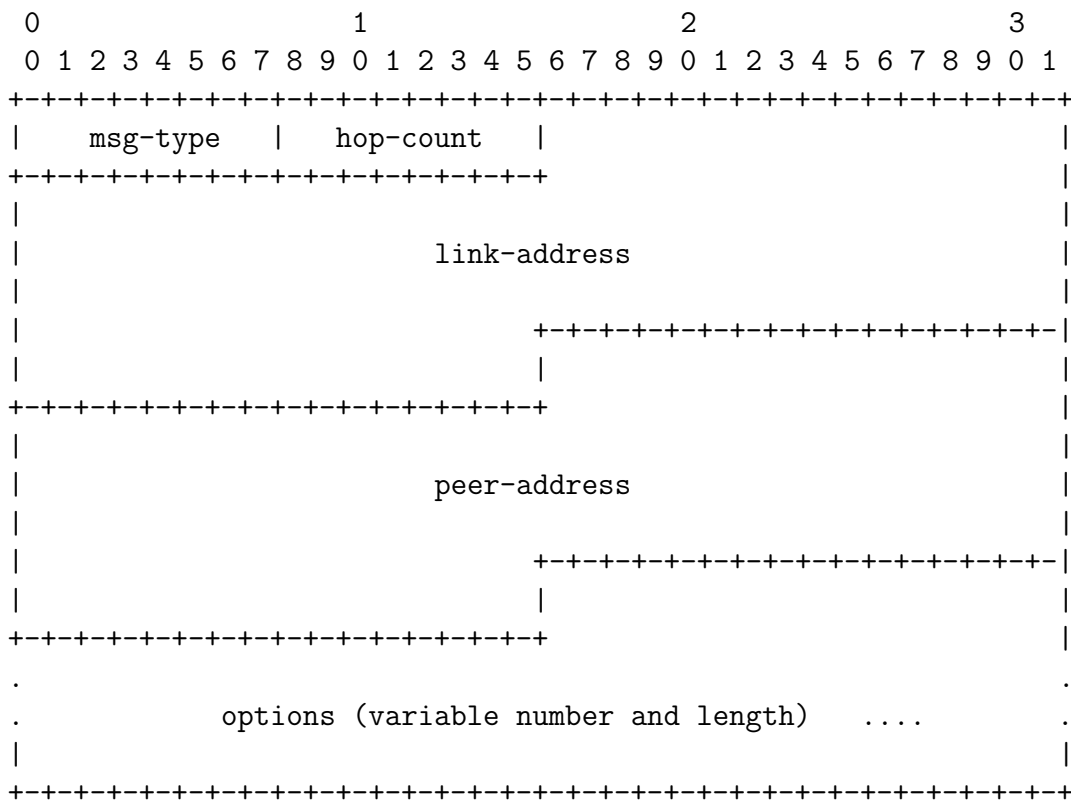
Protokol DHCP umožňuje medzi klientom a serverom zasielanie niekoľkých typov správ. Základné položky správy majú jasne definovaný formát. Toto však neplatí pre položku "options", ktorá má variabilnú dĺžku a počet volieb.

Formát DHCPv6 správ medzi klientom a serverom je nasledujúci:



- transaction-id – identifikácia izolovanej DHCPv6 komunikácie (výmeny správ)

Formát DHCPv6 správ, ktoré odosiela a prijíma relay je nasledujúci:



- hop-count – počet relay agentov, ktorí už preposlali správu

3 Návrh a implementácia programu

Návrh programu `d6r` je realizovaný objektovo orientovane, pričom definícia a implementácia jednotlivých objektov je štrukturovaná do rôznych súborov. Každému objektu je pridelená istá zodpovednosť, konkrétne objekt `ArgParser` realizuje kontrolu vstupných argumentov, `Logger` vypisuje potrebné informácie na štandardný výstup a do systémového logu, `PacketHandler` implementuje prijímanie, spracovávanie a odosielanie jednotlivých paketov. Hlavný program má na starosti volanie metód nad jednotlivými objektami a paralelizáciu. Implementácia je vykonaná v jazyku C++ s využitím štandardných knižníc, sieťových knižníc a knižnice BOOST.

3.1 Hlavný program

Hlavný program je uložený v súbore s názvom `main.c`. Jeho úlohou je inicilizácia jednotlivých objektov, volanie ich metód, vetvenie programu a paralelizácia programu.

Ako prvé je realizované parsovanie argumentov. Následne prebehne kontrola zadaného identifikátora serveru.

Ak používateľ nezadal konkrétne rozhranie, tak relay začne odchytať na všetkých rozhraniach. Pre každé rozhranie je vytvorený osobitný proces pomocou funkcie `fork`. Nasleduje samotné odchytyvanie metódou `receive_client_packet()` z objektu `PacketHandler`. Program vytvára ďalší proces v momente, keď program odchyť DHCPv6 paket od klienta. Synovský proces pokračuje ďalej v programe, pričom vykoná zvyšnú komunikáciu medzi klientom a serverom. Rodičovský opäť začne odchytať na danom rozhraní.

Poslednou úlohou hlavného programu je zavolanie metód pre výpis informácií o pridenej IP adrese/prefixe. To kam budú informácie zapísané špecifikuje používateľ pomocou vstupných prepínačov, pričom sa môže jednať o štandardný výstup alebo systémový log.

3.2 ArgParser

Definícia objektu `ArgParser` sa nachádza v súbore `headers/arg_parser.h`, implementácia jednotlivých metód v súbore `implementation/arg_parser.cpp`. Inicializácia a volanie jednotlivých metód v súbore `main.c`.

Objekt ponúka implementáciu 2 metód, `parser_args()` a `parse_server_input()`. Metóda `parser_args()` s využitím knižnice BOOST spracúva vstupné argumenty relayu, pričom informácie o zadaných prepínačoch ukladá do štruktúry `ParseStruc` definovanej v súbore `headers/parse_struct.h`. Metóda `parse_server_input()` slúži pre spracovanie zadaného serveru. Prijímanie možnosti sú vopred definované doménové meno a IPv6 adresa.

3.3 Logger

Definícia objektu `Logger` sa nachádza v súbore `headers/logger.h`, implementácia jednotlivých metód v súbore `implementation/logger.cpp`. Inicializácia a volanie jednotlivých metód v súbore `main.c`.

V objekte `Logger` sú implementované metódy pre výpis informácií o pridenej adrese/prefixy od DHCPv6 serveru. Metóda `print_debug()` smeruje výpis na štandardný výstup a metóda `print_syslog()` do systémového logu. Pred výpisom do systémového logu je potrebné volať metódu `open_syslog()` a následne po výpise `close_syslog()`. Formát ukážkového výstupu je popísaný v kapitole 5.4.

3.4 PacketHandler

Definícia objektu `PacketHandler` sa nachádza v súbore `headers/packet_handler.h`, implementácia jednotlivých metód v súbore `implementation/packet_handler.cpp`. Inicializácia a volanie jednotlivých metód v súbore `main.c`.

Objekt implementuje jadro programu a to metódy pre prijímanie, spracovávanie a odosielanie paketov medzi klientom a serverom.

Metóda `receive_client_packet()` slúži pre odchytyvanie paketov od klienta. Ako filter sú použité porty 546 a 547. Prostriedy pre odchytyvanie paketu, nastavenie filtru, atď. poskytuje knižnica `pcap`.

Ďalšie metódy slúžia pre prijatie alebo preposlanie špecifického typu správy, tak ako definuje ich názov. Prijímanie a preposielanie paketov je realizované formou socketov. Kopírovanie vybraných partií správ do iných je realizované funkciou `memcpy()`.

Okrem týchto metód objekt navyše obsahuje interné metódy `get_ipv6_addr()` pre získanie požadovaného typu IPv6 adresy zo zadaného rozhrania a `get_mac_addr()` pre získanie MAC adresy zo zadaného rozhrania.

3.5 Ostatné súbory

- `Makefile` - Súbor pre realizáciu prekladu programu.
- `d6r.1` - Manuálový súbor k programu
- `headers/parse_struct.h` - Súbor obsahuje štruktúru, do ktorej sa ukladajú informácie o vstupných argumentoch vložených od používateľa.
- `headers/relay_struct.h` - Súbor obsahuje štruktúru, do ktorej sa ukladajú informácie potrebné pri činnosti relayu.
- `headers/colors.h` - Súbor definuje konštanty pre kolorovaný výstup/chybový výstup.

4 Testovanie programu

Program bol testovaný na debian distribúcií linuxu, konkrétne stabilnej verzii Ubuntu 18.04. Pre samotné realizovanie testov DHCPv6 relayu bolo potrebné zabezpečiť ďalšie dva počítače, jeden pre klienta a druhý pre server. V mojom prípade som túto skutočnosť nasimuloval vytvorením separátnych menných priestorov pomocou utility `ip netns`. Podrobnejší popis prípravy prostredia a realizácie samotných testov je v nasledujúcich odstavcoch.

4.1 Simulácia servera

V procese testovania relayu bolo potrebné simulovať DHCPv6 server. Prvým spôsobom, ktorý som využil bolo zasielanie DHCPv6 správ na vzdialený server vytvorený zadávateľom projektu Ing. Matějom Grégrom, Ph.D. Druhým spôsobom bolo vytvorenie separátneho menného priestoru a využitím programu `dhcpcd` od korporácie Internet System Consortium.

4.2 Simulácia klienta

Simuláciu klienta som realizoval vytovrením separátneho menného priestoru a využitím programu `dhclient` od korporácie Internet System Consortium. DHCPv6 komunikáciu som inicioval pomocou príkazov:

```
$ dhclient -v -6 <rozhranie>
```

pre vyvolanie žiadosti o IPv6 adresu,

```
$ dhclient -v -6 -P <rozhranie>
```

pre vyvolanie žiadosti o prefix pre IPv6 adresu.

4.3 Testovanie relayu

Testovanie programu `d6r` bolo realizované jednak na natívnom IPv6 pripojení, ale aj tunelovanom IPv6 pripojení cez fakultnú VPN. Ku kontrole správnosti práce s obsahom paketov som využíval program Wireshark, kde som ako “capture filter” použil porty 546 a 547. K finálnemu overeniu správnosti odosielania správ som navyše použil program `dhcrelay` opäť od korporácie Internet System Consrtium.

5 Použitie programu

Zdrojový kód je rozdelený do niekoľkých súborov, a tak je potrebné ho prekladať pomocou súboru `Makefile`. V tomto odstavci je krok po kroku popísane vytvorenie a použitie programu.

5.1 Preklad

Po rozbalení projektu do používateľom vybraného priečinka je potrebné vytvoriť spustiteľný súbor `d6r`. To je možné vykonať príkazom `make`, ktorý zdrojové súbory preloží. Pri zavolaní príkazu `make clean` sa spustiteľný súbor vymaže.

5.2 Spustenie

Vytvorený spustiteľný súbor `d6r` sa musí spúšťať v režime root pre správnu prácu s paketmi a knižnicou `libpcap`. Spustenie programu sa realizuje nasledujúcim príkazom:

```
$ sudo ./d6r [-l] [-d] [-i <rozhranie>] -s <server>
```

kde argumenty v hranatých zátvorkách sú voliteľné a všetky ostatné povinné.

5.3 Možnosti spustenia

<code>-s <server></code>	DHCPv6 server, na ktorý je poslaný upravený DHCPv6 paket.
<code>-i <rozhranie></code>	Rozhranie, na ktorom relay naslúcha. Všetky sieťové rozhrania, ak parameter nie je definovaný.
<code>-l</code>	Zapnutie logovania pomocou syslog správ.
<code>-d</code>	Zapnutie debug výpisu na štandardný výstup.

5.4 Ukážka spustenia programu a následného výstupu

Ukážkové použitie programu:

```
$ d6r -s 2001:db8::1111 -d  
2001:db8:a::/56,aa:bb:cc:dd:ee:ff  
2001:db8:b::1,aa:bb:cc:dd:ff:ff
```

6 Záver

Cieľom projektu bolo overiť a rozšíriť nadobudnuté znalosti v predmete Sieťové aplikácie a správa sietí (ISA). Týmto projektom som si tiež prakticky osvojil praktiky protokolu DHCP a prehĺbil znalosti programovania sieťových aplikácií.

Literatúra

- [1] G. Halwasia and S. Bhandari. Client Link-Layer Address Option in DHCPv6. RFC 6939, RFC Editor, May 2013.
- [2] P. Matoušek. *Sít'ové aplikace a jejich architektura*. VUTIUM, 2014.
- [3] T. Mrugalski and M. Siodelski. Dynamic Host Configuration Protocol for IPv6 (DHCPv6). RFC 8416, RFC Editor, November 2018.