

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ



Dokumentace k projektu do předmětu ISA
DHCPv6 relay

18. listopadu 2019

Martin Buchta

Obsah

1	Úvod	3
2	Podstatné pojmy	3
2.1	DHCP server	3
2.2	DHCP relay	3
3	Návrh programu	4
3.1	Paralelismus	4
3.2	Poslouchání zpráv a reakce na ně	4
3.2.1	Zprávy od klienta	4
3.2.2	Zprávy od serveru	5
3.3	Debug výpis	5
3.4	Použité technologie	6
4	Použití programu	6
5	Testování	6
6	Závěr	6
7	Poděkování	6

1 Úvod

Tento dokument ve zkratce popisuje chování DHCPv6 relay, který vznikl jako vypracování školního projektu do předmětu ISA (Síťové aplikace a správa sítí) na Vysokém učení technickém v Brně, Fakultě informačních technologií.

Popisují se zde základní principy chování DHCPv6 relay podle standardu RFC 8415 [1], rozšíření relay podle standardu RFC 6939 [2]. Dále se zde popisuje konkrétní řešení tohoto projektu.

Veškerý níže uvedený text (není-li uvedeno jinak) pod pojmem „IP“ myslí „IPv6“ dle standardu RFC2460 [3] a pod pojmem „DHCP“ myslí „DHCPv6“ podle standardu RFC 8415 [1].

Autorem tohoto dokumentu i samotného řešení popisovaném v tomto dokumentu je Martin Buchta.

2 Podstatné pojmy

2.1 DHCP server

Po připojení zařízení do sítě klient potřebuje znát svou IP adresu, nastavení DNS služeb a další konfiguraci, kterou ale nemusí mít staticky nastavou administrátorem, ale potřebuje si ji získat automaticky od nějaké autority (dostupné v této síti), která mu platnou konfiguraci přidělí. Přesně takovou autoritou bývá DHCP server.

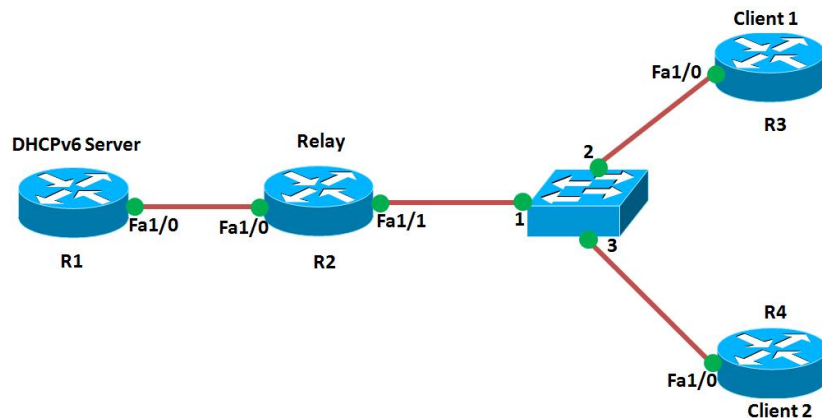
DHCP server obvykle zpracovává požadavky od klienta, které přijdou na port 547 přes UDP [4]. Požadavky mohou chodit buď přímo DHCP serveru, nebo přes multicastovou komunikaci na adrese známou jako *All_DHCP_Servers*, tj. FF05::1:3.

Klient a server si ve 4 zprávách (ve speciálních případech i ve 2 zprávách) vymění požadované informace a klient si podle těchto informací nastaví svou síťovou kartu. Komunikaci začíná klient, který pošle Solicit zprávu na multicastovou adresu *All_DHCP_Servers*, aby našel všechny dostupné DHCP servery. Servery, které mají zájem, klientovi odpoví zprávou typu Advertise, ve které posílají informaci o dostupné IP adrese, případně o celém rozsahu IP adres a délce prefixu (masky). Klient si poté vybere jeden server a jemu pošle zprávu typu Request. Server odpoví zprávou Reply, ve které klientovi potvrzuje danou adresu (nebo prefix).

2.2 DHCP relay

V případě, že se v lokální síti nenachází DHCP server, který by na klientovy požadavky odpovídal, je třeba, aby byl v lokální síti DHCP relay agent (dokáže zachytit požadavky klienta na multicastové adrese), který sám má přístup i mimo vnitřní síť. (Má nastavenou IP adresu, může komunikovat s ostatními DHCP servery přes internet.) Viz Obrázek 1.

DHCP relay pak funguje jako jakási spojka mezi klientem a DHCP serverem. Všechny zprávy od klienta, které zachytí, zabalí do zprávy RELAY-FORW a přes unicast pošle DHCP serveru. DHCP server mu odpoví ve zprávě RELAY-REPL, která obsahuje odpověď pro klienta. DHCP relay tuto zprávu rozbalí a



Obrázek 1: Schéma zapojení DHCP serveru, relay a dvou klientů.

Zdroj: <https://community.cisco.com/t5/networking-documents/stateful-dhcpv6-relay-configuration-example/ta-p/3149338>

obsah přepošle klientovi v rámci vnitřní sítě. Diagram této komunikace ukazuje Obrázek 2.

3 Návrh programu

3.1 Paralelismus

Program musí být schopný odchyťávat zprávy od klientů i od serveru. Uživatel si může zvolit interface, na kterém bude program čekat na zprávy od klienta. Pokud si uživatel žádný interface nezvolí, program má poslouchat na všech dostupných interfacech. Jelikož čekání na packet (přes knihovnu *libpcap* i přes *recvfrom*) je implementováno pomocí aktivního čekání, je třeba program udělat pomocí více procesů nebo vláken.

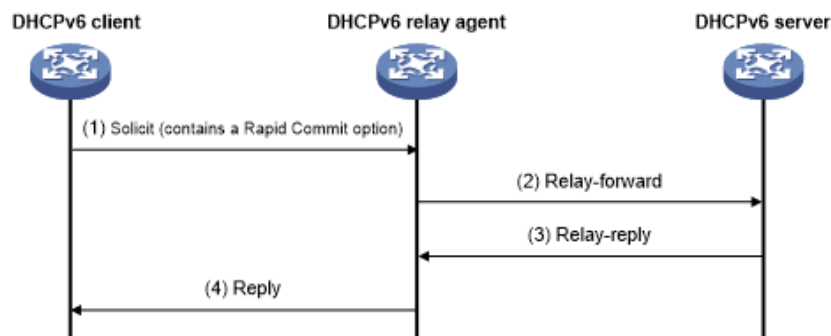
Program je navržen tak, že hlavní proces spustí jedno další vlákno pro poslouchání zpráv ze serveru. Pro každý interface, na kterém má program poslouchat zprávy od klienta, se poté vytvoří vlastní vlákno. Hlavní proces pak jde do nekonečné smyčky, ze které nikdy sám neodejde.

3.2 Poslouchání zpráv a reakce na ně

3.2.1 Zprávy od klienta

Program poslouchá zprávy od klienta pomocí funkce *pcap_loop*. Ta se spouští v proceduře *sniffInterface*. Filtrování zpráv je nastaveno na port 547, na který klienti posílají požadavky podle standardu RFC 8415 [1].

Po zachycení zprávy se volá procedura *callback*, která obdrží mimo jiné zachycený packet. V packetu je ethernetová hlavička a IP hlavička, které je obě nutno přeskočit. Za pevnou částí IP hlavičky mohou být další *options*, které se taky přeskakují. Po přeskočení následující UDP hlavičky následuje DHCP



Obrázek 2: Proces výměny zpráv mezi klientem, DHCP relay a DHCP serverem.
Zdroj: <https://bit.ly/2XznW0l>

zpráva. Alokuje se místo pro novou RELAY FORWARD zprávu, která se naplní následujícími informacemi:

msg-type	12
hop-count	0
link-address	Ip adresa interface, na kterém byla zpráva zachycena
peer-address	Ip adresa klienta (lokální)

Za tyto informace se dá `OPTION_RELAY_MSG` (9), která obsahuje původní zprávu. Dále následuje `INTERFACE_ID_OPTION` (18), která podle standardu RFC 6939 [2] poskytne DHCP serveru MAC adresu klienta.

Nakonec se otevře nový socket typu `SOCK_DGRAM` s DHCP serverem na portu 547 a pomocí funkce `sendto` se zpráva pošle DHCP serveru.

3.2.2 Zprávy od serveru

Program poslouchá zprávy od serveru ve funkci `sniffServer`, která vytvoří socket typu `SOCK_DGRAM` s DHCP serverem. Pomocí funkce `recvfrom` čeká na přijatá data. Po přijetí socketu se volá funkce `callbackServer`, která přijme data zprávy typu `RELY_REPLY` (13). Na rozdíl od přijímání dat pomocí knihovny `pcap`, pokud zachytáváme data přes socket typu `SOCK_DGRAM`, tak nemusíme ořezávat data od hlaviček Ethernetu, IP ani UDP.

Program v daném packetu vyhledá option typu `OPTION_RELAY_MSG` a její obsah přepošle klientovi přes nově vytvořený socket typu `SOCK_DGRAM`. IP adresu klienta vezme ze pole `peer-address`, kterou DHCP server odešle zpátky stejnou, jako ji přijal v `RELAY-FORW` zprávě. Interface, na kterém je cílený klient dostupný, najde v option typu `OPTION_INTERFACE_ID` (18), kterou DHCP server taky nemění a pouze ji přepošle v odpovědi.

3.3 Debug výpis

Program umí vypisovat na standardní výstup a logovat pomocí knihovny `syslog` informace o tom, kterému klientovi (identifikovaný MAC adresou) byla přiřazena která IP adresa. Program tedy vypíše dvojice `[ip adresa, mac adresa]`.

Přiřazená IP adresa je získána ze zprávy Advertise, kterou posílá server. V ní ale není obsažena MAC adresa klienta, ale pouze *DHCP Unique Identifier (DUID)*, která může, ale nemusí obsahovat MAC adresu klienta, viz RFC 8415 [1]. Proto je nutné MAC adresy klienta někde uchovávat. Program ukládá dvojice *[peer address, mac address]* v globální mapě při Solicit požadavku. Při zprávě Advertise si v mapě najde MAC adresu klienta na základě jeho peer adresy.

3.4 Použité technologie

Program je implementován v jazyce C++. Překlad probíhá pomocí G++ překladače a probíhá po zavolání *make*. Po překladu se vytvoří spustitelný soubor *dör*, který se spouští s parametry popsané v kapitole 4.

Program je napsán pro Linuxové systémy, na kterých se nachází knihovna *libpcap* a umí pracovat s *BSD sockets*.

4 Použití programu

Program se musí spustit s root právy.

- -s adresa_serveru (Povinný parametr)
- -l (Zapne logování pomocí Syslogu)
- -d (Zapne výpis informací na standardní výstup)
- -i název_rozhraní (Název rozhraní, na kterém naslouchá. Pokud není uvedeno, poslouchá na všech rozhráních.)

5 Testování

Testování probíhalo přes DHCP server, který byl pro tyto účely přístupný na adrese 2001:67c:1220:80c::93e5:dd2. DHCP relay byl přeložen a spuštěn na počítači s operačním systémem Fedora 29. Na tomto počítači byl spuštěn přes Virtual-Box další počítač s operačním systémem Ubuntu 18, který byl propojený v lokální síti s počítačem běžícím na Fedoře. Tento virtuální stroj měl nastavenou MAC adresu 2e:6d:e4:32:4b:54. Na virtuálním počítači se zapl program dhclient, který spustil proces dotazování se na DHCP server. Funkčnost programu byla potvrzena správným výstupem, sledováním komunikace ve Wiresharku a především přidělením IP adresy virtuálnímu počítači.

6 Závěr

Toto je školní projekt a byl stvořen pouze pro zlepšení vědomostí a zkušeností autora v oblasti síťových aplikací.

7 Poděkování

Děkuji panu Matějovi Grégrovi za zpracování kvalitního zadání a věcnou pomoc studentům na fóru.

Reference

- [1] T. Mrugalski, M. Siodelski, B. Volz, A. Yourtchenko, M. Richardson, S. Jiang, T. Lemon, and T. Winters, “Dynamic host configuration protocol for ipv6 (dhcpv6),” RFC 8415, RFC Editor, November 2018.
- [2] G. Halwasia, S. Bhandari, and W. Dec, “Client link-layer address option in dhcpv6,” RFC 6939, RFC Editor, May 2013.
- [3] S. E. Deering and R. M. Hinden, “Internet protocol, version 6 (ipv6) specification,” RFC 2460, RFC Editor, December 1998. <http://www.rfc-editor.org/rfc/rfc2460.txt>.
- [4] J. Postel, “User datagram protocol,” STD 6, RFC Editor, August 1980. <http://www.rfc-editor.org/rfc/rfc768.txt>.