VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

DHCPV6 RELAY S PODPOROU VLOŽENÍ MAC ADRESY
SÍŤOVÉ APLIKACE A SPRÁVA SÍTÍ

AUTOR **SAMUEL SLÁVKA, xslavk02**

BRNO 2019

Obsah

Introduction

Task was to implement DHCPV6 relay agent with support of inserting MAC address. A relay agent passes messages between client, server and occasionally other agents when client and server. This is useful when server and clients are on different subnets and server has no way to find out on which subnet it should assign IP address. [2]

Relay agent converts clients broadcast message to unicast and communicates with server and client preferably by unicast. Agent should have list of unicast server addresses and use them to communicate with server, only if that is not possible agent uses multicast communication with servers. However, in this project agent gets server address from input and multicast functionality is not needed.

Relay agent includes clients IP address in link-address field, which the server uses to identify the link on which the client is located. Client must includes DUID(DHCP Unique Identifier) which server uses to identify clients.

Due to DUID having multiple types, variable lengths and not being optimal for some uses[1], relay agent may include Client Link-Layer Address Option which allows server to identify the client by his mac address.

Dhcp relay message contains:

message type - which is used to identify purpose and source.

hop count - identifying number of agents who processed this message

link address - globally scoped unicast address used by server to identify where to assign IP address. Agent places its own addresses prefix.

peer address - address relay agent to communicate with client agent places src address from incoming packet.

options - depending on type of message can be different

Implementation

Packet sniffing

I used library libpcap for sniffing for packets. Which caused ICMP port not reachable messages when server sent packets. To solve that I created fork process which just receives packets from server and throws them out. However, this created process which stay after agent closes. So I catch termination attempt ctrl+c and before stop agent first kill process free allocated memory and close sockets.

Packet is casted to structures of headers and later to structures of options which allow easy access to its contents.
After receiving a message, switch statement decides if it came from server or client and accordingly either creates TYPE_RELAY_FORW message or extracts dhcp message from relay reply. If agent gets advertise message from server and was run with parameter -l, the agent logs to syslog IP address and mac address and if there is parameter -d outputs the information to stdout.

Existing problems

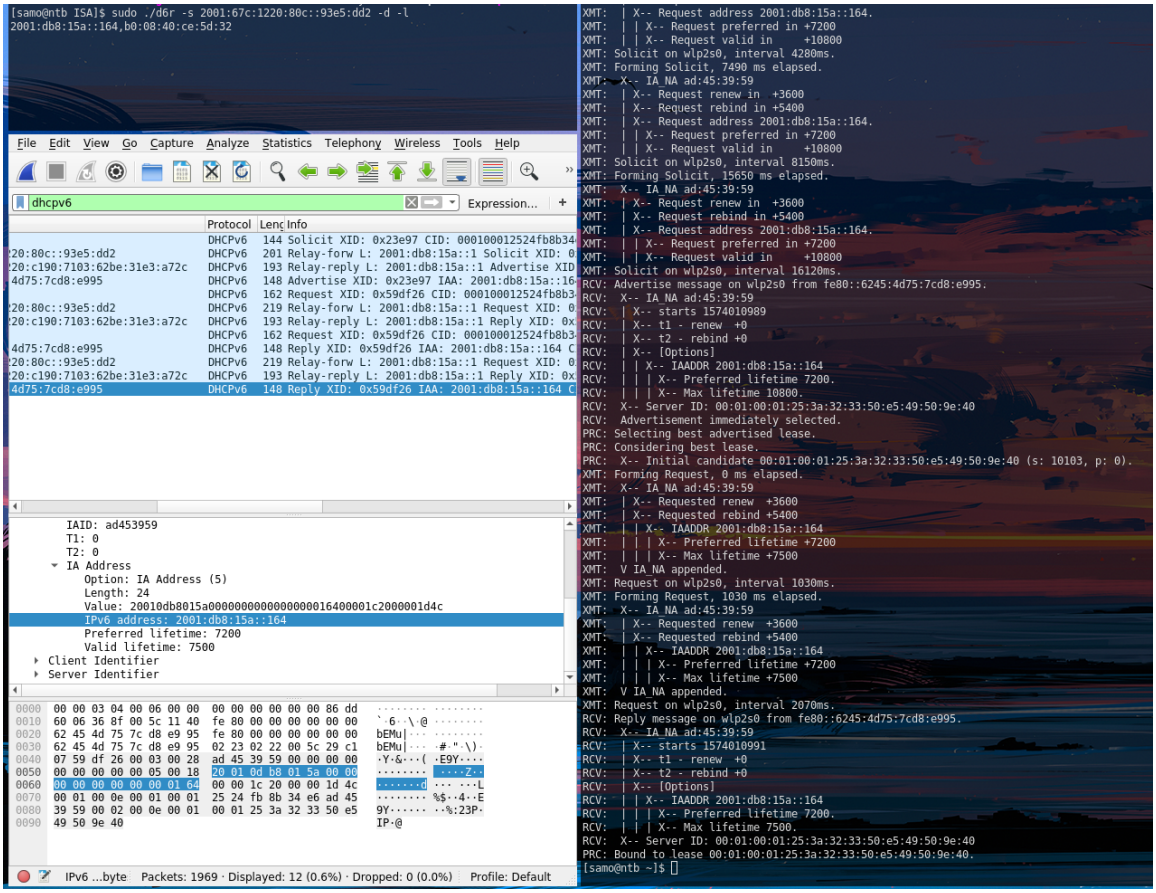
When agent is run with wrong interface name agent continues on interface "any".

I decided that for getting mac address to parse clients link-local address. Which works only if clients link-local address was generated from mac address. If it wasn't the agent will take mac address from Ethernet header which might be address of device connected to client.

For link address I chose my own link-local address, because I didn't find other way to get it.

When agent does not finish communication with sever before another solicit message comes. Entire communication completes two times.

Testing



Obrázek 1: Agent sending data to deterministic server.

Manual

Usage: d6r -s server [-l] [-d] [-i interface]

-s: DHCPv6 server IPv6 address.

-l: Turn on logging with syslog.

-i: Interface on which relay listens if not defined listens on any.

-d: Turn on debug output to stdout.

Literatura

[1] Gaurav Halwasia, W. D. *Client Link-Layer Address Option in DHCPv6* [Internet Requests for Comments]. RFC 6939. Internet Engineering Task Force , May 2013. 1-7 s. Dostupné z: https://tools.ietf.org/html/rfc6939.

[2] Tomek Mrugalski, B. V. A. Y. M. C. R. *Dynamic Host Configuration Protocol for IPv6 (DHCPv6)* [Internet Requests for Comments]. RFC 8415. Internet Engineering Task Force , November 2018. 1-138 s. Dostupné z: https://tools.ietf.org/html/rfc8415.