BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

ISA – project
Monitoring of DHCP communication

November 19, 2023

Denys Petrovskyi
xpetro27

Contents

Problematics

Dynamic host configuration protocol (DHCP) is a network protocol, main purpose of which is to dynamicly assign an IP address to a host. It operates based on client/server model, where the server uses *UDP port 67* and client uses *UDP port 68*. DHCP communicates in 4 stages:[3]

- Discover
- Offer
- Request
- Acknowlege

Usually utilization of an IP prefix can be received using data from DHCP server. Main purpuse of the program is to substitute this function of DHCP server.

Application design

The task can be roughly divided into following steps.

Intercepting DHCP packet

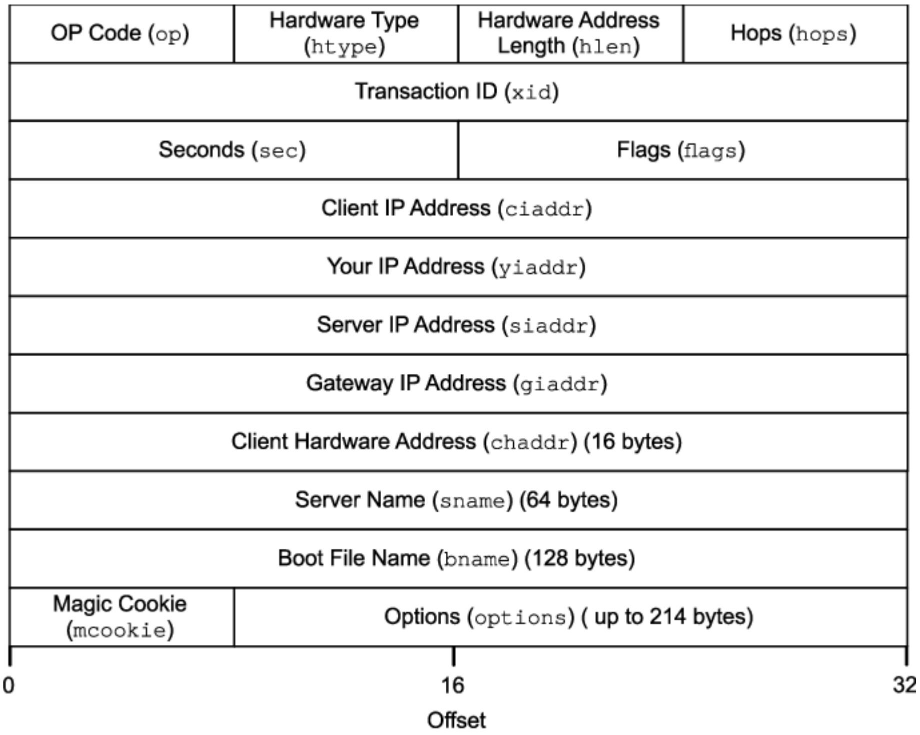
This part is essential for functioning of all program. Its purpuse is to catch DHCP packets from the general flow on a specified interface or pcap file.
As was mentioned earlier, it monitors *udp port 67 and 68*. This part runs in continuous loop and can be interrupted with `Ctrl + C`. After receiveng a packet, it is being passed to further part.

Processing packet

When packet is being intercepted, it is being processed in this part. To correctly handle packet two things have to be considered.

Getting source IP

To analyze utilization of an IP prefix, extracting source IP from the packet is essential. To do that it is important to understand DHCP packet structure. In this application source IP is being extracted as 4 bytes starting from ciaddr (Client IP Address). Ciaddr is used to indicate clients current IP address.


OP Code (op)
Hardware Type (htype)
Hardware Address Length (hlen)
Hops (hops)
Transaction ID (xid)
Seconds (sec)
Flags (flags)
Client IP Address (ciaddr)
Your IP Address (yiaddr)
Server IP Address (siaddr)
Gateway IP Address (giaddr)
Client Hardware Address (chaddr) (16 bytes)
Server Name (sname) (64 bytes)
Boot File Name (bname) (128 bytes)
Magic Cookie (mcookie)
Options (options) (up to 214 bytes)
0
16
32
Offset

DHCP packet format

Processing source IP

In order to get current utilization of an IP prefix, it has to be splitted in two parts: *CIDR* and its IP part itself. *CIDR* is being used to calculate the maximum amount of hosts that this IP prefix can hold. The next step is to understand wheather source IP belong to the prefix or not. To find this out using source IP the network IP is being received. If this network ip is the same as the IP part of prefix, IP belongs to the prefix.

Implementation

This application was fully implemented in language C.

Receiving DHCP packets

In order to receive DHCP packets and their data, header `<pcap.h>` is used. Implementation of obtaining packets slightly differs depending on method with which DHCP packets are being provided.

Receiving from an network interface

To successfully receive data from an interface, firstly, pcap handler needs to be opened on that network interface with `pcap_open_live()`. Then application will continuously catch dhcp packets using function `pcap_loop()` and pass them further, this process will be interrupted with *CTRL + C* combination or if utilization of a prefix will reach 100%.

Receiving from pcap file

Same as working with interface, to obtain data from a file, it is being opened using function `pcap_open_offline()`. After that data will be processed until file ends or program will be interrupted.

Processing packet data

To extract source IP from the packet and to process it further, is used `struct in_addr`, from the header `<arpa/inet.h>`, and its functions. With help of several functions from `<string.h>` IP prefix is being splitted into two parts and then using bitwise operations is being checked if IP coresponds to a prefix.[2]

Output

Every time new packet was processed, updated information is printed to the output. If application runs on a specidied interface, it can be considered as a console application. In that case output is executed using header `<ncurses.h>`[4]. If utilization of a prefix exceeds 50%, information about this is printed to the log using `<syslog.h>`.[1]

Instructions for use

After unzipping tar archive with the project, compile code with command `make`. Executable file dhcp-stats will be created. It can be runned using the following command:

```
./dhcp-stats [-r <filename>] [-i <interface-name>] <ip-prefix> [ <ip-prefix> [ ... ] ]
```

- **[-r <filename>]**: path to a pcap file.
- **[-i <interface-name>]**: network interface.
- **<ip-prefix> [<ip-prefix> [...]]**: list of IP prefixes separated by a space.

Interface or pcap file must be provided for the program to work but not both simultaneously. At least one IP prefix has to be given also.

References

[1] *18.2.5 Syslog Example.*

[2] GeeksForGeeks. C library - <string.h>. [online].

[3] Microsoft. Dynamic host configuration protocol (dhcp). [online], 2021.

[4] Opensource.com. Position text on your screen in linux with ncurses. [online], 2021.