BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF INFORMATION TECHNOLOGY
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

ISA - MONITORING DHCP COMMUNICATION

ISA - MONITOROVÁNÍ DHCP KOMUNIKACE

AUTHOR **ANDREJ SMATANA**
AUTOR PRÁCE

BRNO 2023

SMATANA, Andrej. *ISA - Monitoring DHCP communication.* Brno, 2023. . Brno University of Technology, Faculty of Information Technology.

Contents

1. The issue

Dynamic Host Configuration Protocol (DHCP)[4] is a network protocol used to automate the process of configuring devices on IP networks by their distribution, thus allowing them to use other network services such as DNS, NTP, and any communication protocol based on UDP or TCP. DHCP is key to configuring subnet masks, default gateways, and DNS server information.

As those networks grow, it is essential to plan your capacity requirements, ensuring the constant availability of your network. So, tracking the allocation of IP addresses for specific subnets could help you identify the utilization pattern over some time[1].

2. Application Design

The DHCP-Stats application is developed in C++. The application is divided into several components, each responsible for a specific task:

2.1 Packet Capture

This component uses the pcap library[1] to capture packets from the network. It filters the traffic only to include DHCP packets, which are then passed to the next component for analysis.

2.2 Packet Analysis

This component is responsible for analyzing the captured packets. It extracts relevant information from the packets, such as the DHCP message type and the client's IP address. This information is then used to update the application's statistics.

[1] https://www.dnsstuff.com/dhcp-configuration

[1] https://www.tcpdump.org/pcap.html

2.3 Statistics Management

This component maintains the application's statistics. It keeps track of the number of different types of DHCP messages that have been captured, as well as the number of unique clients.

The NCURSES library[2] will be used to show the statistics to the user. It provides a real-time view of the application's statistics similar to the `top`[3] utility.

2.4 Error Handling

The application should be able to deal with incorrectly given arguments in the command line utility.

When monitoring or calculating the statistics of IP prefixes, it prints out the error message to stdout that the prefix has exceeded 50 % of utilization and at syslog, too. Respectively, it logs the message to syslog if it exceeds 80 % and 100 % of utilization.

3. Implementation

3.1 Packet Capture

The filter used for capturing packets is set to `port 67 or port 68`, as according to RFC 2131, the DHCP communication flows through these ports *(from client to DHCP server it is port 67 and in opposite direction, it is port 68 on the client)* [4].

It is used either in offline mode *(reading pcap files[1])* or live mode, which sniffs on the user-given network interface in non-promiscuous mode, meaning it will not read frames intended for other machines or network devices than DHCP server.

3.2 Packet Analysis

DHCP packet is an Ethernet frame containing an IP packet, which contains a UDP datagram, which contains the DHCP data [3] [2] [1].

My implementation first checks whether a frame is long enough to contain the information about DHCP. This is according to the previously mentioned standards `14 + 20 + 8 + 312 = 354` octets, where `14` is the length of the Ethernet frame header, `20` is the length of IP header and `8` is the length for UDP header in octets. The last number, `312` is the minimum size in octets required for DHCP packet to contain the necessary information for analysis.

[2] https://tldp.org/HOWTO/NCURSES-Programming-HOWTO/intro.html

[3] https://man7.org/linux/man-pages/man1/top.1.html

[1] https://www.endace.com/learn/what-is-a-pcap-file

[2] https://avocado89.medium.com/dhcp-packet-analysis-c84827e162f0

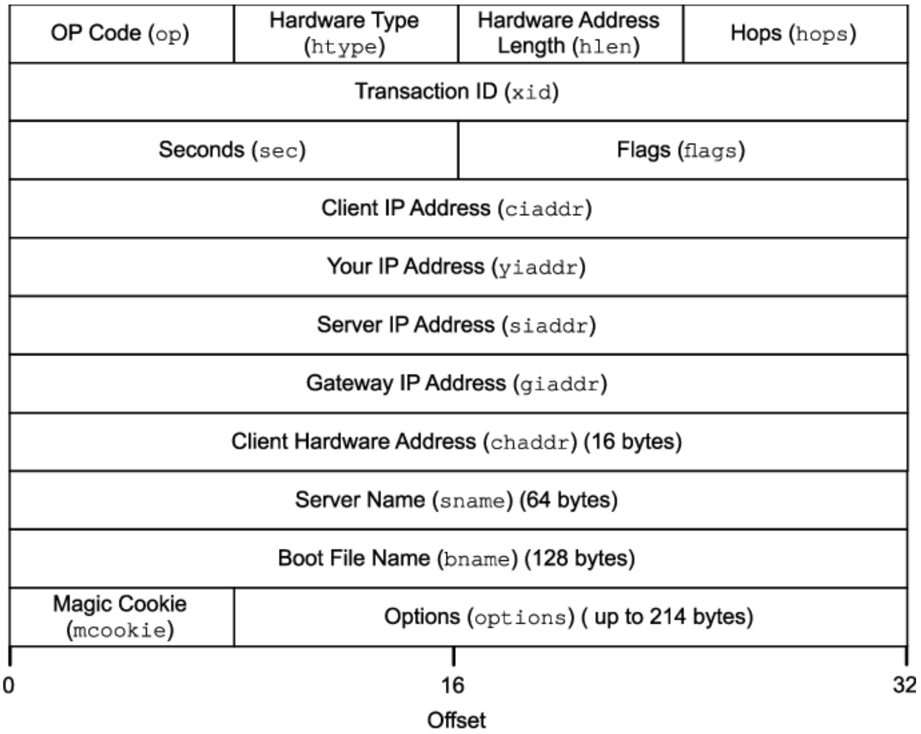

OP Code (op)
Hardware Type (htype)
Hardware Address Length (hlen)
Hops (hops)
Transaction ID (xid)
Seconds (sec)
Flags (flags)
Client IP Address (ciaddr)
Your IP Address (yiaddr)
Server IP Address (siaddr)
Gateway IP Address (giaddr)
Client Hardware Address (chaddr) (16 bytes)
Server Name (sname) (64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

Figure 3.1: DHCP Packet header[2]

Then, the examination of the contents of the frame is done by first checking the OP code `op` as shown in 3.1, whether it contains `2 - DHCPREPLY` code as a reply from DHCP server to client.

Next, the 4-octets magic cookie `mcookie` in before the Options is checked to see whether it equals decimal values 99, 130, 83, and 99 [4]. If so, it is confirmed that it is a DHCP message, not the BOOTP one[3].

As a last thing, it iterates through `options` until the option equals `53 - DHCP Msg type`[4] or `255 - End`. After that, it is then checked for `5 - DHCPACK` type of the DHCP message to confirm that the server sends an ACK message to the client, providing what the IP address of a client will be in `yiaddr` field of DHCP header. Such communication is shown at 3.2.

3.3 Statistics Management

After the IP address is extracted from DHCP header `yiaddr` field (3.1), the address is stored in statistics, from which the ncurses window with stats is updated.

3.4 Logging messages

In the case of 50 % utilization of IP addresses at prefix, the message is printed out to `stdout` and `syslog`.

[3] https://community.cisco.com/t5/switching/why-dhcp-option-has-quot-magic-cookie-quot/td-p/1764244

[4] https://www.iana.org/assignments/bootp-dhcp-parameters/bootp-dhcp-parameters.xhtml

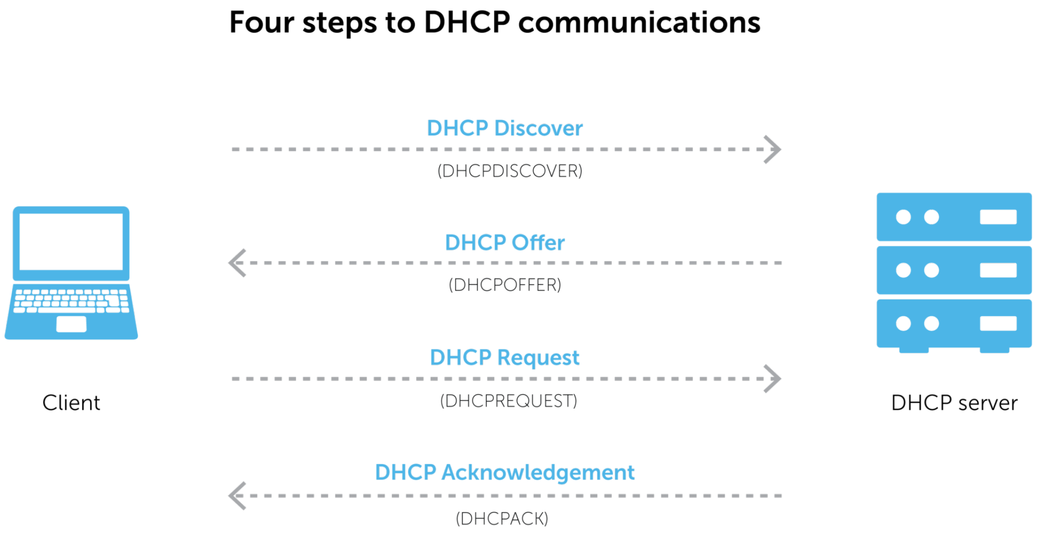


Figure 3.2: DHCP Communication

The remaining two types of messages (80 % and 100 % utilization) are logged only at syslog.

The messages at syslog are started with `dhcp-stats` as the name of the application. Their priority is set to `LOG_NOTICE`.

4. User guide

4.1 Introduction

The `dhcp-stats` command displays statistics for each given IP-PREFIX for a DHCP server. It can be used to monitor the usage of IP addresses in a network and to identify potential issues with the DHCP configuration. The application should be able to end peacefully anytime a SIGINT signal is received.

4.2 Requirements

`dhcp-stats` requires a Linux operating system with libpcap, syslog and ncurses installed. It also requires root privileges to access the network interface.

4.3 Compile the binary

If you want to compile the program, make sure the `g++` is also present at the system, best if in version `g++ (Debian 12.2.0-14) 12.2.0` and also other requirements fro the previous section are fulfilled.

To get the binary `dhcp-stats`, make sure you are in a directory, where files `dhcpmonitor.cpp`, `dhcpmonitor.h` and `Makefile` are present. If so, type in terminal `make` to create a binary file named `dhcp-stats`.

4.4 Options

- `-r, -read=FILENAME`: Read statistics from a pcap file instead of sniffing on an interface. Prints the stats directly to stdout.
- `-i, -interface=INTERFACE-NAME`: Specify the name of the interface to sniff on without using promiscuous mode. Prints the statistics in a ncurses window that is frequently updated.

4.5 Arguments

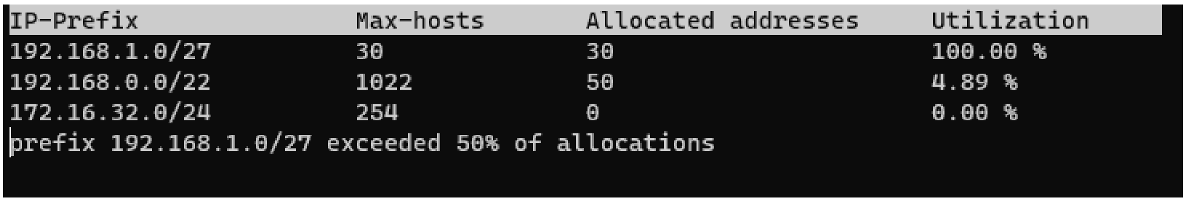
- `IP-PREFIX`: The prefix of the subnet where the statistics will be computed on.

4.6 Return Codes

`dhcp-stats` returns 0 for a successful read of the pcap file or ending program with SIGINT signal, 1 if `dhcp-stats` returns an error.

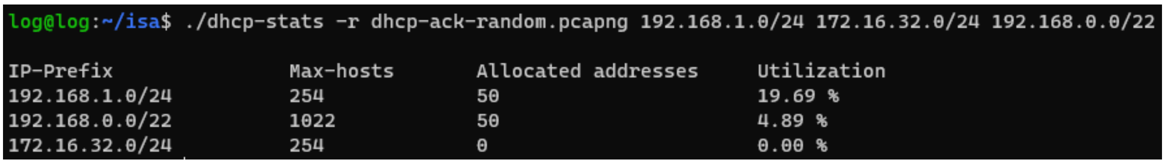
4.7 Examples

- To display statistics for the subnets 192.168.1.0/27, 172.16.32.0/24, and 192.168.0.0/22 on interface lo, run as a root user: `dhcp-stats -i lo 192.168.1.0/27 172.16.32.0/24 192.168.0.0/22` and the output may be as at a figure 4.1
- To display statistics for the subnet 192.168.1.0/24 from a pcap file, run: `dhcp-stats -r stats.pcap 192.168.1.0/24` and the output may look like in figure 4.2.



```
IP-Prefix                Max-hosts        Allocated addresses      Utilization
192.168.1.0/27           30               30                       100.00 %
192.168.0.0/22           1022             50                       4.89 %
172.16.32.0/24           254              0                        0.00 %
prefix 192.168.1.0/27 exceeded 50% of allocations
```

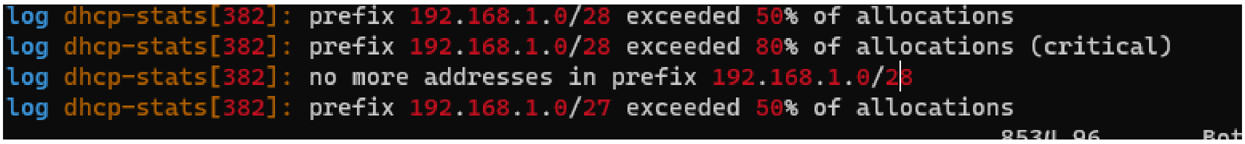
Figure 4.1: DHCP monitoring - example 1



```
log@log:~/isa$ ./dhcp-stats -r dhcp-ack-random.pcapng 192.168.1.0/24 172.16.32.0/24 192.168.0.0/22

IP-Prefix               Max-hosts       Allocated addresses     Utilization
192.168.1.0/24          254             50                      19.69 %
192.168.0.0/22          1022            50                      4.89 %
172.16.32.0/24          254             0                       0.00 %
```

Figure 4.2: DHCP monitoring - example 2



```
log dhcp-stats[382]: prefix 192.168.1.0/28 exceeded 50% of allocations
log dhcp-stats[382]: prefix 192.168.1.0/28 exceeded 80% of allocations (critical)
log dhcp-stats[382]: no more addresses in prefix 192.168.1.0/28
log dhcp-stats[382]: prefix 192.168.1.0/27 exceeded 50% of allocations
```

Figure 4.3: Messages in syslog

Bibliography

[1] *User Datagram Protocol* [RFC 768]. RFC Editor, august 1980. DOI: 10.17487/RFC0768. Available at: https://www.rfc-editor.org/info/rfc768.

[2] *A Standard for the Transmission of IP Datagrams over Ethernet Networks* [RFC 894]. RFC Editor, 1. april 1984. DOI: 10.17487/RFC0894. Available at: https://www.rfc-editor.org/info/rfc894.

[3] IEEE Standard for Ethernet. *IEEE Std 802.3-2022 (Revision of IEEE Std 802.3-2018).* 2022, p. 1–7025. DOI: 10.1109/IEEESTD.2022.9844436.

[4] Droms, R. *Dynamic Host Configuration Protocol* [RFC 2131]. RFC Editor, march 1997. DOI: 10.17487/RFC2131. Available at: https://www.rfc-editor.org/info/rfc2131.