



# DHCP Monitoring

Manual for dhcp-stats.cpp

Tereza Lapčíková, xlapci03

[xlapci03@fit.vutbr.cz](mailto:xlapci03@fit.vutbr.cz)

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	DHCP messages . . . . .	2
1.2	DHCP options . . . . .	2
<b>2</b>	<b>Implementation</b>	<b>2</b>
2.1	User Input Parser . . . . .	2
2.2	Packet Sniffer . . . . .	2
2.3	Packet Handling . . . . .	2
<b>3</b>	<b>Usage</b>	<b>3</b>
<b>4</b>	<b>References</b>	<b>4</b>

# 1 Introduction

Abbreviation DHCP stands for Dynamic Host Configuration Protocol, which is a protocol used on Internet Protocol (IP) networks to automatically assign IP addresses and other network configuration information to devices on network. DHCP communication uses UDP [1].

## 1.1 DHCP messages

The process of leasing an address consists of four phases with the relevant four messages – discover, offer, request and acknowledge. The discover message is broadcasted by a DHCP client to available servers on the network subnet when requesting a possible IP address lease. When a DHCP server receives the discover message, it reserves an IP address for the client and sends an offer message with lease offer. The client can receive multiple offers but accepts only one and send a request message to the relevant server as a response. When the server receives the request message from the client, it response with the acknowledge message, that includes the lease time and other configuration information [1].

## 1.2 DHCP options

DHCP packets contain octet strings of varying length called options. Options offer more detailed configuration parameters [1].

# 2 Implementation

The implementation consists of three main parts - user input parser, packet sniffer and packet handling.

## 2.1 User Input Parser

Input arguments are received from command line in format described in chapter 3. Parsing of input arguments is done by function `parse_args()`. Packets can be received either from an interface or from a pcap file. When both options occur, the first one (interface) is chosen. Next argument, that is expected is IP prefix in following format: `x.x.x.x/(length of subnet mask)`, where x is octet of an IP address in decimal format.

## 2.2 Packet Sniffer

External library `Libpcap` was used in implementation of packet sniffer. Function `pcap_open_live()` was used to open the interface for capturing. Function `pcap_open_offline()` was used for file respectively. Capturing is provided by function `pcap_loop()`.

## 2.3 Packet Handling

Received packet is read layer by layer (IP protocol, UDP) using a callback function `ph_routine()`. As it reaches DHCP layer, by checking the option (53) is analyzed DHCP message type. My implementation takes into consideration both request and acknowledge messages, because deciding

only based on acknowledge would not be precise. For an IP address to be counted as leased, it has to be requested by request message and then confirmed using acknowledge message sent by the server. The requested IP address is stored. When an intercepted acknowledge packet's address corresponds with address that was previously requested, it can be counted as leased. It is decided in which subnet IP address belongs and the statistics are calculated and then displayed. Addresses available for assignment contain both network address and broadcast address.

### 3 Usage

Program is compiled with g++ using Makefile

```
$ make
```

The program can be run using following command:

```
$ ./dhcp-stats [-r <pcap-file>] [-i <interface>] <ip-prefix> [<ip-prefix> [...]]
```

When using the interface source option, the program requires root privileges because of promiscuous mode when receiving packets.

Example of monitoring statistics:

```
$ ./dhcp-stats -r ./tests/DHCP_DORA.pcap 172.16.20.96/28 172.16.20.7/29
```

```
IP-Prefix Max-hosts Allocated-addresses Utilization
172.16.20.96/28 16 1 6.25%
172.16.20.7/29 8 0 0.00%
```

## 4 References

- [1] Ralph Droms. *Dynamic Host Configuration Protocol*. RFC 2131. Mar. 1997. DOI: 10.17487/RFC2131. URL: <https://www.rfc-editor.org/info/rfc2131>