

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

## FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

ISA – Síťové aplikace a správa sítí  
Projekt: Monitorování DHCP komunikace

# Obsah

<b>1</b>	<b>Úvod</b>	<b>2</b>
<b>2</b>	<b>Shrnutí teorie</b>	<b>2</b>
<b>3</b>	<b>Návrh a implementace</b>	<b>2</b>
<b>4</b>	<b>Návod k použití</b>	<b>2</b>
<b>5</b>	<b>Testování</b>	<b>3</b>
5.1	Spuštění programu pomocí $-i$ . . . . .	4
5.2	Spuštění programu pomocí $-r$ . . . . .	5

# 1 Úvod

Cílem projektu je vytvořit program C pro sběr statistik o provozu DHCP. Pokud je prefix zaplněn na více než 50%, program to ohlásí administrátorovi na `stdout` a prostřednictvím logování přes `syslog`.

## 2 Shrnutí teorie

Dynamic Host Configuration Protocol (DHCP) – je síťový protokol používaný k automatickému přidělování IP adres a konfiguračních informací (jako je brána, DNS server apod.) zařízením připojeným do síťového prostředí. DHCP umožňuje snadné a efektivní spravování IP adres v síti tím, že přiřazuje adresy dynamicky, což eliminuje konflikty a umožňuje flexibilní konfiguraci síťových zařízení.

Server naslouchá na portu UDP číslo 67 a klient na portu UDP číslo 68. Operace DHCP se často označují zkratkou DORA pro Discovery, Offer, Request a Acknowledgement.

## 3 Návrh a implementace

Program začíná tím, že při spuštění analyzuje argumenty. Dále je inicializován slovník `stats_map` k ukládání a budoucí aktualizaci statistik prefixů, které byly získány při spuštění programu. Je vytvořen `pcap_handler` s filtrem "`udp and (port 67)`".

Kontroluje se také, zda byl program spuštěn s příznakem '`-r`' nebo '`-i`'. Na základě toho se určí použití `pcap_open_offline()` nebo `pcap_open_live()`. Následuje funkce `pcap_loop()`, která zachytí a zpracuje síťový provoz.

Program kontroluje všechny pakety a volá funkci `packet_handler()`, které spadají pod filtr. Aby bylo jisté, že se jedná o paket DHCPACK[3], kontrolují se tři věci:

- Zda se jedná o `reply-packet` (`opcode == 2`)
- Zda se jedná o DHCP Message Type (Tag 53)
- Jestli hodnota tohoto DHCP Message Type se rovná 5.

Poté se pomocí funkce `is_in_prefixes()` zkontroluje, zda je paket zařazen do daného síťového prefixu. Ještě předtím se však zkontroluje, zda tento paket již nebyl přidán do statistik.

Například adresa typu `yiaddr=192.168.1.1` se tedy započítá pouze jednou, pokud existují 2 takové pakety. K tomuto účelu se používá funkce `is_in_allocated_address()`.

A konečně, v závislosti na tom, jak byl program spuštěn, zda s '`-r`' nebo '`-i`' – určí, zda program bude zobrazovat statistiky v reálném čase, nebo ne. V případě `offline` režimu program projde všechny pakety, shromáždí statistiky a poté je vypíše na `stdout`. Pokud je zatížení síťového prefixu větší než 50%, zapíše se do log file pomocí protokolu `syslog`.

V režimu `online` se program chová jako konzolová aplikace a aktualizuje statistiky v reálném čase. K implementaci jsem použil knihovnu `ncurses`[4].

## 4 Návod k použití

- Rozbalte archiv pomocí příkazu: `tar -xf xassat00.tar`
- Spusťte příkaz `make`
- Příklad spuštění programu: `sudo ./dhcp-stats -i eth0 192.168.50.0/24`

Další informace o používání programu najdete v manuálu[6] s příkazem: `man -l dhcp-stats.1`

## 5 Testování

Pro testování jsem použil jazyk Python s knihovnou Scapy[2], která umožňuje vytvářet pakety, které lze posílat po síti. Pro sbírání dat v reálném čase jsem použil knihovnu libpcap[5].

Takto vypadá můj malý skript, který jsem napsal pro vytvoření DHCP ACK packetů podle dokumentace[1] Scapy. Bohužel skript byl napsán narychlo, takže pokud je potřeba posílat pakety na více prefixů, bude nutné přidat/změnit přímo v kódu skriptu.

```
from scapy.all import *

x = 1 # You can edit while cycle to test for example '< 150'
while x < 10:

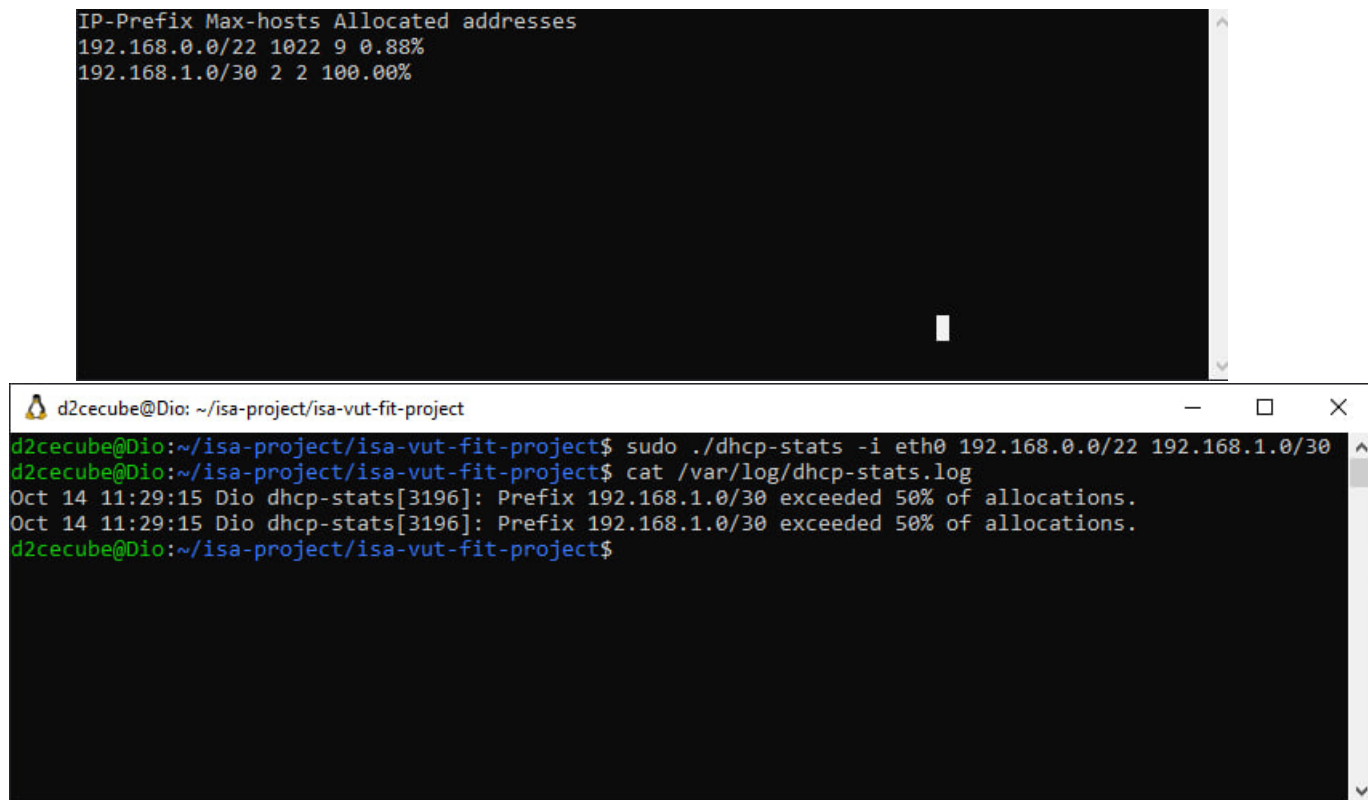
    # Change yiaddr there to specific prefix, that you want to test/get stats
    yiaddr_addr = str(x)

    dhcp_ack_1 = Ether(dst="ff:ff:ff:ff:ff:ff") / \
        IP(src="0.0.0.0", dst="255.255.255.255") / UDP(sport=67, dport=68) / \
        BOOTP(op=2, xid=0x01020304, yiaddr=yiaddr_addr) / \
        DHCP(options=[("message-type", "ack"), \
            ("subnet_mask", "255.255.255.0"), "end"])

    sendp(dhcp_ack_1, iface="eth0")
    x += 1
```

## 5.1 Spuštění programu pomocí `-i`

Pro tento testovací scénář budu vytvářet a odesílat pakety z adresy 192.168.1.1 do 192.168.1.9. A program bude spuštěn takto: `sudo ./dhcp-stats -i eth0 192.168.0.0/22 192.168.1.0/30`



The image shows a terminal window with two parts. The top part is a screenshot of the output of the `dhcp-stats` command, showing a table of IP prefixes, maximum hosts, allocated addresses, and their percentage of allocation. The bottom part is a screenshot of a terminal window showing the command being executed and the output of the `cat /var/log/dhcp-stats.log` command, which shows two log entries indicating that the prefix 192.168.1.0/30 exceeded 50% of allocations.

```
IP-Prefix Max-hosts Allocated addresses
192.168.0.0/22 1022 9 0.88%
192.168.1.0/30 2 2 100.00%
```

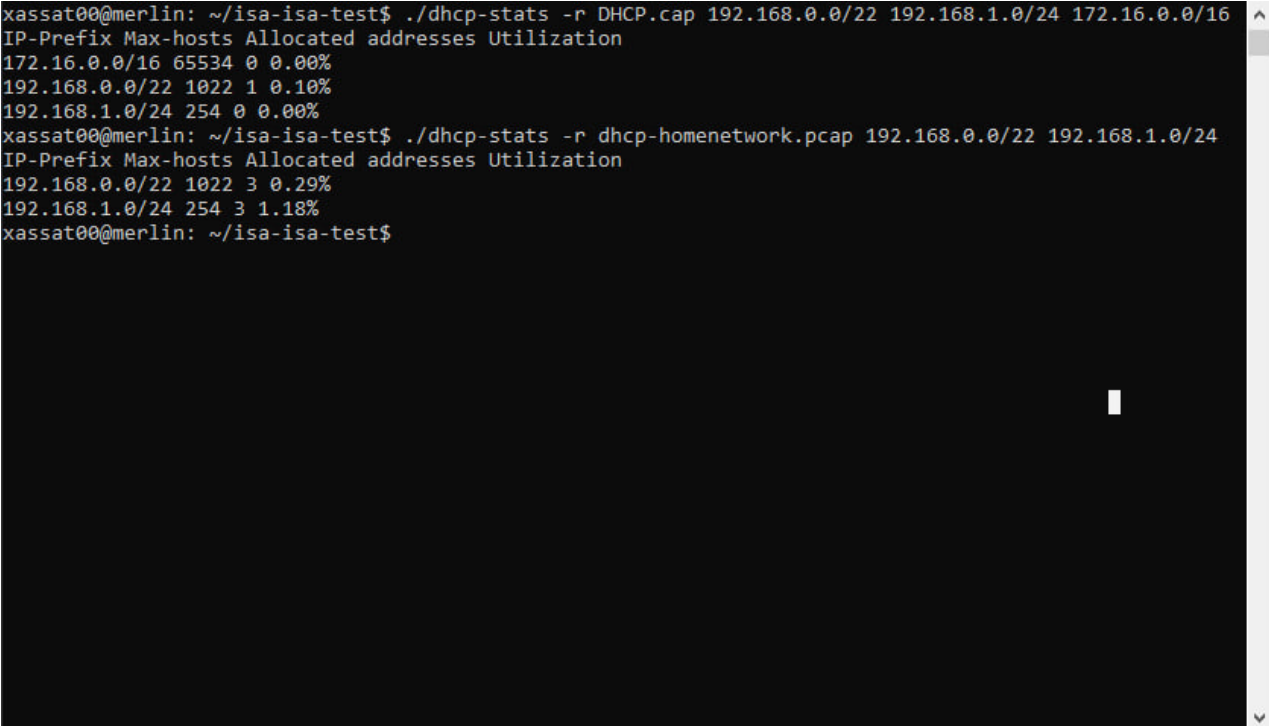
```
d2cecube@Dio: ~/isa-project/isa-vut-fit-project
d2cecube@Dio:~/isa-project/isa-vut-fit-project$ sudo ./dhcp-stats -i eth0 192.168.0.0/22 192.168.1.0/30
d2cecube@Dio:~/isa-project/isa-vut-fit-project$ cat /var/log/dhcp-stats.log
Oct 14 11:29:15 Dio dhcp-stats[3196]: Prefix 192.168.1.0/30 exceeded 50% of allocations.
Oct 14 11:29:15 Dio dhcp-stats[3196]: Prefix 192.168.1.0/30 exceeded 50% of allocations.
d2cecube@Dio:~/isa-project/isa-vut-fit-project$
```

Obrázek 1: Výsledky testu a snímek obrazovky souboru protokolu `-i`

Bohužel je problém v tom, že se do souboru logu zapisuje dvakrát místo jednou. Zatím se mně tento problém nepodařilo vyřešit.

## 5.2 Spuštění programu pomocí `-r`

Pro tento testovací scénář budu číst ze souboru pcap nalezeného na internetu, kde probíhala komunikace DHCP. Protože pro tento příkaz nebudu potřebovat `sudo`, rozhodl jsem se ho otestovat na serveru `merlin.fit.vutbr.cz`.



```
xassat00@merlin: ~/isa-isa-test$ ./dhcp-stats -r DHCP.cap 192.168.0.0/22 192.168.1.0/24 172.16.0.0/16
IP-Prefix Max-hosts Allocated addresses Utilization
172.16.0.0/16 65534 0 0.00%
192.168.0.0/22 1022 1 0.10%
192.168.1.0/24 254 0 0.00%
xassat00@merlin: ~/isa-isa-test$ ./dhcp-stats -r dhcp-homenetwork.pcap 192.168.0.0/22 192.168.1.0/24
IP-Prefix Max-hosts Allocated addresses Utilization
192.168.0.0/22 1022 3 0.29%
192.168.1.0/24 254 3 1.18%
xassat00@merlin: ~/isa-isa-test$
```

Obrázek 2: Výsledky testu a snímek obrazovky souboru protokolu pomocí `-r`

## Bibliography

- [1] Biondi, P.; the Scapy community: "scapy.layers.dhcp". Dostupné z: <https://scapy.readthedocs.io/en/latest/api/scapy.layers.dhcp.html>
- [2] Biondi, P.; the Scapy community: "Welcome to Scapy's documentation!". Dostupné z: <https://scapy.readthedocs.io/en/latest/index.html>
- [3] InternetAssignedNumbersAuthority: "Dynamic Host Configuration Protocol (DHCP) and Bootstrap Protocol (BOOTP) Parameters". Dostupné z: <https://www.iana.org/assignments/bootp-dhcp-parameters/bootp-dhcp-parameters.xhtml>
- [4] Padala, P.: "NCURSES Programming HOWTO". Dostupné z: <https://tldp.org/HOWTO/NCURSES-Programming-HOWTO/>
- [5] Van Jacobson, C. L.; Steven McCanne, a. o. t. L. B. N. L. U. o. C.: "PCAP(3PCAP) MAN PAGE". Dostupné z: <https://www.tcpdump.org/manpages/pcap.3pcap.html>
- [6] Wirzenius, L.: "Writing manual pages". Dostupné z: <https://liw.fi/manpages/>