

## Wyzwanie - Warsztaty 2 - Maciej Dengusiak

### Krok 1 - konfiguracja

Poprzez ustawienia każdej z maszyn w trybie expert w zakładce network konfigurujemy kolejne karty sieciowe połączone z wirtualnymi sieciami opisanymi w wyzwaniu.

### Krok 2 - nazwy interfejsów

Uruchomiłem wszystkie maszyny.

- Uruchom maszyny i nazwij ich interfejsy tak jak na rysunku powyżej. Uzyskaj konfigurację sieciową dla interfejsu `enp-ext` maszyny *Virbian0* poleceniem

```
V0#> dhclient -v enp-ext
```

Nazywamy kolejne interfejsy jak na rysunku komendą `ip link set 'stara_nazwa' name 'nowa_nazwa'`. Dla rysunku z wyzwania komendy będą wyglądały następująco.

#### Virbian0

```
user@virbian:~$ sudo ip link set enp0s3 name enp-ext
user@virbian:~$ sudo ip link set enp0s8 name enp-loc0
user@virbian:~$
```

#### Virbian1

```
user@virbian:~$ sudo ip link set enp0s3 name enp-loc0
user@virbian:~$ sudo ip link set enp0s8 name enp-loc1
user@virbian:~$
```

#### Virbian2

```
user@virbian:~$ sudo ip link set enp0s3 name enp-loc1
user@virbian:~$ S
```

Aby upewnić się, że dobrze nazwałem interfejsy, użyłem komendy `ip link`. Przykład z virbiana 1 na którym widać już nowo ustawione nazwy kart sieciowych.

```

user@virbian:~$ ip link
1: lo: <LOOPBACK,UP,LOWER UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAU
LT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: enp-loc0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode DEFAULT
group default qlen 1000
    link/ether 08:00:27:2a:bc:dd brd ff:ff:ff:ff:ff:ff
    altname enp0s3
3: enp-loc1: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode DEFAULT
group default qlen 1000
    link/ether 08:00:27:34:53:0e brd ff:ff:ff:ff:ff:ff
    altname enp0s8

```

### Konfiguracja sieciowa enp-ext

```

user@virbian:~$ sudo dhclient -v enp-ext
Internet Systems Consortium DHCP Client 4.4.3-P1
Copyright 2004-2022 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Listening on LPF/enp-ext/08:00:27:07:5a:5c
Sending on   LPF/enp-ext/08:00:27:07:5a:5c
Sending on   Socket/fallback
DHCPDISCOVER on enp-ext to 255.255.255.255 port 67 interval 4
DHCPOFFER of 10.0.2.15 from 10.0.2.2
DHCPREQUEST for 10.0.2.15 on enp-ext to 255.255.255.255 port 67
DHCPACK of 10.0.2.15 from 10.0.2.2
bound to 10.0.2.15 -- renewal in 39072 seconds.
user@virbian:~$

```

### Krok 3 - przypisanie IP

Każdą kartę aktywuję poleceniem `ip link set up dev 'nazwa'` a następnie przypisuje adres IP poleceniem `sudo ip addr add 'IP' dev 'nazwa'`.

#### Virbian0

```

user@virbian:~$ sudo ip link set up dev enp-loc0
user@virbian:~$ sudo ip addr add 192.168.0.1/24 dev enp-loc0
user@virbian:~$

```

#### Virbian1

```

user@virbian:~$ sudo ip link set up dev enp-loc0
user@virbian:~$ sudo ip link set up dev enp-loc1
user@virbian:~$ sudo ip addr add 192.168.0.2/24 dev enp-loc0
user@virbian:~$ sudo ip addr add 192.168.1.1/24 dev enp-loc1
user@virbian:~$

```

## Virbian2

```
user@virbian:~$ sudo ip link set up dev enp-loc1
user@virbian:~$ sudo ip addr add 192.168.1.2/24 dev enp-loc1
```

## Sprawdzenie pingiem

Z każdej maszyny sprawdziłem osiągalność interfejsów bezpośrednio połączonych maszyn. Poniżej przykład sprawdzenia osiągalności Virbiana2 z Virbiana1.

```
user@virbian:~$ ping 192.168.1.2
PING 192.168.1.2 (192.168.1.2) 56(84) bytes of data.
64 bytes from 192.168.1.2: icmp_seq=1 ttl=64 time=0.315 ms
64 bytes from 192.168.1.2: icmp_seq=2 ttl=64 time=0.176 ms
64 bytes from 192.168.1.2: icmp_seq=3 ttl=64 time=0.182 ms
64 bytes from 192.168.1.2: icmp_seq=4 ttl=64 time=0.197 ms
^C
--- 192.168.1.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3067ms
rtt min/avg/max/mdev = 0.176/0.217/0.315/0.056 ms
```

## Krok 4 - konfiguracja routingu

### Virbian0

Określamy, że do sieci 192.168.1.0/24 można dostać się wysyłając pakiety do 192.168.0.2

```
user@virbian:~$ sudo ip route add 192.168.1.0/24 via 192.168.0.2
```

### Virbian1

Ustawiamy bramę domyślną równą 192.168.0.1

```
user@virbian:~$ sudo ip route add default via 192.168.0.1
user@virbian:~$
```

### Virbian2

Ustawiamy bramę domyślną równą 192.168.1.1

```
user@virbian:~$ sudo ip route add default via 192.168.1.1
user@virbian:~$
```

## Krok 5 - test osiągalności interfejsów

Sprawdzimy czy z każdej maszyny są dostępne wszystkie interfejsy. Poniżej przykład sprawdzenia pingiem osiągalności Virbiana2 z Virbiana0, czyli IP 192.168.1.2 z Virbian0.

```

user@virbian:~$ ping 192.168.1.2
PING 192.168.1.2 (192.168.1.2) 56(84) bytes of data.
64 bytes from 192.168.1.2: icmp_seq=1 ttl=63 time=0.289 ms
64 bytes from 192.168.1.2: icmp_seq=2 ttl=63 time=0.334 ms
64 bytes from 192.168.1.2: icmp_seq=3 ttl=63 time=0.326 ms
64 bytes from 192.168.1.2: icmp_seq=4 ttl=63 time=0.332 ms
64 bytes from 192.168.1.2: icmp_seq=5 ttl=63 time=0.335 ms
^C
--- 192.168.1.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4073ms
rtt min/avg/max/mdev = 0.289/0.323/0.335/0.017 ms
user@virbian:~$

```

Trasa z maszyny Virbian2 do maszyny Virbian0 przy pomocy traceroute.

```

user@virbian:~$ sudo traceroute -I 192.168.0.1
traceroute to 192.168.0.1 (192.168.0.1), 30 hops max, 60 byte packets
 1  192.168.1.1 (192.168.1.1)  0.166 ms  0.147 ms *
 2  * * *
 3  * * *
 4  * * *
 5  * * *
 6  * 192.168.0.1 (192.168.0.1)  0.329 ms  0.228 ms
user@virbian:~$

```

## Krok 6 - ping 8.8.8.8

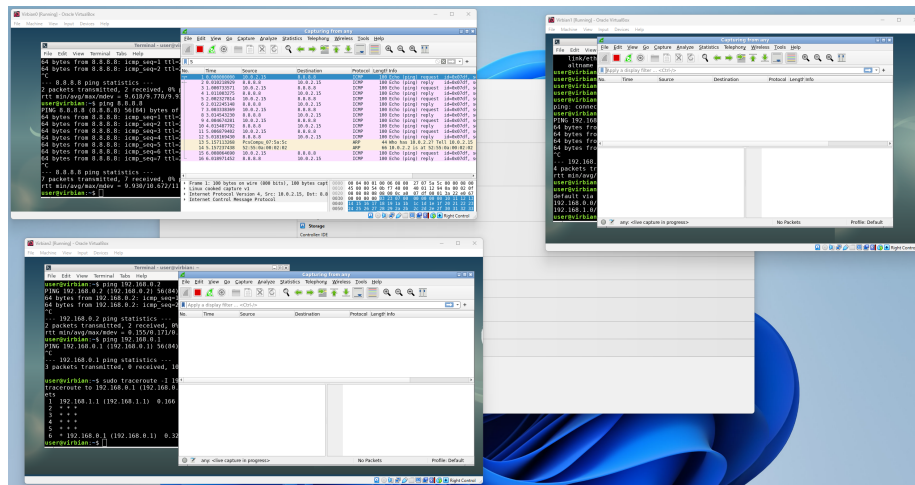
### Virbian0

Otrzymujemy odpowiedź oraz podglądając wiresharki odpowiedź widzimy wyłącznie na maszynie Virbian0.

```

user@virbian:~$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=255 time=10.2 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=255 time=10.3 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=255 time=9.93 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=255 time=11.2 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=255 time=10.8 ms
64 bytes from 8.8.8.8: icmp_seq=6 ttl=255 time=11.3 ms
64 bytes from 8.8.8.8: icmp_seq=7 ttl=255 time=10.9 ms
^C
--- 8.8.8.8 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6005ms
rtt min/avg/max/mdev = 9.930/10.672/11.306/0.490 ms
user@virbian:~$

```



## Virbian2

W przypadku pingowania z Virbiana2 zapytanie dochodzi do do maszyny Virbian0 co obserwujemy w wiresharku. Nie otrzymujemy za to odpowiedzi. Wynika to z tego, że Virbian2 jest w lokalnej sieci, google nie wie o jego istnieniu i nie potrafi odesłać odpowiedzi. Żeby odpowiedź wracała do Virbiana2 trzeba by powiadomić google o konfiguracji naszej sieci.

```
user@virbian:~$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
^C
--- 8.8.8.8 ping statistics ---
7 packets transmitted, 0 received, 100% packet loss, time 6123ms
```

