

Sieci komputerowe

Warsztaty 5

Mateusz Markiewicz

29 kwietnia 2020

1 Pierwsze zadanie do zaprezentowania

Pierwszym etapem zadania było stworzenie 2 maszyn wirtualnych z odpowiednią konfiguracją sieciową (posiadały one kartę sieciową połączoną z sieciami zewnętrznymi local0).

Następnie nazwałem interfejsy tych maszyn, aktywowałem je i nadałem im adresy zgodnie z wytycznymi, użyłem w tym celu następujących poleceń:

- V1#> ip link set enp0s3 name enp0
- V1#> ip link set up dev enp0
- V1#> ip addr add 192.168.0.1/24 dev enp0

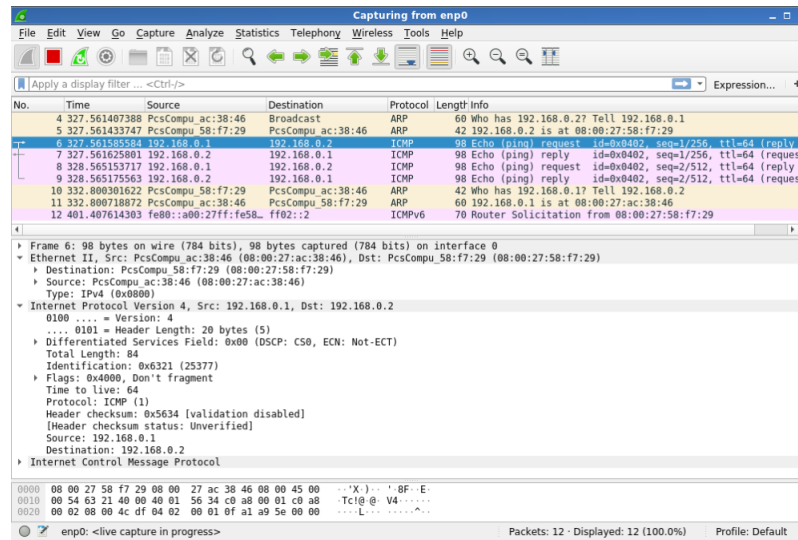
- V2#> ip link set enp0s3 name enp0
- V2#> ip link set up dev enp0
- V2#> ip addr add 192.168.0.2/24 dev enp0

Następnie poleceniem ip link zobaczyłem adresy MAC na obu maszynach, są one następujące:

```
user@virbian:~$ ip link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode
  DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: enp0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast s
  tate UP mode DEFAULT group default qlen 1000
    link/ether 08:00:27:ac:38:46 brd ff:ff:ff:ff:ff:ff

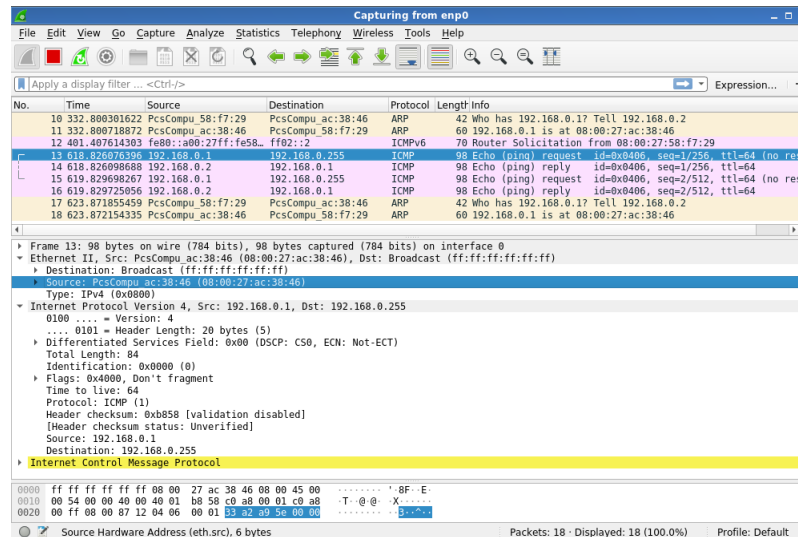
user@virbian:~$ ip link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT
  group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: enp0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP mo
  de DEFAULT group default qlen 1000
    link/ether 08:00:27:58:f7:29 brd ff:ff:ff:ff:ff:ff
```

Następnie z maszyny Virbian1 pingowałem maszynę Virbian2. W rezultacie na maszynie Virbian2 za pomocą Wiresharku zobaczyłem następujące pakiety:



Jak widać w ramce Ethernetowej w polach nadawcy i odbiorcy znajdują się adresy MAC nadawcy oraz odbiorcy, w pakiecie IP w polach nadawcy oraz odbiorcy znajdują się ich adresy IP, które wcześniej przypisaliśmy.

Następnie z maszyny Virbian1 pingowałem adres rozgłoszeniowy i tym razem za pomocą Wiresharku zobaczyłem następujące pakiety:

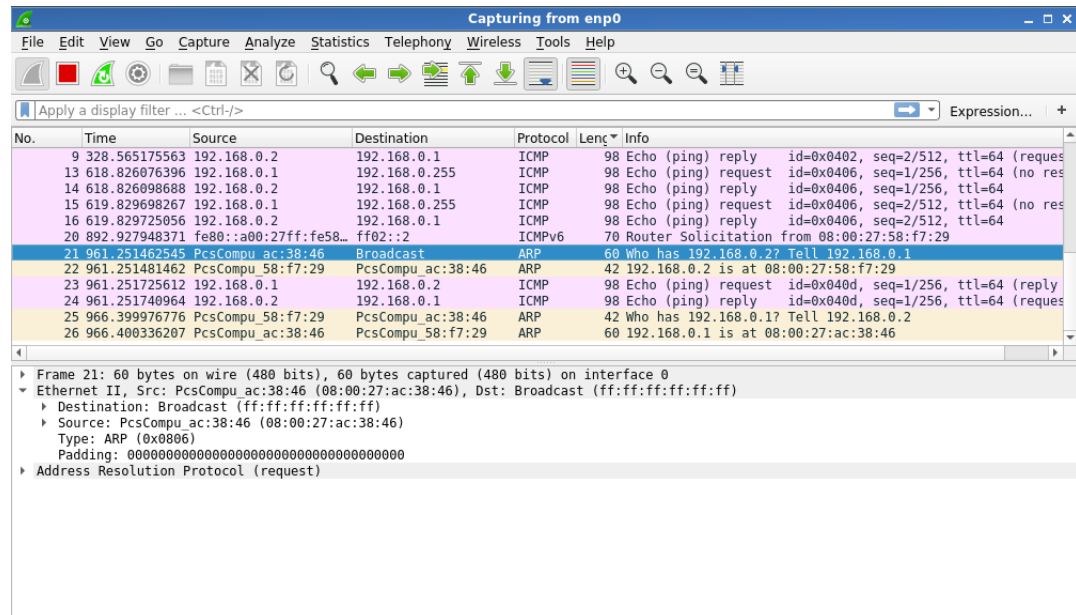


Tym razem w ramce Ethernetowej docelowy adres MAC to ff:ff:ff:ff:ff:ff, a adres IP to adres rozgłoszeniowy tej sieci (który pingowaliśmy), czyli 192.168.0.255.

Następnie na obu maszynach zobaczyłem a następnie wyczyściłem tablicę ARP za pomocą poleceń:

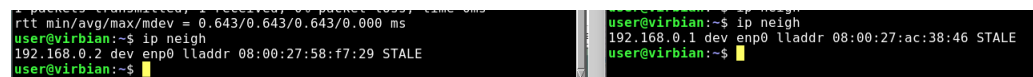
- V1#> ip neigh
- V1#> ip neigh flush all
- V2#> ip neigh
- V2#> ip neigh flush all

Następnie ponownie z maszyny Virbian1 pingowałem maszynę Virbian2, po czym w Wiresharku zaobserwowałem następujące zapytania:



Jak widać Virbian1 pyta o adres MAC maszyny Virbian2 (a dokładniej karty o IP 192.168.0.2) i prosi o odesłanie do niej takiej informacji. Virbian2 jej odpowiada wysyłając odpowiedni adres MAC.

Po tym wszystkim tablice ARP wyglądały następująco:



Jak widać po tej wymianie wiadomości obie maszyny uzyskały informacje o adresie MAC drugiej maszyny, co zapisały w tablicy ARP.

Następnie przyjrzałem się wiadomościom ARP które wyglądały następująco:

The image displays two screenshots of the Wireshark network protocol analyzer. The top screenshot shows packet 21, an ARP request from 192.168.0.1 to 192.168.0.2. The bottom screenshot shows packet 22, the corresponding ARP reply from 192.168.0.2 back to 192.168.0.1. Both packets are Ethernet II frames with an ARP payload.

Packet 21: ARP Request

No.	Time	Source	Destination	Protocol	Length	Info
21	961.251462545	PcsCompu_ac:38:46	Broadcast	ARP	60	Who has 192.168.0.2? Tell 192.168.0.1

Packet 22: ARP Reply

No.	Time	Source	Destination	Protocol	Length	Info
22	961.251481462	PcsCompu_58:f7:29	PcsCompu_ac:38:46	ARP	42	192.168.0.2 is at 08:00:27:58:f7:29

The bottom screenshot also shows the packet details for packet 22, confirming it is an ARP reply from 192.168.0.2 to 192.168.0.1.

Zapytanie ARP zawiera informacje kto pyta o dany adres MAC (Sender MAC address oraz Sender IP address). Zawiera również informacje o IP sieci, której MAC adres chcemy poznać (Target IP address). W zapytaniu jest też pole Target MAC address, które jest jeszcze puste. Zapytania wysyłane są na adres rozgłoszeniowy (destination: Broadcast). Nie wiemy do jakiego konkretnego komputera mielibyśmy wysłać takie zapytanie. Odpowiedzi wysyłane są do konkretnego komputera, w tym przypadku wiemy już kto pytał o adres MAC maszyny o danym IP, stąd możemy przesłać odpowiedź do tej konkretnej maszyny.

2 Drugie zadanie do zaprezentowania

Zacząłem od dodania 2 nowych maszyn oraz skonfigurowaniu wszystkich maszyn za pomocą poleceń:

- V1#> ip link set enp0s3 name enp0
- V1#> ip link set up dev enp0
- V1#> ip addr add 192.168.1.1/24 dev enp0

- V2#> ip link set enp0s3 name enp0
- V2#> ip link set up dev enp0
- V2#> ip addr add 192.168.1.2/25 dev enp0

- V3#> ip link set enp0s3 name enp0
- V3#> ip link set up dev enp0
- V3#> ip addr add 192.168.1.129/24 dev enp0

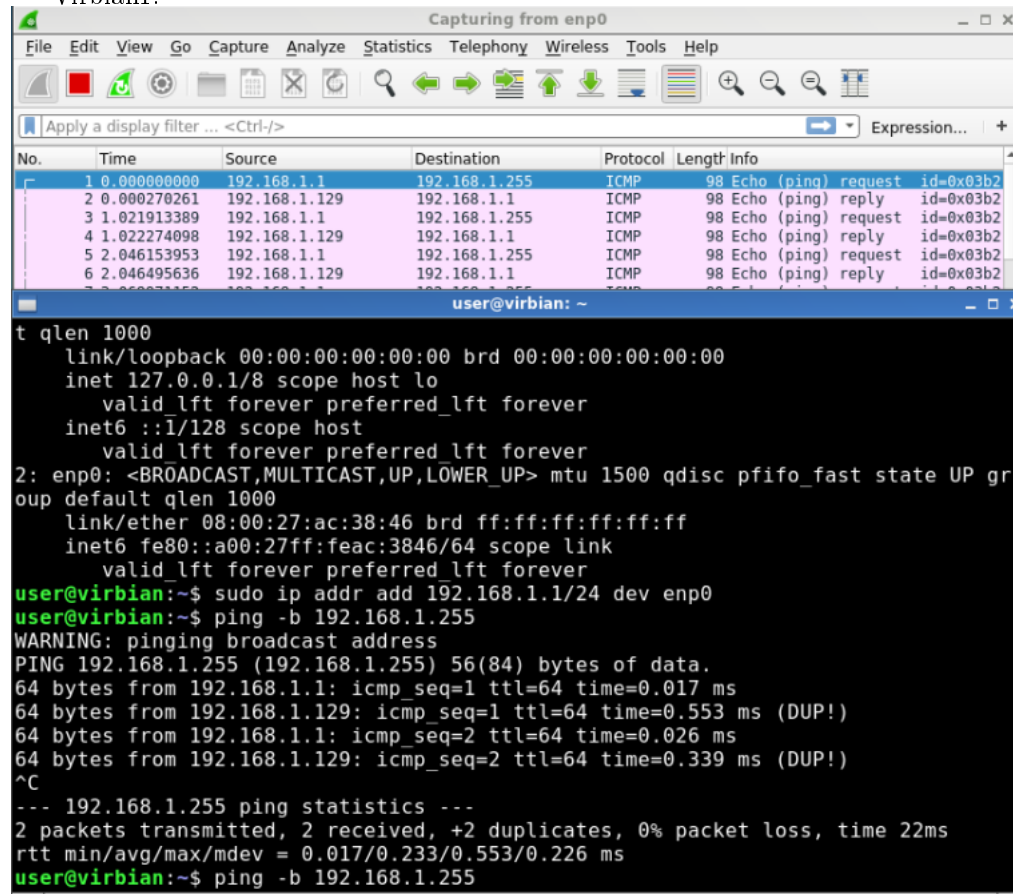
- V4#> ip link set enp0s3 name enp0
- V4#> ip link set up dev enp0
- V4#> ip addr add 192.168.1.130/25 dev enp0

Zakresy sieci są następujące:

	Virbian1	Virbian2	Virbian3	Virbian4
Początek	192.168.1.0	192.168.1.0	192.168.1.0	192.168.1.128
Koniec	192.168.1.255	192.168.1.127	192.168.1.255	192.168.1.255

Następnie z maszyny Virbian1 pingowałem jej adres rozgłoszeniowy po czym zaobserwowałem następujące pakiety za pomocą Wiresharka:

Virbian1:



The screenshot displays two windows from a Linux system named 'Virbian1'.

The top window is Wireshark, capturing traffic on the 'enp0' interface. The packet list shows six ICMP Echo (ping) packets. The first packet is a request from 192.168.1.1 to 192.168.1.255. The subsequent five packets are replies from 192.168.1.129 to 192.168.1.1, with the last one being a duplicate.

The bottom window is a terminal with the following commands and output:

```
user@virbian: ~  
t qlen 1000  
  link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
  inet 127.0.0.1/8 scope host lo  
    valid_lft forever preferred_lft forever  
  inet6 ::1/128 scope host  
    valid_lft forever preferred_lft forever  
2: enp0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP gr  
oup default qlen 1000  
  link/ether 08:00:27:ac:38:46 brd ff:ff:ff:ff:ff:ff  
  inet6 fe80::a00:27ff:feac:3846/64 scope link  
    valid_lft forever preferred_lft forever  
user@virbian:~$ sudo ip addr add 192.168.1.1/24 dev enp0  
user@virbian:~$ ping -b 192.168.1.255  
WARNING: pinging broadcast address  
PING 192.168.1.255 (192.168.1.255) 56(84) bytes of data:  
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=0.017 ms  
64 bytes from 192.168.1.129: icmp_seq=1 ttl=64 time=0.553 ms (DUP!)  
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=0.026 ms  
64 bytes from 192.168.1.129: icmp_seq=2 ttl=64 time=0.339 ms (DUP!)  
^C  
--- 192.168.1.255 ping statistics ---  
2 packets transmitted, 2 received, +2 duplicates, 0% packet loss, time 22ms  
rtt min/avg/max/mdev = 0.017/0.233/0.553/0.226 ms  
user@virbian:~$ ping -b 192.168.1.255
```

Virbian2:

Wireshark interface showing a live capture on interface enp0. The packet list displays 9 packets, all ICMP Echo (ping) requests from 192.168.1.1 to 192.168.1.255. The packet details pane shows the structure of the first packet: Ethernet II, Internet Protocol Version 4, and Internet Control Message Protocol. The packet bytes pane shows the raw data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.1.1	192.168.1.255	ICMP	98	Echo (ping) request id=0x03b2, seq=
2	1.021936368	192.168.1.1	192.168.1.255	ICMP	98	Echo (ping) request id=0x03b2, seq=
3	2.046138026	192.168.1.1	192.168.1.255	ICMP	98	Echo (ping) request id=0x03b2, seq=
4	3.069958461	192.168.1.1	192.168.1.255	ICMP	98	Echo (ping) request id=0x03b2, seq=
5	4.093469418	192.168.1.1	192.168.1.255	ICMP	98	Echo (ping) request id=0x03b2, seq=
6	5.117339763	192.168.1.1	192.168.1.255	ICMP	98	Echo (ping) request id=0x03b2, seq=
7	6.142748933	192.168.1.1	192.168.1.255	ICMP	98	Echo (ping) request id=0x03b2, seq=
8	7.165925562	192.168.1.1	192.168.1.255	ICMP	98	Echo (ping) request id=0x03b2, seq=
9	60.965809329	fe80::a00:27ff:febd::	ff02::2	ICMPv6	70	Router Solicitation from 08:00:27:fc

Frame 1: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0
 Ethernet II, Src: PcsCompu ac:38:46 (08:00:27:ac:38:46), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
 Internet Protocol Version 4, Src: 192.168.1.1, Dst: 192.168.1.255
 Internet Control Message Protocol

enp0: <live capture in progress> Packets: 9 · Displayed: 9 (100.0%) Profile: Default

Virbian3:

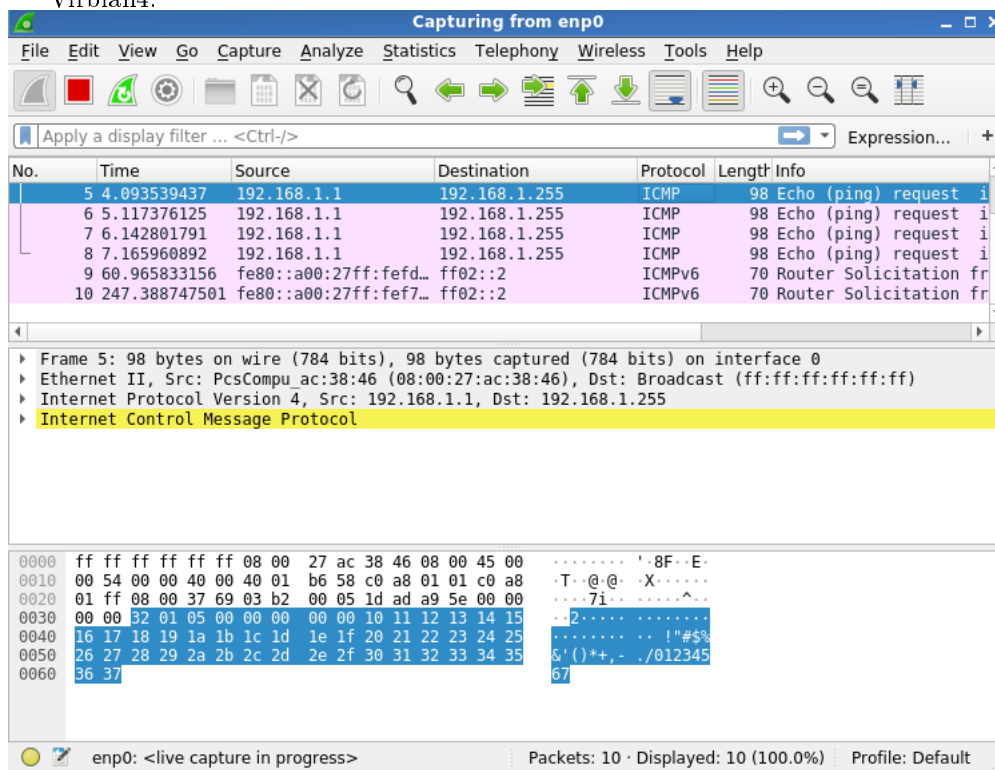
Wireshark interface showing a live capture on interface enp0. The packet list displays 20 packets, including ICMP Echo (ping) requests and replies, ARP requests, and Router Solicitation. The packet details pane shows the structure of the 15th packet: Ethernet II, Internet Protocol Version 4, and Internet Control Message Protocol. The packet bytes pane shows the raw data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
12	5.117406961	192.168.1.129	192.168.1.1	ICMP	98	Echo (ping) reply id=0x03b2, seq=
13	5.158314667	PcsCompu fd:e6:39	PcsCompu ac:38:46	ARP	42	Who has 192.168.1.1? Tell 192.168.1.1
14	5.158655936	PcsCompu ac:38:46	PcsCompu fd:e6:39	ARP	60	192.168.1.1 is at 08:00:27:ac:38:46
15	6.142857050	192.168.1.1	192.168.1.255	ICMP	98	Echo (ping) request id=0x03b2, seq=
16	6.142881725	192.168.1.129	192.168.1.1	ICMP	98	Echo (ping) reply id=0x03b2, seq=
17	7.165974986	192.168.1.1	192.168.1.255	ICMP	98	Echo (ping) request id=0x03b2, seq=
18	7.166002214	192.168.1.129	192.168.1.1	ICMP	98	Echo (ping) reply id=0x03b2, seq=
19	60.965671212	fe80::a00:27ff:febd::	ff02::2	ICMPv6	70	Router Solicitation from 08:00:27:fc
20	247.388931589	fe80::a00:27ff:fe7::	ff02::2	ICMPv6	70	Router Solicitation from 08:00:27:fc

Frame 15: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0
 Ethernet II, Src: PcsCompu ac:38:46 (08:00:27:ac:38:46), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
 Internet Protocol Version 4, Src: 192.168.1.1, Dst: 192.168.1.255
 Internet Control Message Protocol

enp0: <live capture in progress> Packets: 20 · Displayed: 20 (100.0%) Profile: Default

Virbian4:



Wszystkie maszyny otrzymały komunikat echo request, ponieważ są połączone jedną siecią wewnętrzną local0. W odpowiedzi maszyna Virbian3 wysłała echo replay. Virbian1 nie wysłała odpowiedzi sama do siebie (choć ping dostał od niej wszystkie informacje, odbyło się to za pomocą innej warstwy). Virbian2 nie wysłała odpowiedzi, ponieważ ma inny adres rozgłoszeniowy (ten był spoza jej sieci). Virbian4 nie wysłała odpowiedzi, chociaż ma taki sam adres rozgłoszeniowy, ale nadawca echo request jest spoza jej sieci. Wszystkie wysyłane odpowiedzi dochodziły do nadawcy.

Następnie powtórzyłem to dla pozostałych maszyn:

- Pingując z maszyny Virbian2 echo request dotarł do wszystkich maszyn, ale nikt nie odesłał odpowiedzi, stąd Virbian2 w pingu miała informacje tylko od siebie samej. Adres rozgłoszeniowy pozostałych maszyn jest inny, więc nie odpowiadały.
- Pingując z maszyny Virbian3 echo request dotarł do wszystkich maszyn, odpowiedziały maszyny Virbian1 oraz Virbian4 (mają ten sam adres rozgłoszeniowy, adres Virbian3 należy do ich sieci).
- Pingując z maszyny Virbian4 echo request dotarł do wszystkich maszyn,

odpowiedziały maszyny Virbian1 oraz Virbian3 (mają ten sam adres rozgłoszeniowy, adres Virbian4 należy do ich sieci.