

Warsztaty 5

Dawid Dieu
302052

Zadanie do zaprezentowania (3 pkt.)

W tej części przyjrzymy się dokładniej warstwie łącza danych i współpracy pomiędzy tą warstwą a warstwą sieciową.

- Uruchom dwie maszyny wirtualne Virbian1 i Virbian2, każdą z jedną kartą sieciową zmostkowaną z interfejsem local0. Zmień nazwę wirtualnego interfejsu w każdej z maszyn na enp0.
- Aktywuj interfejsy enp0 obu maszyn i przypisz im adresy IP równe odpowiednio 192.168.0.1/24 i 192.168.0.2/24. Na każdej maszynie uruchom Wiresharka. Uwaga: włącz obserwację wyłącznie interfejsu enp0: jeśli włączysz obserwację wszystkich interfejsów podgląd warstwy łącza danych będzie utrudniony.

o V1:

```
sudo ip link set enp0s3 name enp0
sudo ip link set up dev enp0
sudo ip addr add 192.168.0.1/24 dev enp0
```

o V2:

```
sudo ip link set enp0s3 name enp0
sudo ip link set up dev enp0
sudo ip addr add 192.168.0.2/24 dev enp0
```

- Poleceniem `ip link` wyświetl adresy MAC kart sieciowych na obu maszynach. Z maszyny Virbian1 pingnij maszynę Virbian2 i obejrzyj przesyłane ramki w Wiresharku. Jakie są pola nadawcy i odbiorcy ramki ethernetowej? A jakie są pola nadawcy i odbiorcy zawartego w niej pakietu IP?

o ping V1 --> V3

```
user@virbian:~$ ping 192.168.0.2
PING 192.168.0.2 (192.168.0.2) 56(84) bytes of data:
64 bytes from 192.168.0.2: icmp_seq=1 ttl=64 time=0.815 ms
64 bytes from 192.168.0.2: icmp_seq=2 ttl=64 time=0.480 ms
64 bytes from 192.168.0.2: icmp_seq=3 ttl=64 time=0.601 ms
64 bytes from 192.168.0.2: icmp_seq=4 ttl=64 time=0.577 ms
64 bytes from 192.168.0.2: icmp_seq=5 ttl=64 time=0.357 ms
```

o Wireshark + Frame

The screenshot shows the Wireshark interface with the title bar 'Capturing from enp0'. The menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, and Help. The toolbar contains various icons for file operations, capture control, and analysis. The display filter is set to 'Apply a display filter ... <Ctrl-/>'. The packet list shows seven packets, all ICMP Echo (ping) requests and replies between 192.168.0.1 and 192.168.0.2. The packet details pane for the first packet shows the following structure:

- Frame 1: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0
- Ethernet II, Src: PcsCompu_c2:e4:0e (08:00:27:c2:e4:0e), Dst: PcsCompu_9c:50:a0 (08:00:27:9c:50:a0)
- Internet Protocol Version 4, Src: 192.168.0.1, Dst: 192.168.0.2
- Internet Control Message Protocol

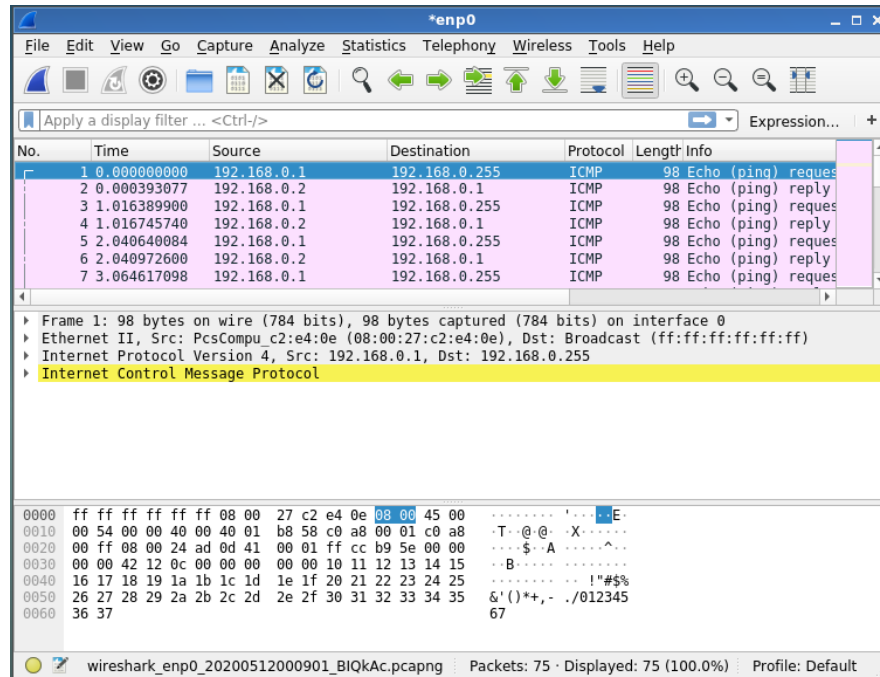
The packet bytes pane shows the raw data in hexadecimal and ASCII. The first few bytes are 0000 08 00 27 9c 50 a0 08 00 27 c2 e4 0e 08 00 45 00, which correspond to the Ethernet II header.

- Z maszyny Virbian1 pingnij adres rozgłoszeniowy 192.168.0.255. Jakie są tym razem pola nadawcy i odbiorcy ramki ethernetowej? A jakie są pola nadawcy i odbiorcy zawartego w niej pakietu IP?

- Ping

```
user@virbian:~$ ping 192.168.0.255 -b
WARNING: pinging broadcast address
PING 192.168.0.255 (192.168.0.255) 56(84) bytes of data.
64 bytes from 192.168.0.1: icmp_seq=1 ttl=64 time=0.017 ms
64 bytes from 192.168.0.2: icmp_seq=1 ttl=64 time=0.409 ms (DUP!)
64 bytes from 192.168.0.1: icmp_seq=2 ttl=64 time=0.023 ms
64 bytes from 192.168.0.2: icmp_seq=2 ttl=64 time=0.416 ms (DUP!)
^C
```

- Wireshark + Frame



- W maszynie Virbian1 obejrzyj tablicę ARP poleceniem
v1\$> ip neigh i usuń z niej wszystkie wpisy poleceniem v1\$> ip neigh flush all
Wykonaj to samo polecenie w maszynie Virbian2.

- V1\$> ip neigh

```
user@virbian:~$ ip neigh
192.168.0.2 dev enp0 lladdr 08:00:27:9c:50:a0 STALE
user@virbian:~$
```

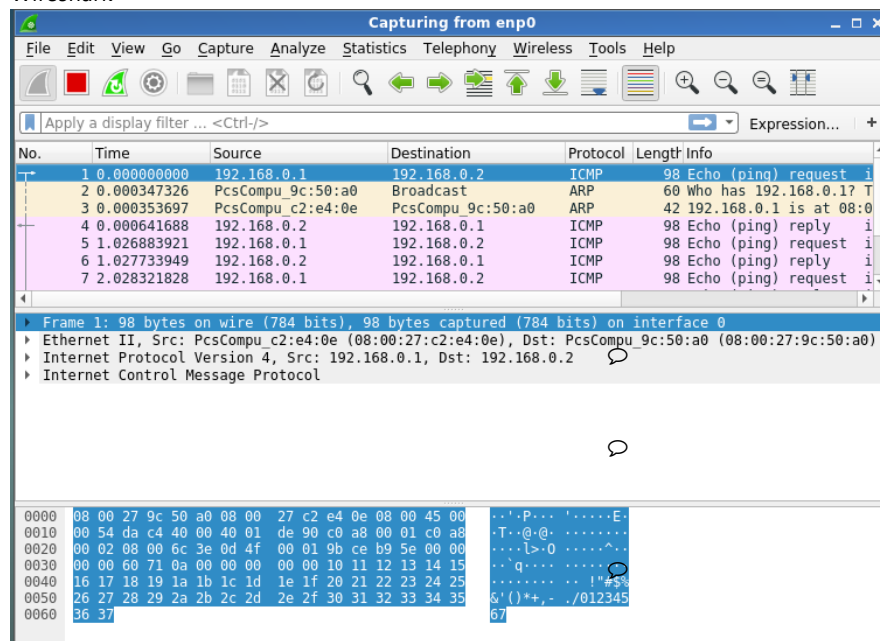
- V1\$> ip neigh flush all

(https://i.imgur.com/5gLLqZE.png)

- Analogicznie na V2

- Z maszyny Virbian1 pingnij maszynę Virbian2. W Wiresharku zaobserwuj, że maszyna najpierw wysyła zapytanie ARP, otrzymuje na nie odpowiedź, a następnie wysyła komunikaty ICMP echo i otrzymuje na nie odpowiedzi. Jak zmienił się stan tablicy ARP obu maszyn

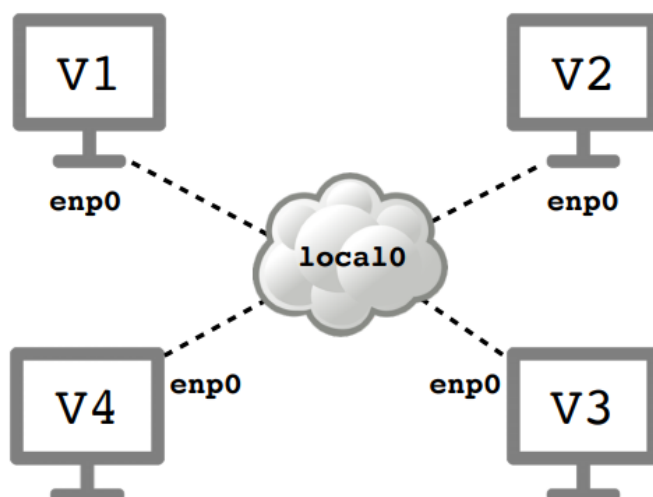
o Wireshark



- Przyjrzyj się dokładnie przesyłanemu w poprzednim punkcie zapytaniu i odpowiedzi ARP. Odpowiedz na następujące pytania:
 1. Co jest danymi ramki w przypadku zapytań ARP?
 - Adres, o który się pytamy oraz adres na który ma odpowiedzieć
 2. Czy zapytania ARP są wysyłane do konkretnego komputera czy na adres rozgłoszeniowy?
 - Rozgłoszeniowy
 3. Czy odpowiedzi ARP są wysyłane do konkretnego komputera czy na adres rozgłoszeniowy?
 - Do konkretnego komputera

Zadanie do zaprezentowania (2 pkt.)

Uruchom dwie dodatkowe maszyny wirtualne Virbian3 i Virbian4, każdą z jedną kartą sieciową zmostkowaną z interfejsem local0. Zmień nazwę wirtualnego interfejsu w każdej z maszyn na enp0, otrzymując konfigurację z poniższego rysunku.



Zobaczmy teraz, do czego prowadzi mieszanie wielu sieci IP w jednej sieci Ethernet. Włącz na wszystkich komputerach Wiresharka, jeśli jeszcze nie jest włączony.

- Przypisz interfejsom enp0 maszyn wirtualnych następujące adresy:
 - Virbian1 : 192.168.1.1/24
 - Virbian2 : 192.168.1.2/25
 - Virbian3 : 192.168.1.129/24
 - Virbian4 : 192.168.1.130/25

```
sudo ip link set enp0s3 name enp0
sudo ip link set up dev enp0
sudo ip addr add <IP> dev enp0
```

- Zauważ, że maszyny leżą w jednej sieci warstwy drugiej, ale w trzech różnych podsieciach IP (różnych sieciach warstwy trzeciej). Jakie są zakresy adresów tych sieci?

o V1

Address (Host or Network)	Netmask (i.e. 24)	Netmask for sub/supernet
192.168.1.1	/ 24	move to:
<input type="button" value="Calculate"/>	<input type="button" value="Help"/>	

```

Address: 192.168.1.1      11000000.10101000.00000001 .00000001
Netmask: 255.255.255.0 = 24 11111111.11111111.11111111 .00000000
Wildcard: 0.0.0.255      00000000.00000000.00000000 .11111111
=>
Network: 192.168.1.0/24   11000000.10101000.00000001 .00000000 (Class C)
Broadcast: 192.168.1.255 11000000.10101000.00000001 .11111111
HostMin: 192.168.1.1     11000000.10101000.00000001 .00000001
HostMax: 192.168.1.254   11000000.10101000.00000001 .11111110
Hosts/Net: 254           (Private Internet)

```

o V2

Address (Host or Network)	Netmask (i.e. 24)	Netmask for sub/supernet
192.168.1.2	/ 25	move to:
<input type="button" value="Calculate"/>	<input type="button" value="Help"/>	

```

Address: 192.168.1.2      11000000.10101000.00000001.0 0000010
Netmask: 255.255.255.128 = 25 11111111.11111111.11111111.1 0000000
Wildcard: 0.0.0.127      00000000.00000000.00000000.0 1111111
=>
Network: 192.168.1.0/25   11000000.10101000.00000001.0 0000000 (Class C)
Broadcast: 192.168.1.127 11000000.10101000.00000001.0 1111111
HostMin: 192.168.1.1     11000000.10101000.00000001.0 0000001
HostMax: 192.168.1.126   11000000.10101000.00000001.0 1111110
Hosts/Net: 126           (Private Internet)

```

o V3

Address (Host or Network)	Netmask (i.e. 24)	Netmask for sub/supernet
192.168.1.129	/ 24	move to:
<input type="button" value="Calculate"/>	<input type="button" value="Help"/>	

```

Address: 192.168.1.129    11000000.10101000.00000001 .10000001
Netmask: 255.255.255.0 = 24 11111111.11111111.11111111 .00000000
Wildcard: 0.0.0.255      00000000.00000000.00000000 .11111111
=>
Network: 192.168.1.0/24   11000000.10101000.00000001 .00000000 (Class C)
Broadcast: 192.168.1.255 11000000.10101000.00000001 .11111111
HostMin: 192.168.1.1     11000000.10101000.00000001 .00000001
HostMax: 192.168.1.254   11000000.10101000.00000001 .11111110
Hosts/Net: 254           (Private Internet)

```

o V4

Address (Host or Network)	Netmask (i.e. 24)	Netmask for sub/supernet
192.168.1.130	/ 25	move to:
<input type="button" value="Calculate"/>	<input type="button" value="Help"/>	

```

Address: 192.168.1.130    11000000.10101000.00000001.1 0000010
Netmask: 255.255.255.128 = 25 11111111.11111111.11111111.1 0000000
Wildcard: 0.0.0.127      00000000.00000000.00000000.0 1111111
=>
Network: 192.168.1.128/25 11000000.10101000.00000001.1 0000000 (Class C)
Broadcast: 192.168.1.255 11000000.10101000.00000001.1 1111111
HostMin: 192.168.1.129   11000000.10101000.00000001.1 0000001
HostMax: 192.168.1.254   11000000.10101000.00000001.1 1111110
Hosts/Net: 126           (Private Internet)

```

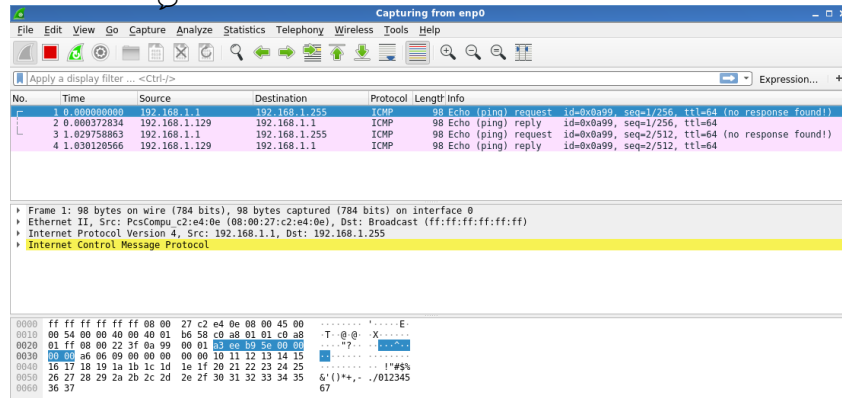
- Z maszyny Virbian1 pingnij jej adres rozgłoszeniowy, a następnie odpowiedz na następujące pytania:

o V1

■ Ping

```
user@virbian:~$ ping 192.168.1.255 -b
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=0.014 ms
64 bytes from 192.168.1.129: icmp_seq=1 ttl=64 time=0.396 ms (DUP!)
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=0.020 ms
64 bytes from 192.168.1.129: icmp_seq=2 ttl=64 time=0.379 ms (DUP!)
```

■ Wireshark

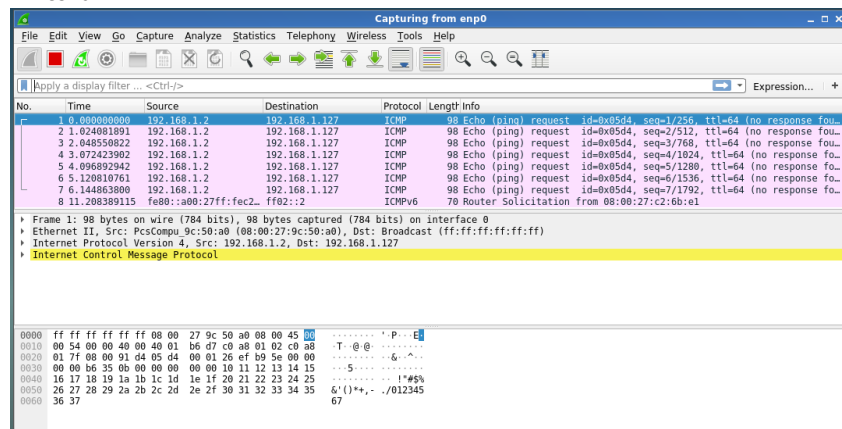


○ V2

■ Ping

```
user@virbian:~$ ping 192.168.1.127 -b
WARNING: pinging broadcast address
PING 192.168.1.127 (192.168.1.127) 56(84) bytes of data.
64 bytes from 192.168.1.2: icmp_seq=1 ttl=64 time=0.014 ms
64 bytes from 192.168.1.2: icmp_seq=2 ttl=64 time=0.018 ms
64 bytes from 192.168.1.2: icmp_seq=3 ttl=64 time=0.019 ms
64 bytes from 192.168.1.2: icmp_seq=4 ttl=64 time=0.030 ms
64 bytes from 192.168.1.2: icmp_seq=5 ttl=64 time=0.019 ms
64 bytes from 192.168.1.2: icmp_seq=6 ttl=64 time=0.018 ms
64 bytes from 192.168.1.2: icmp_seq=7 ttl=64 time=0.020 ms
```

■ Wireshark

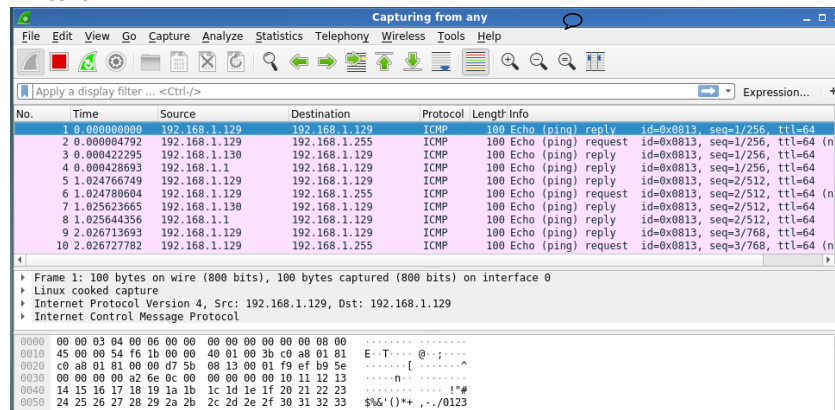


○ V3

■ Ping

```
user@virbian:~$ ping 192.168.1.255 -b
WARNING: pinging broadcast address
PING 192.168.1.255 (192.168.1.255) 56(84) bytes of data.
64 bytes from 192.168.1.129: icmp_seq=1 ttl=64 time=0.016 ms
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=0.404 ms (DUP!)
64 bytes from 192.168.1.130: icmp_seq=1 ttl=64 time=0.635 ms (DUP!)
64 bytes from 192.168.1.129: icmp_seq=2 ttl=64 time=0.041 ms
64 bytes from 192.168.1.130: icmp_seq=2 ttl=64 time=0.655 ms (DUP!)
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=0.672 ms (DUP!)
```

Wireshark

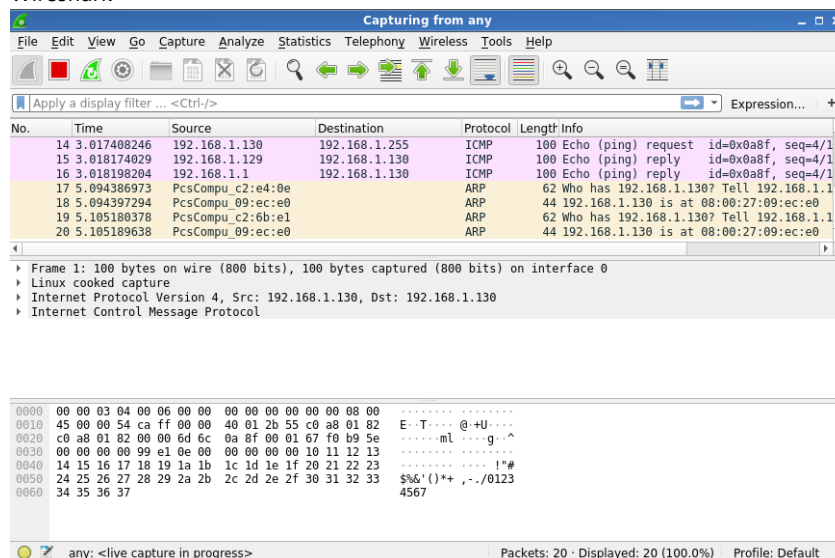


- o V4

Ping

```
user@virbian:~$ ping 192.168.1.255 -b
WARNING: pingng broadcast address
PING 192.168.1.255 (192.168.1.255) 56(84) bytes of data.
64 bytes from 192.168.1.130: icmp_seq=1 ttl=64 time=0.018 ms
64 bytes from 192.168.1.129: icmp_seq=1 ttl=64 time=0.402 ms (DUP!)
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=0.409 ms (DUP!)
64 bytes from 192.168.1.130: icmp_seq=2 ttl=64 time=0.054 ms
64 bytes from 192.168.1.129: icmp_seq=2 ttl=64 time=0.848 ms (DUP!)
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=0.869 ms (DUP!)
64 bytes from 192.168.1.130: icmp_seq=3 ttl=64 time=0.054 ms
```

Wireshark



Pytania:

1. Które maszyny otrzymały komunikat ICMP echo request? Które nie otrzymały i dlaczego?
 - o Wszystkie otrzymały, bo są w tej samej sieci.
 2. Które maszyny wysłały w odpowiedzi komunikat ICMP echo reply? Które nie wysłały i dlaczego?
 - o Virban 1 nie wysła, bo jest w innej warstwie
 - o Virban 2 wysła tylko request'y, bo jest ma inny broadcast
 - o Virban 3 wysła
 - o Virban 4 nie wysła, bo nadawca requesta jest w innej wieci
 3. Które odpowiedzi dotarły do maszyny Virbian1 ? Które nie dotarły i dlaczego?
 - o Wszystkie dotarły
- Wykonaj powyższy punkt, ale z maszyny Virbian2, z maszyny Virbian3, a na końcu z maszyny Virbian4.
 - Zdekonfiguruj interfejsy enp0 i wyłącz wszystkie maszyny.