

**Міністерство освіти і науки України
Національний технічний університет України «КП» імені Ігоря Сікорського
Кафедра обчислювальної техніки ФІОТ**

**ЗВІТ
з лабораторної роботи №1
з навчальної дисципліни «Тестування та контроль якості (QA) вбудованих систем»**

**Тема:
НАЛАШТУВАННЯ МЕРЕЖНОГО ОТОЧЕННЯ ТА ТЕСТУВАННЯ ПРОТОКОЛУ
ARP**

Виконав:
студент IV курсу ФІОТ
групи ІВ-93
Залікова №9325
Варіант – 1

Перевірила:
Клименко І.А.

Київ 2022

I. Мета: Навчитися налаштовувати мережне оточення для тестування вбудованих систем та пристроїв IoT. Навчитися використовувати утиліту wireshark для аналізу трафіка в комп'ютерній мережі. Протестувати мережне оточення на канальному рівні моделі OSI.

II. Завдання:

- налаштувати оточення для експериментального дослідження основних процесів та артефактів канального рівня моделі OSI, зокрема форматів фреймів Ethernet, формату MAC адрес, протоколу ARP;
 - експериментально ознайомитися з протоколом передавання службових повідомлень ICMP;
 - навчитися користуватися мережною утилітою ping;
 - з використанням програми wireshark дослідити основні етапи отримання MAC адрес протоколом ARP на канальному рівні моделі OSI.
- QA завдання: QA Embedded Testing на прикладі протоколу ARP
Розроблення ARP Test Cases.

По варіанту:

- 1) Переконатися в тому, що arp-таблиця оновлюється при отриманні arp-reply.

1. На PC2 куди мы будемо відсилати пакети дізнаємося IP адресу через команду **ipconfig** у Windows PowerShell

```
Администратор: Windows PowerShell
PS C:\WINDOWS\system32> ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

    Connection-specific DNS Suffix  . : 
    Link-local IPv6 Address . . . . . : fe80::c106:c32b:fb29:7a5c%14
    IPv4 Address. . . . . : 192.168.31.23
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.31.1

Ethernet adapter VMware Network Adapter VMnet1:

    Connection-specific DNS Suffix  . : 
    Link-local IPv6 Address . . . . . : fe80::90d9:f4d3:c398:8d22%16
    IPv4 Address. . . . . : 192.168.32.1
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 

Ethernet adapter VMware Network Adapter VMnet8:

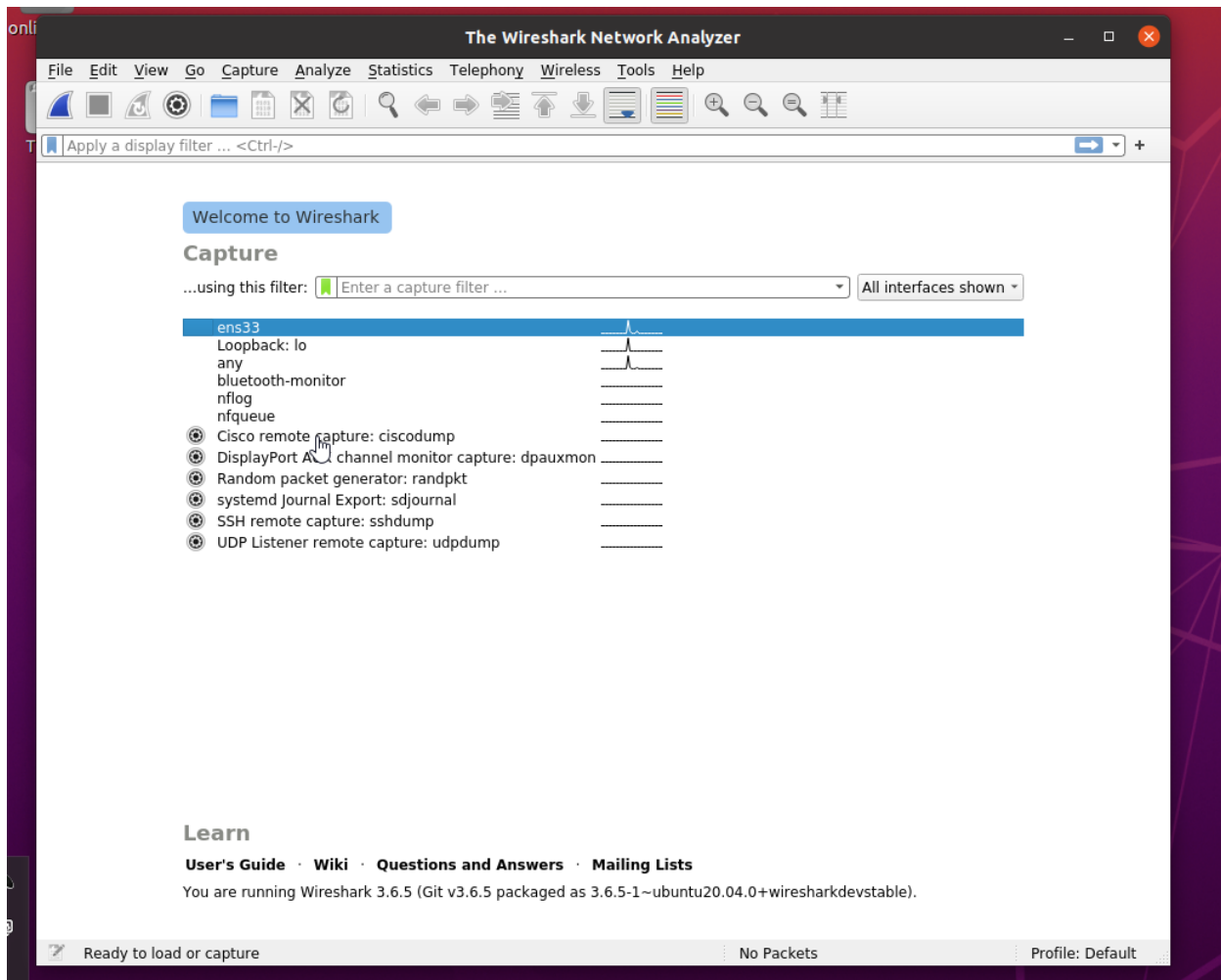
    Connection-specific DNS Suffix  . : 
    Link-local IPv6 Address . . . . . : fe80::fc99:c451:90ae:a25b%3
    IPv4 Address. . . . . : 192.168.61.1
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 
PS C:\WINDOWS\system32>
```

2. Перевіряємо на PC1 **arp** таблицю щоб там не було кешу про PC2 через команду **arp -a** у терміналі

```
onlinegod@ubuntu:~/Desktop$ arp -a
_ gateway (192.168.61.2) at 00:50:56:e6:b2:04 [ether] on ens33
onlinegod@ubuntu:~/Desktop$
```

3. Запускаємо Wireshark на PC1 командою **sudo wireshark** попердньо його встановивши там обновивши як показано у звіті до лабораторної роботи

```
onlinegod@ubuntu: ~/Desktop
onlinegod@ubuntu:~/Desktop$ sudo wireshark
** (wireshark:2740) 19:46:10.291615 [GUI WARNING] -- QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-root'
```



4.Знаючи IP адресу PC2 посилаємо пакети на PC1 командою **ping 192.168.31.23**

```
onlinegod@ubuntu: ~/Desktop
onlinegod@ubuntu:~/Desktop$ ping 192.168.31.23
PING 192.168.31.23 (192.168.31.23) 56(84) bytes of data.
64 bytes from 192.168.31.23: icmp_seq=1 ttl=128 time=0.612 ms
64 bytes from 192.168.31.23: icmp_seq=2 ttl=128 time=0.548 ms
64 bytes from 192.168.31.23: icmp_seq=3 ttl=128 time=0.560 ms
```

5.Обираємо **ens33** та починаємо відстежувати пакети

Capturing from ens33

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|---------------|-----------------|-----------------|----------|--------|--|
| 79 | 383.20966167 | 192.168.61.128 | 192.168.31.23 | ICMP | 98 | Echo (ping) request id=0x0002, seq=2/512, ttl=64 (r |
| 80 | 383.721493661 | 192.168.31.23 | 192.168.61.128 | ICMP | 98 | Echo (ping) reply id=0x0002, seq=2/512, ttl=128 (r |
| 81 | 384.722496840 | 192.168.61.128 | 192.168.31.23 | ICMP | 98 | Echo (ping) request id=0x0002, seq=3/768, ttl=64 (r |
| 82 | 384.723038033 | 192.168.31.23 | 192.168.61.128 | ICMP | 98 | Echo (ping) reply id=0x0002, seq=3/768, ttl=128 (r |
| 83 | 385.736967020 | 192.168.61.128 | 192.168.31.23 | ICMP | 98 | Echo (ping) request id=0x0002, seq=4/1024, ttl=64 (r |
| 84 | 385.737440115 | 192.168.31.23 | 192.168.61.128 | ICMP | 98 | Echo (ping) reply id=0x0002, seq=4/1024, ttl=128 (r |
| 85 | 386.739067514 | 192.168.61.128 | 192.168.31.23 | ICMP | 98 | Echo (ping) request id=0x0002, seq=5/1280, ttl=64 (r |
| 86 | 386.739778805 | 192.168.31.23 | 192.168.61.128 | ICMP | 98 | Echo (ping) reply id=0x0002, seq=5/1280, ttl=128 (r |
| 87 | 387.753263025 | 192.168.61.128 | 192.168.31.23 | ICMP | 98 | Echo (ping) request id=0x0002, seq=6/1536, ttl=64 (r |
| 88 | 387.753815219 | 192.168.31.23 | 192.168.61.128 | ICMP | 98 | Echo (ping) reply id=0x0002, seq=6/1536, ttl=128 (r |
| 89 | 387.944889308 | VMware_9c:2d:10 | VMware_e6:b2:04 | ARP | 42 | Who has 192.168.61.2? Tell 192.168.61.128 |
| 90 | 387.945017207 | VMware_e6:b2:04 | VMware_9c:2d:10 | ARP | 60 | 192.168.61.2 is at 00:50:56:e6:b2:04 |
| 91 | 388.777004586 | 192.168.61.128 | 192.168.31.23 | ICMP | 98 | Echo (ping) request id=0x0002, seq=7/1792, ttl=64 (r |

Frame 1: 217 bytes on wire (1736 bits), 217 bytes captured (1736 bits) on interface ens33, id 0
 Ethernet II, Src: VMware_c0:00:08 (00:50:56:c0:00:08), Dst: IPv4mcast_7f:ff:fa (01:00:5e:7f:ff:fa)
 Internet Protocol Version 4, Src: 192.168.61.1, Dst: 239.255.255.250
 User Datagram Protocol, Src Port: 52891, Dst Port: 1900
 Simple Service Discovery Protocol

6. Бачимо що Wireshark відстежує наш ping та як працює ARP протокол а саме питає хто це(IP адреса) та наш PC2 відповідає на це своєю MAC адресою(00:50:56:e6:b2:04)

7. Також перевіряємо після цих взаємодій з іншим PC нашу arp таблицю за допомогою команди **arp -a** у терміналі лінуксу

```
onlinegod@ubuntu:~/Desktop$ arp -a
? (192.168.61.25 at 00:50:56:f0:8a:fb [ether] on ens33
_gateway (192.168.61.2) at 00:50:56:e6:b2:04 [ether] on ens33
onlinegod@ubuntu:~/Desktop$
```

Тепер у нас є кеш про PC2.(його можна видалити командою **arp -d [IP адреса]**)

Завдання по варіанту виконано arp-таблиця оновлюється при отриманні arp-reply та на це зроблен Test Case.

Summary: ARP Functionality

Test designed by Rustamov Arsen from IB-93 on 01.10.2022

Description: Verify that **ARP** protocol update arp table

Attachments:

```
onlinegod@ubuntu: ~/Desktop
onlinegod@ubuntu:~/Desktop$ arp
Address            HWtype  HWaddress      Flags Mask    Iface
_gateway           ether    00:50:56:e6:b2:04  C             ens33
onlinegod@ubuntu:~/Desktop$
```

```
Администратор: Windows PowerShell
PS C:\WINDOWS\system32> ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

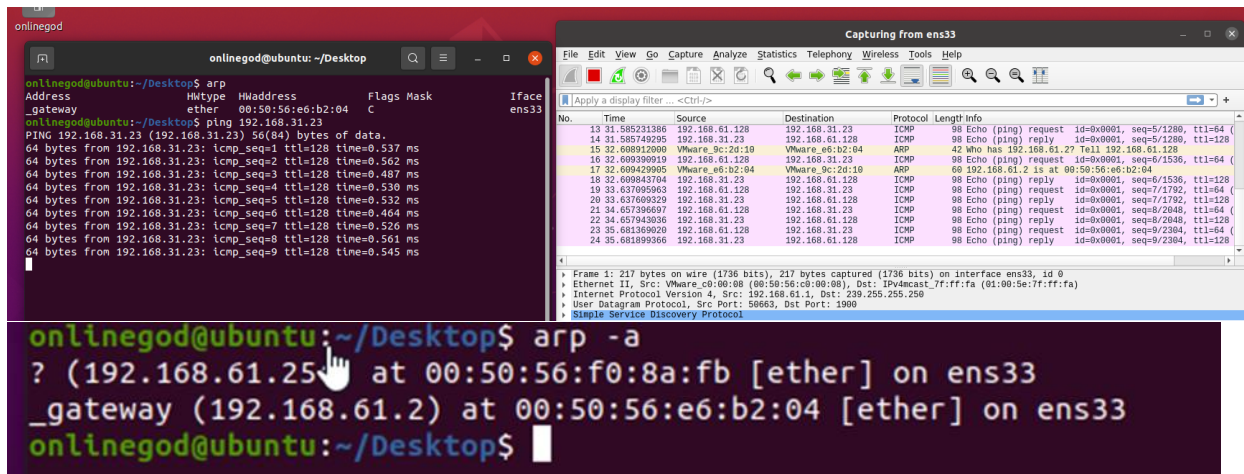
    Connection-specific DNS Suffix  . : 
    Link-local IPv6 Address . . . . . : fe80::c106:c32b:fb29:7a5c%14
    IPv4 Address. . . . . : 192.168.31.23
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.31.1

Ethernet adapter VMware Network Adapter VMnet1:

    Connection-specific DNS Suffix  . : 
    Link-local IPv6 Address . . . . . : fe80::90d9:f4d3:c398:8d22%16
    IPv4 Address. . . . . : 192.168.32.1
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 

Ethernet adapter VMware Network Adapter VMnet8:

    Connection-specific DNS Suffix  . : 
    Link-local IPv6 Address . . . . . : fe80::fc99:c451:90ae:a25b%3
    IPv4 Address. . . . . : 192.168.61.1
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 
PS C:\WINDOWS\system32>
```



Version: **ARP v1**

Setup Description:

PC1-----ethernet1-----PC2

PC1: 192.168.61.128

PC2: 192.168.31.23

Steps (with ER):

1. Check arp table

arp -a <for PC1>

ER: verify that arp table (cache) has no information about PC2

2. Check IP address of PC2

ipconfig <for PC2>

ER: ip of PC2 is shown in PowerShell

3. Run Wireshark for ethernet1

4. Run ping from PC1 to PC2

ER: ping is running

5. Verify that ARP request and ARP reply are present for IP and MAC of PC2 in Wireshark

6. Check arp table one more time

arp -a <for PC1>

ER: verify that arp table now has information about PC2

Висновок: Ми зрозуміли як працює ARP протокол, що таке MAC адреса, навчилися з'єднувати системи через Ethernet ,посилати пакети командою ping

та відстежувати трафік через утіліту Wireshark. Також написали ARP Test Case.