

Міністерство освіти і науки України  
Національний технічний університет України  
«Київський політехнічний інститут»

**Лабораторна робота №3**  
*з дисципліни «Комп'ютерні мережі»*

**«Аналіз просування IP-пакетів в об'єднаній  
мережі з використанням аналізатора трафіку  
Wireshark. Рівень мережевих інтерфейсів.  
Фрагментація IP-дейтаграм»**

Виконала студентка групи: КВ-11

ПІБ: Михайліченко Софія Віталіївна

Перевірив: \_\_\_\_\_

**Київ 2024**

## ***Мета роботи:***

Засвоєння функцій модулів мережевих інтерфейсів, структури заголовку кадру Ethernet, структури мережевого адаптера, процедури фрагментації IP-дейтаграм за допомогою аналізатора мережевого трафіку Wireshark.

## ***План виконання лабораторної роботи:***

1. Ознайомитися та засвоїти теоретичні відомості, викладені в посібнику до лабораторної роботи.
2. За допомогою аналізатора Wireshark виконати захоплення та провести аналіз фрагментованих мережевих пакетів.

## ***Завдання:***

1. В лабораторній роботі проводиться дослідження виконання фрагментації на мережевому рівні стеку TCP/IP. При виконанні роботи використовується програмне забезпечення для аналізу протоколів комп'ютерних мереж Wireshark.
2. Визначіть значення максимального розміру пакету MTU, який може бути переданий канальним рівнем без фрагментації на тому інтерфейсі Вашого комп'ютера, на якому буде відбуватися захоплення пакетів програмою Wireshark.

У Windows для цього можна скористатися командою із командного рядка

**netsh interface ipv4 show subinterfaces,**

а в Unix про значення MTU можна дізнатися за допомогою команди

**ifconfig**

В мережах типу Ethernet значення MTU зазвичай дорівнює 1500 байтів.

1. Запустіть програму Wireshark. Виберіть інтерфейс для захоплення трафіку (меню Capture/Interface) та активізуйте режим захоплення.
2. Перейдіть в командний рядок і виконайте команду ping, вказавши цільову IP-адресу, наприклад, вашого маршрутизатора і параметр -l xxxx, де значення xxxx має перевищувати значення MTU, щоб була виконана фрагментація (наприклад, 5000).
3. Після захоплення трафіку, який виник в результаті виконання команди ping, зупиніть захоплення програмою Wireshark. Проведіть аналіз структури фрагментів, що утворилися. Зверніть увагу на процес фрагментації IP-дейтаграм, що відбувся, та на величину блоку корисного навантаження у фрагментованих пакетах.
4. Результати захоплення фрагментованих пакетів занесіть у звіт.

## ***Короткі теоретичні відомості:***

**ARP (Address Resolution Protocol)** служить для зв'язування IP-адрес з фізичними MAC-адресами в локальних мережах. Коли пристрій надсилає дані, він використовує ARP-запит для отримання MAC-адреси відповідної IP-адреси. Це дозволяє мережевому рівню коректно передавати дані на канальному рівні.

**DNS (Domain Name System)** перетворює доменні імена на IP-адреси, полегшуючи знаходження пристроїв в Інтернеті. Коли користувач вводить доменне ім'я, DNS-сервер виконує запит для отримання відповідної IP-адреси. Це забезпечує коректну маршрутизацію даних між мережевими рівнями.

**Формування DNS-запиту** починається, коли користувач намагається отримати доступ до веб-ресурсу через його доменне ім'я. Пристрій створює DNS-запит, вказуючи, яке доменне ім'я потрібно перетворити в IP-адресу. Цей запит містить дані про тип і відправляється на DNS-сервер. Сервер обробляє запит, перевіряє свої записи, а потім надсилає відповідь з відповідною IP-адресою назад до запитувача.

**Фрагментація IP-дейтаграм** — це процес розділення великих IP-дейтаграм на менші фрагменти для того, щоб їх можна було передати через мережу з обмеженнями розміру кадру (Maximum Transmission Unit, MTU). Коли IP-дейтаграма перевищує значення MTU, вона розбивається на кілька частин, кожна з яких має свій заголовок. Після доставки всіх фрагментів до одержувача, вони знову збираються в оригінальну дейтаграму.

**Рівень мережевого інтерфейсу** забезпечує взаємодію між мережевими протоколами та фізичним середовищем передачі. Він відповідає за інкапсуляцію IP-дейтаграм у кадри для передачі по фізичній мережі. На цьому рівні реалізуються такі протоколи, як Ethernet, Wi-Fi, та інші, які визначають правила передачі даних.

## **Канальний рівень та заголовок кадру:**

Канальний рівень керує доступом до фізичного середовища та обробляє передачу даних у вигляді кадрів. Заголовок кадру містить важливу інформацію для передачі, таку як адреси отримувача та відправника, тип даних, а також контрольні суми для перевірки коректності отриманих даних.

## **Формат заголовку кадру:**

Заголовок кадру Ethernet II складається з кількох основних полів:

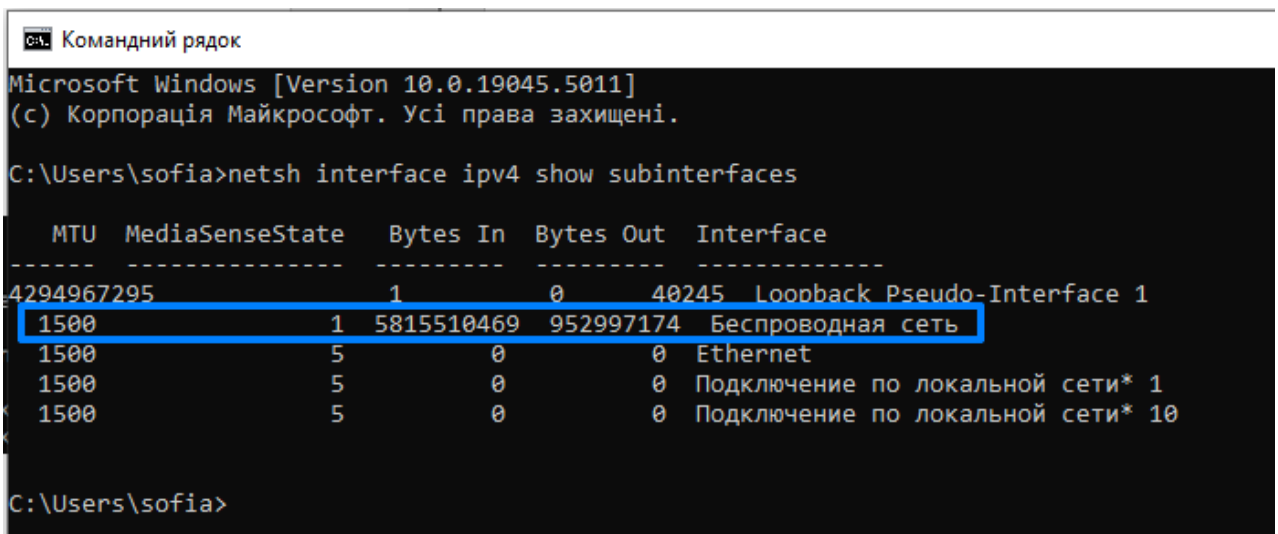
- **Preamble** — для синхронізації передачі;
- **SFD (Start Frame Delimiter)** — роздільник початку кадру;
- **DA (Destination Address)** — MAC-адреса отримувача;
- **SA (Source Address)** — MAC-адреса відправника;
- **Type** — поле визначає який протокол верхнього рівня передає свої дані в кадрі;
- **Data** — містить дані, що передаються протоколом верхнього (мережевого) рівня;
- **CRC (Cyclic Redundancy Check)** — контрольна послідовність кадру містить контрольну суму кадру, яка обчислена за алгоритмом CRC-32.

**Фізичний рівень** відповідає за передачу бітів через фізичне середовище, включаючи електричні, оптичні або радіосигнали. Він визначає характеристики сигналів, типи з'єднань та методи кодування даних, що дозволяє встановлювати з'єднання між пристроями в мережі.

## Порядок виконання роботи:

Спочатку нам потрібно визначити значення максимального розміру пакету MTU, який може бути переданий канальним рівнем без фрагментації на інтерфейсі комп'ютера, на якому буде відбуватися захоплення пакетів програмою Wireshark. Для цього нам потрібна команда із командного рядка, оскільки у нас система Windows, команда матиме наступний вигляд:

***netsh interface ipv4 show subinterfaces***



```
Командный рядок
Microsoft Windows [Version 10.0.19045.5011]
(c) Корпорация Майкрософт. Все права защищены.

C:\Users\sofia>netsh interface ipv4 show subinterfaces

    MTU  MediaSenseState  Bytes In  Bytes Out  Interface
-----
4294967295          1          0      40245  Loopback Pseudo-Interface 1
1500             1 5815510469 952997174 Беспроводная сеть
1500             5          0          0 Ethernet
1500             5          0          0 Подключение по локальной сети* 1
1500             5          0          0 Подключение по локальной сети* 10

C:\Users\sofia>
```

Інтерфейс на якому відбуватиметься захоплення це Беспроводная сеть, як ми бачимо **MTU =1500**, що є нормою для більшості стандартних мереж. Також завдяки цій команді ми змогли побачити:

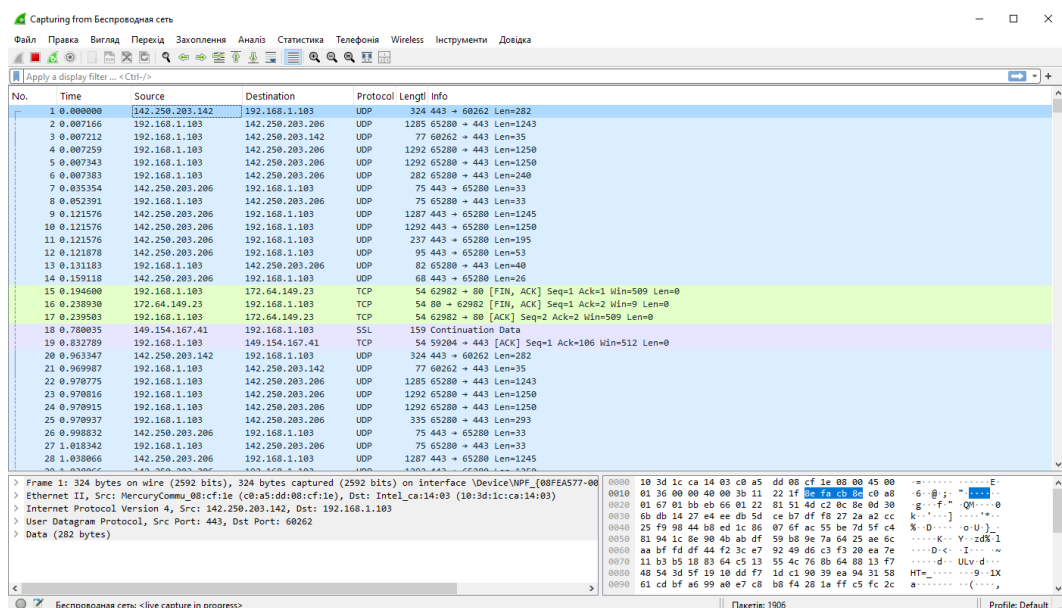
**MediaSenseState** — стан мережевого підключення (значення 1 вказує, що підключення активне);

**Bytes In** — кількість байтів, отриманих через інтерфейс;

**Bytes Out** — кількість байтів, відправлених через інтерфейс;

**Interface** — назва інтерфейсу.

Далі нам потрібно запустити програму Wireshark. Вибрати інтерфейс для захоплення трафіку (меню Capture/Interface) та активізувати режим захоплення:



Далі ми маємо знов перейти у командний рядок та виконати команду **ping**, вказавши цільову IP-адресу, наприклад, маршрутизатора і параметр **-l \*\*\*\***, де значення **\*\*\*\*** має перевищувати значення MTU, щоб була виконана фрагментація (наприклад, 5000). Команда **ping** з параметром **-l** використовується для відправки пакетів певного розміру на вказану IP-адресу. У нашому випадку це IP-адреса маршрутизатора:

```
C:\Users\sofia>ping 192.168.1.1 -l 5000

Pinging 192.168.1.1 with 5000 bytes of data:
Reply from 192.168.1.1: bytes=5000 time=3ms TTL=64
Reply from 192.168.1.1: bytes=5000 time=2ms TTL=64
Reply from 192.168.1.1: bytes=5000 time=2ms TTL=64
Reply from 192.168.1.1: bytes=5000 time=7ms TTL=64

Ping statistics for 192.168.1.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 2ms, Maximum = 7ms, Average = 3ms

C:\Users\sofia>
```

Можемо побачити, що **Packets: Sent = 4, Received = 4, Lost = 0 (0% loss)**: Відправлено 4 пакети, всі 4 пакети були успішно отримані, втрачених пакетів немає.

*Беспроводная сеть						
Файл Правка Видгляд Перехід Захоплення Аналіз Статистика Телефонія Wireless Інструменти Довідка						
Apply a display filter ... <Ctrl-/>						
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	142.250.203.142	192.168.1.103	UDP	326	443 → 60262 Len=284
2	0.005539	192.168.1.103	142.250.203.206	UDP	1285	65280 → 443 Len=1243
3	0.005680	192.168.1.103	142.250.203.206	UDP	1292	65280 → 443 Len=1250
4	0.005699	192.168.1.103	142.250.203.206	UDP	1292	65280 → 443 Len=1250
5	0.005732	192.168.1.103	142.250.203.206	UDP	344	65280 → 443 Len=302
6	0.018589	192.168.1.103	142.250.203.142	UDP	75	60262 → 443 Len=33
7	0.034855	142.250.203.206	192.168.1.103	UDP	77	443 → 65280 Len=35
8	0.049908	192.168.1.103	142.250.203.206	UDP	77	65280 → 443 Len=35
9	0.104283	142.250.203.206	192.168.1.103	UDP	1287	443 → 65280 Len=1245
10	0.104283	142.250.203.206	192.168.1.103	UDP	1292	443 → 65280 Len=1250
11	0.104283	142.250.203.206	192.168.1.103	UDP	633	443 → 65280 Len=591
12	0.104479	142.250.203.206	192.168.1.103	UDP	286	443 → 65280 Len=244
13	0.113755	192.168.1.103	142.250.203.206	UDP	84	65280 → 443 Len=42
14	0.142207	142.250.203.206	192.168.1.103	UDP	70	443 → 65280 Len=28
15	0.348384	MercuryComm_08:cf:...	Broadcast	ARP	42	who has 192.168.1.100? Tell 192.168.1.1
16	0.360680	142.250.203.142	192.168.1.103	UDP	326	443 → 60262 Len=284
17	0.367611	192.168.1.103	142.250.203.142	UDP	75	60262 → 443 Len=33
18	0.371926	192.168.1.103	142.250.203.206	UDP	1285	65280 → 443 Len=1243
19	0.372075	192.168.1.103	142.250.203.206	UDP	1292	65280 → 443 Len=1250
20	0.372097	192.168.1.103	142.250.203.206	UDP	1292	65280 → 443 Len=1250
21	0.372129	192.168.1.103	142.250.203.206	UDP	293	65280 → 443 Len=251

> Frame 4: 1292 bytes on wire (10336 bits), 1292 bytes captured (10336 bits) on interface \Device\NPF\_{08FEA57...}

> Ethernet II, Src: Intel\_ca:14:03 (10:3d:1c:ca:14:03), Dst: MercuryComm\_08:cf:1e (c0:a5:dd:08:cf:1e)

> Internet Protocol Version 4, Src: 192.168.1.103, Dst: 142.250.203.206

> User Datagram Protocol, Src Port: 65280, Dst Port: 443

> Data (1250 bytes)

0000 c0 a5 dd 08 cf 1e 1

0010 04 fe de 08 40 00 4

0020 cb ce ff 00 01 bb 0

0030 70 73 95 4f 4c f7 1

0040 02 25 f6 7a cc 25 3

0050 51 d8 89 29 cb 4c c

0060 cd f6 6b 67 c7 21 3

Це повідомлення є частиною протоколу **ARP (Address Resolution Protocol)**, який використовується для визначення MAC-адреси пристрою в локальній мережі на основі його IP-адреси.

*Беспроводная сеть						
Файл Правка Видгляд Перехід Захоплення Аналіз Статистика Телефонія Wireless Інструменти Довідка						
icmp						
No.	Time	Source	Destination	Protocol	Length	Info
→ 373	4.423693	192.168.1.103	192.168.1.1	ICMP	602	Echo (ping) request id=0x0001, seq=46/11776, ttl=64 (reply in 377)
← 377	4.425984	192.168.1.1	192.168.1.103	ICMP	602	Echo (ping) reply id=0x0001, seq=46/11776, ttl=64 (request in 373)
441	5.436677	192.168.1.103	192.168.1.1	ICMP	602	Echo (ping) request id=0x0001, seq=47/12032, ttl=64 (reply in 445)
445	5.441583	192.168.1.1	192.168.1.103	ICMP	602	Echo (ping) reply id=0x0001, seq=47/12032, ttl=64 (request in 441)
508	6.452900	192.168.1.103	192.168.1.1	ICMP	602	Echo (ping) request id=0x0001, seq=48/12288, ttl=64 (reply in 512)
512	6.459602	192.168.1.1	192.168.1.103	ICMP	602	Echo (ping) reply id=0x0001, seq=48/12288, ttl=64 (request in 508)
565	7.466900	192.168.1.103	192.168.1.1	ICMP	602	Echo (ping) request id=0x0001, seq=49/12544, ttl=64 (reply in 569)
569	7.474507	192.168.1.1	192.168.1.103	ICMP	602	Echo (ping) reply id=0x0001, seq=49/12544, ttl=64 (request in 565)

> Frame 373: 602 bytes on wire (4816 bits), 602 bytes captured (4816 bits) on interface \Device\NPF\_{08FEA57...}

> Ethernet II, Src: Intel\_ca:14:03 (10:3d:1c:ca:14:03), Dst: MercuryComm\_08:cf:1e (c0:a5:dd:08:cf:1e)

> Internet Protocol Version 4, Src: 192.168.1.103, Dst: 192.168.1.1

> Internet Control Message Protocol

0000 c0 a5 dd 08 cf 1e 10 3d 1c ca 14 03 08 00 45 00 .....^

0010 02 4c 88 9f 02 2b 40 01 00 00 c0 a8 01 67 c0 a8 .L....g

0020 01 01 71 72 73 74 75 76 77 61 62 63 64 65 66 67 ..qrstu

0030 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 77 hijklmr

0040 61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f 60 abcdefg

0050 71 72 73 74 75 76 77 61 62 63 64 65 66 67 68 69 qrstuvw

0060 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 77 61 62 jklmnop

0070 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 cdefghi

0080 73 74 75 76 77 61 62 63 64 65 66 67 68 69 6a 6b stuvwat

0090 6c 6d 6e 6f 70 71 72 73 74 75 76 77 61 62 63 64 tuvwxyz

**ICMP (Internet Control Message Protocol)** — міжмережевий протокол керуючих повідомлень, що входить в стек протоколів TCP/IP. В основному ICMP використовується для передачі повідомлень про помилки й інші виняткові ситуації, що виникли при передачі даних.



```

Destination Address: 192.168.1.1
▼ [4 IPv4 Fragments (5008 bytes): #370(1480), #371(1480), #372(1480), #373(568)]
  [Frame: 370, payload: 0-1479 (1480 bytes)]
  [Frame: 371, payload: 1480-2959 (1480 bytes)]
  [Frame: 372, payload: 2960-4439 (1480 bytes)]
  [Frame: 373, payload: 4440-5007 (568 bytes)]
  [Fragment count: 4]
  [Reassembled IPv4 length: 5008]
  [Reassembled IPv4 data [...]: 080041210001002e6162636465666768696a6b6c6d6e6f7071727374757677616:
  [Stream index: 10]

```

Виконаємо аналіз фрагментованих IP-пакетів, які отримані в результаті виконання команди ping. Ми отримали 4 фрагменти оригінального пакета, загальна довжина якого становить 5008 байт. Перші три фрагменти мають розмір 1480 байт, тоді як останній фрагмент є меншим — 568 байт.

The image shows a Wireshark capture of network traffic. The packet list pane displays several packets, with packets 370, 371, 372, and 373 highlighted in blue. These packets are identified as '1514 Fragmented IP protocol (proto=ICMP 1, off=0, ID=889f) [Reassembled in #373]'. Packet 373 is also identified as a '602 Echo (ping) request id=0x0001, seq=46/11776, ttl=64 (reply in 377)'. The packet details pane for packet 373 shows the 'Internet Protocol Version 4' section, where the 'Destination Address' is 192.168.1.1. The 'Reassembled IPv4 in frame: 373' section is highlighted in blue. The packet bytes pane shows the raw data of the packet, with the first 1480 bytes highlighted in blue.

Повідомлення "[Reassembled IPv4 in frame: 373]" в Wireshark означає, що дані з фрагментованих IPv4-пакетів були об'єднані в один цілісний пакет, і цей пакет відображається в кадрі з номером 373.

У протоколах IPv4 і ICMP (Internet Control Message Protocol) зазвичай можна спостерігати однакові значення ідентифікації (ID) для пакетів, які належать до однієї сесії пінгу.

**IP ID** використовується для ідентифікації пакетів, які фрагментуються. Коли великий пакет розділяється на менші фрагменти, всі ці фрагменти отримують однакове значення ID. Це дозволяє отримувачу правильно зібрати фрагменти в оригінальний пакет.

**ICMP ID** використовується в запитах Echo (ping) для ідентифікації запитів та відповідей. Значення ID ICMP відповідає тому ж значенню, яке вище наведено для IP-пакетів. Це дозволяє перевірити, що відповідь ICMP (Echo reply) належить до конкретного запиту (Echo request).

### ***Висновок:***

В ході виконання лабораторної роботи було проведено аналіз просування IP-пакетів у об'єднаній мережі з використанням аналізатора трафіку Wireshark. Завдяки отриманим даним було засвоєно ключові аспекти роботи мережевих інтерфейсів, структури заголовків кадрів Ethernet, а також процесу фрагментації IP-дейтаграм. При виконанні команди ping з параметром, що перевищує максимальний розмір пакету MTU, спостерігалось утворення чотирьох фрагментів оригінального пакета, що підтверджує правильність процесу фрагментації. Аналіз отриманих фрагментів дозволив зрозуміти, як IP ID та ICMP ID допомагають у ідентифікації пакетів та відстеженні зв'язку між запитами та відповідями. В результаті було визначено, що максимальний розмір пакету MTU для нашого мережевого інтерфейсу становить 1500 байтів, що є стандартним значенням для мереж Ethernet. Отримані дані свідчать про ефективність роботи протоколів ARP та ICMP в локальних мережах, а також про їх важливість у забезпеченні коректної передачі даних. Ця лабораторна робота дозволила краще зрозуміти основи роботи з мережевими протоколами та їхню роль у забезпеченні стабільного та ефективного обміну даними в комп'ютерних мережах.