

Національний технічний університет України

“Київський політехнічний інститут імені Ігоря Сікорського”

Факультет прикладної математики

Кафедра системного програмування та спеціалізованих комп’ютерних систем

# Лабораторна робота №14

***з дисципліни «Комп’ютерні мережі»***

**«Аналіз мережевої взаємодії при використанні послідовних інтерфейсів на маршрутизаторах. Протокол HDLC»**

Виконав

студент 4-го курсу

групи КВ-11

Терентьєв Іван Дмитрович

Перевірив: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Київ 2024**

***Мета роботи***

Засвоєння принципів взаємодії мережевих пристроїв при використанні послідовних інтерфейсів на маршрутизаторах за допомогою програми симуляції комп’ютерних мереж Cisco Packet Tracer.

***План виконання лабораторної роботи***

1. Засвоєння теоретичних відомостей;
2. Побудова топології досліджуваної мережі;
3. Налаштування мережевого обладнання;
4. Аналіз процесів фрагментації на маршрутизаторах при пересиланні пакетів різних протоколів із однієї мережі в іншу при використанні послідовних інтерфейсів.

***Завдання***

1. Побудувати модель комп’ютерної мережі, яка зображена на рисунку 14.5.
2. Покроково виконати необхідні мережеві налаштування мережевих пристроїв.
3. Провести аналіз процесу фрагментації на маршрутизаторах при виконанні команди ping на комп’ютері PC0.
4. Провести аналіз процесу фрагментації на маршрутизаторах при виконанні HTTP-запиту до Web-сервера на Server0.
5. Провести аналіз процесу фрагментації на маршрутизаторах при виконанні команди ping на маршрутизаторі.

***Теоретичні відомості***

Підключення комп’ютерів до мережі Інтернет може відбуватися через локальні (LAN) або глобальні (WAN) мережі. Глобальні мережі можуть використовувати різні технології канального рівня, такі як **HDLC (High-Level Data Link Control), PPP, Frame Relay, ATM, ISDN**.

Протокол **HDLC** - це стандартний протокол другого рівня моделі OSI, який забезпечує надійну передачу даних через послідовні інтерфейси. Він підтримує режими **точка-точка (Point-to-Point) та точка-багатоточка (Point-to-Multipoint)** і виконує функції **кадрування, виявлення та виправлення помилок, керування потоком**.

HDLC-кадр включає такі поля:

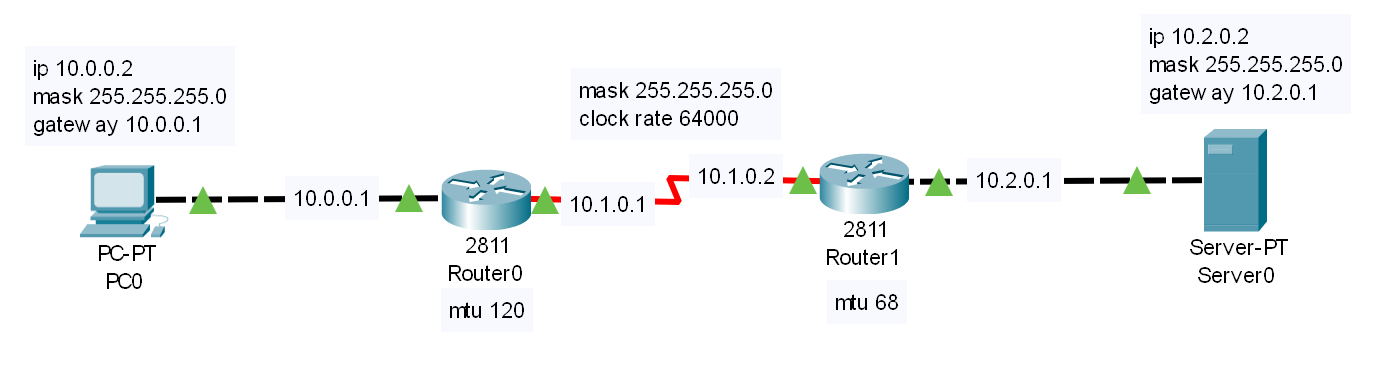
* **FLAG** – позначає початок і кінець кадру.
* **ADDRESS** – містить адресу отримувача.
* **CONTROL** – службова інформація.
* **PROTOCOL** – номер протоколу третього рівня.
* **DATA** – корисні дані.
* **FCS** – контрольна сума для перевірки цілісності.

На маршрутизаторах Cisco використовується варіант **Cisco HDLC**, який додає поле **PROTOCOL**, що дозволяє підтримувати мультиплексування різних мережевих протоколів.

У мережах, що використовують послідовні інтерфейси, може виникати **фрагментація IP-дейтаграм**, коли довжина пакета перевищує **MTU (Maximum Transmission Unit)** інтерфейсу. Це призводить до поділу пакета на менші частини, які потім збираються на пристрої-отримувачі.

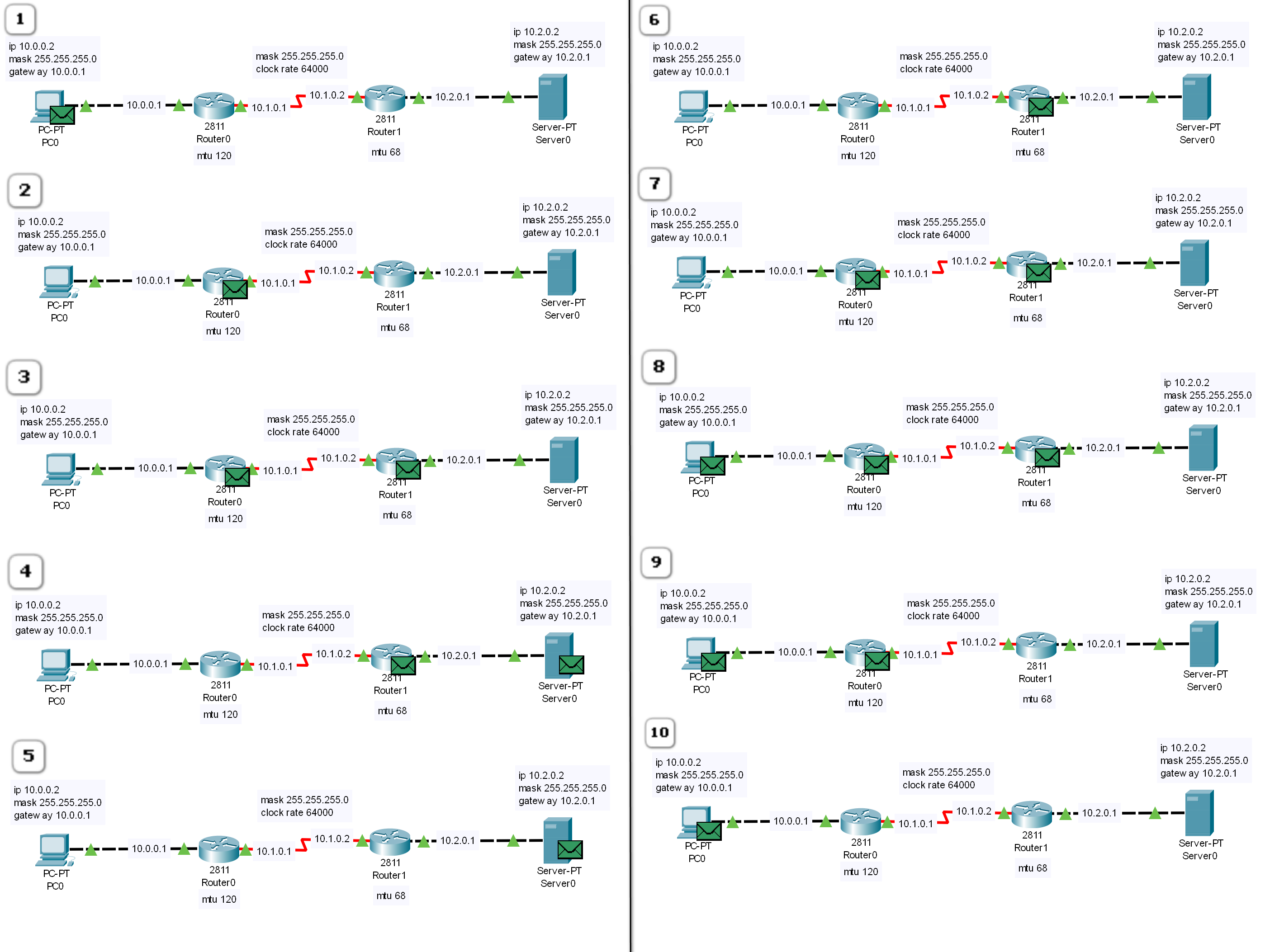
***Хід роботи***

Побудуємо модель комп’ютерної мережі, результат можна побачити на рисунку 1.



*Рис. 1 – Модель комп’ютерної мережі*

Далі покроково виконаємо необхідні мережеві налаштування мережевих пристроїв, перевіряючи виконані налаштування й перейдемо до виконання завдання.



*Рис. 2 – Аналіз фрагментації при виконанні команди ping на PC0*

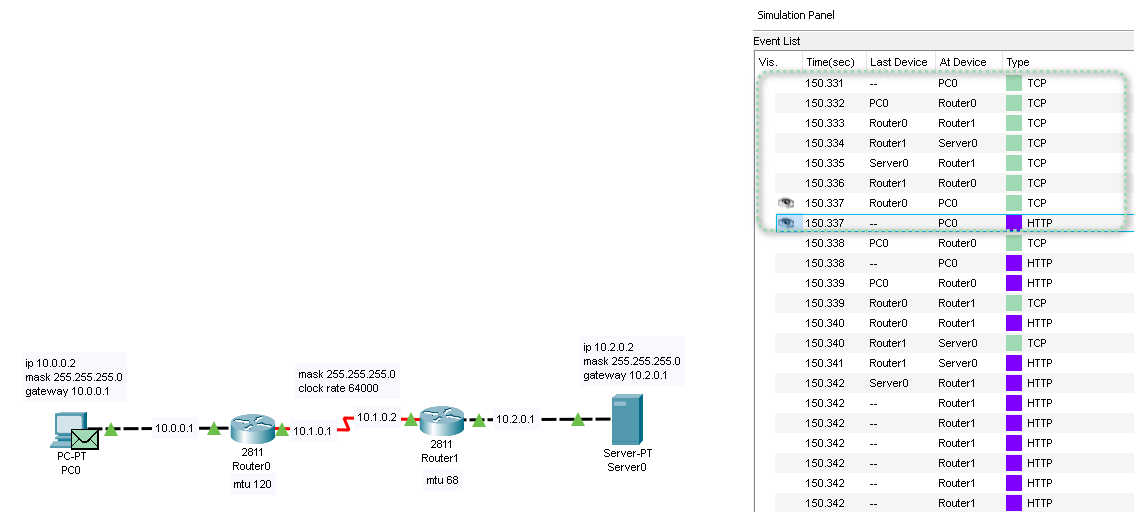
Проведемо аналіз передивившись явище зображене на рис. 2.

Фрагментація IP-дейтаграм відбувається через те, що максимальний розмір переданого пакета (MTU – Maximum Transmission Unit) на різних інтерфейсах маршрутизаторів може відрізнятися. У цьому завданні цей процес спостерігається під час проходження ICMP-запиту (ping) від ПК до сервера через два маршрутизатори.

**Основні етапи фрагментації:**

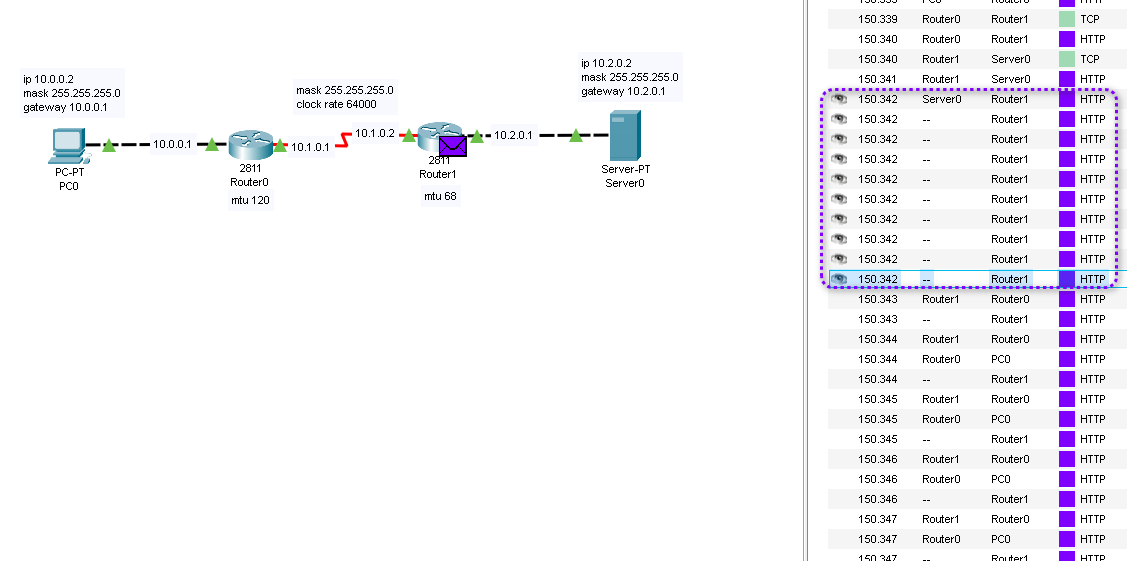
1. **Відправка ICMP-запиту:**
   * Комп’ютер **PC0** відправляє ping-пакет (ICMP Echo Request) на сервер **Server0**.
   * Початковий розмір IP-дейтаграми – **128 байтів**.
2. **Фрагментація на Router0:**
   * Інтерфейс **fa0/0** маршрутизатора **Router0** отримує дейтаграму в **128 байтів**.
   * Оскільки MTU послідовного інтерфейсу **se1/0** = **120 байтів**, пакет розбивається на два фрагменти:
     + **Перший фрагмент** – **120 байтів**
     + **Другий фрагмент** – **28 байтів**
   * Кожен з фрагментів інкапсулюється у **HDLC-кадри** та передається на **Router1**.
3. **Перетворення HDLC → Ethernet на Router1:**
   * Router1 отримує **HDLC-кадри**, що містять фрагменти IP-дейтаграми.
   * Відбувається їх **дефрагментація** – збірка назад у початкову IP-дейтаграму.
   * Далі **Ethernet-кадр** передається на **Server0**.
4. **Фрагментація у зворотному напрямку (відповідь від сервера):**
   * Server0 створює відповідь **ICMP Echo Reply** розміром **128 байтів**.
   * Ця дейтаграма передається на **Router1**, але MTU його інтерфейсу **se1/0** = **68 байтів**.
   * Відбувається **додаткова фрагментація** на **три фрагменти**:
     + **Перший** – **68 байтів**
     + **Другий** – **68 байтів**
     + **Третій** – **32 байти**
   * Фрагменти інкапсулюються в **HDLC-кадри** та передаються на Router0.
5. **Перетворення HDLC → Ethernet на Router0:**
   * Router0 знову отримує **фрагментовані пакети**, але тепер вони передаються у **Ethernet-мережу**.
   * Він конвертує **HDLC-кадри** назад у **Ethernet-кадри** та надсилає фрагменти на **PC0**, де вони **збираються** у вихідний ICMP-пакет.

Перейдемо до другого завдання та розглянемо фрагментацію при виконанні HTTP-запиту до Web-серверу Server0.



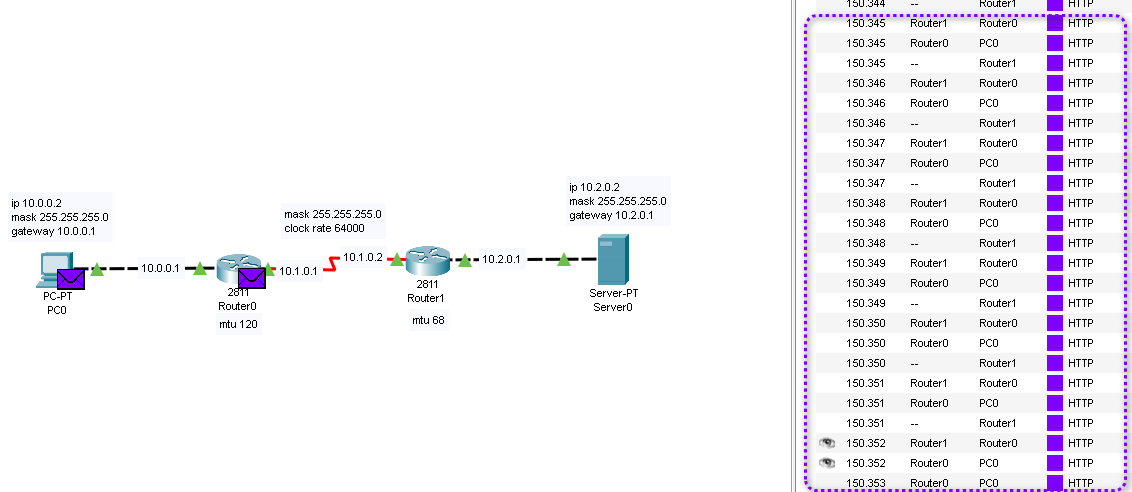
*Рис. 3 – Встановлення TCP з’єднання*

В першу чергу PC0 та Server0 встановлюють TCP з’єднання під час якого не відбувається фрагментація. Після цього відбувається HTTP з’єднання зображене на рис. 4.

**

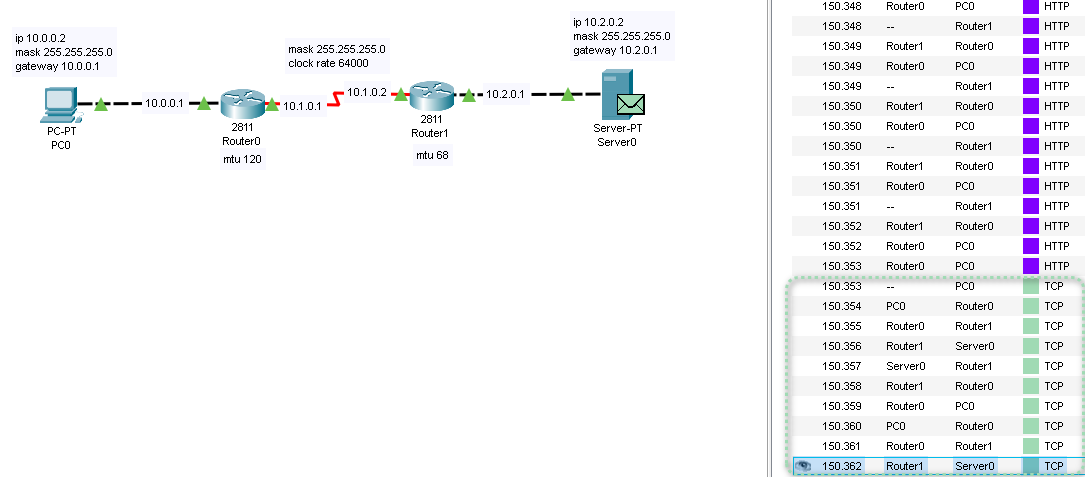
*Рис. 4 – Встановлення HTTP з’єднання*

Як можна побачити так як кадр з інкапсульованим HTTP пакетом розміром 491 байт, значно перевищує MTU інтерфейсу se1/0 маршрутизатора, то маршрутизатор виконує фрагментацію й дейтаграма розбивається на 10 фрагментів довжина яких не перевищує 68 байтів.

**

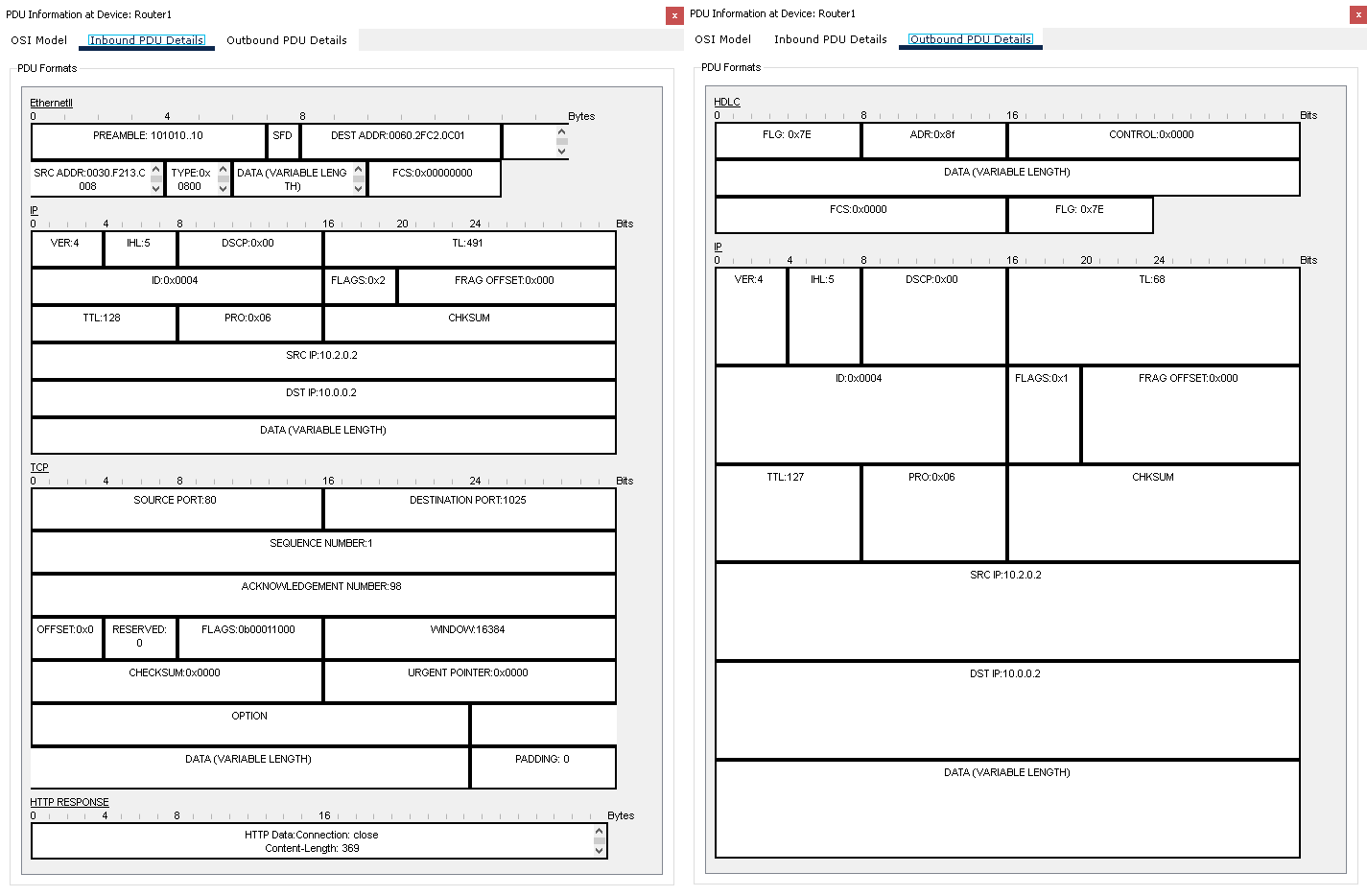
*Рис. 5 – Збирання пакетів на PC0*

На рис. 5 можна побачити, що фрагменти були інкапсульовані в HDLC кадри і надіслані маршрутизатору Router0. Маршрутизатор замінив заголовки HDLC на заголовки Ethernet та фрагменти надходять на PC0, де вже відбувається збирання первинної дейтаграми. Та в кінці відбувається розірвання TCP з’єднання, що можна побачити на рис. 6.

**

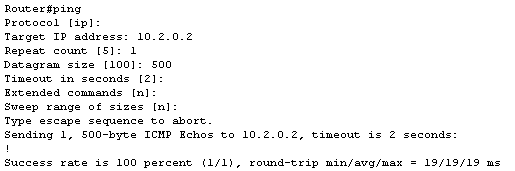
*Рис. 6 – Розірвання з’єднання TCP*

Більш детально перетворення початкової дейтаграми HTTP в HDLC-кадри при передачі від Server0 до Router1 можна побачити на рис.7.

**

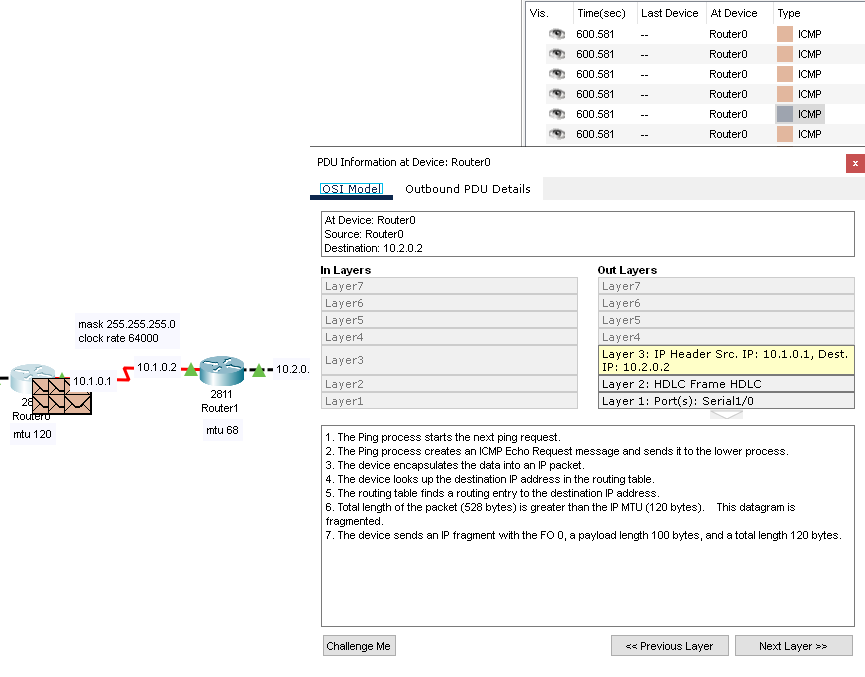
*Рис. 7 – Перетворення початкової дейтаграми в HDLC-кадри*

Перейдемо до розгляду процесу фрагментації при виконанні розширеної команди ping на маршрутизаторі Router0 зображеної на рис. 8.

**

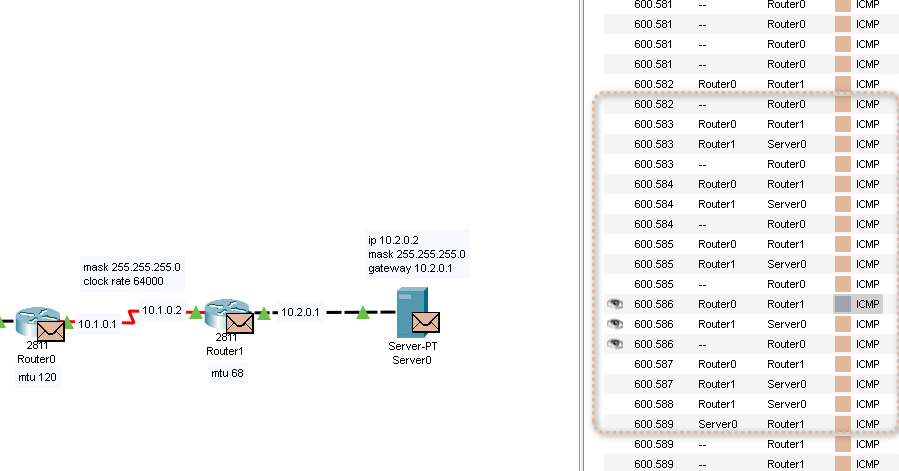
*Рис. 8 – Розширена команда ping*

На рис. 9 можна побачити, що при відправці з Router0 одразу відбулася інкапсуляція в HDLC-кадри, а саме пакет становив 528 байт, що було вище ніж MTU 120 байт. Тому було створено 5 фрагментів по 120 байтів та 1 фрагмент на 28 байтів.

**

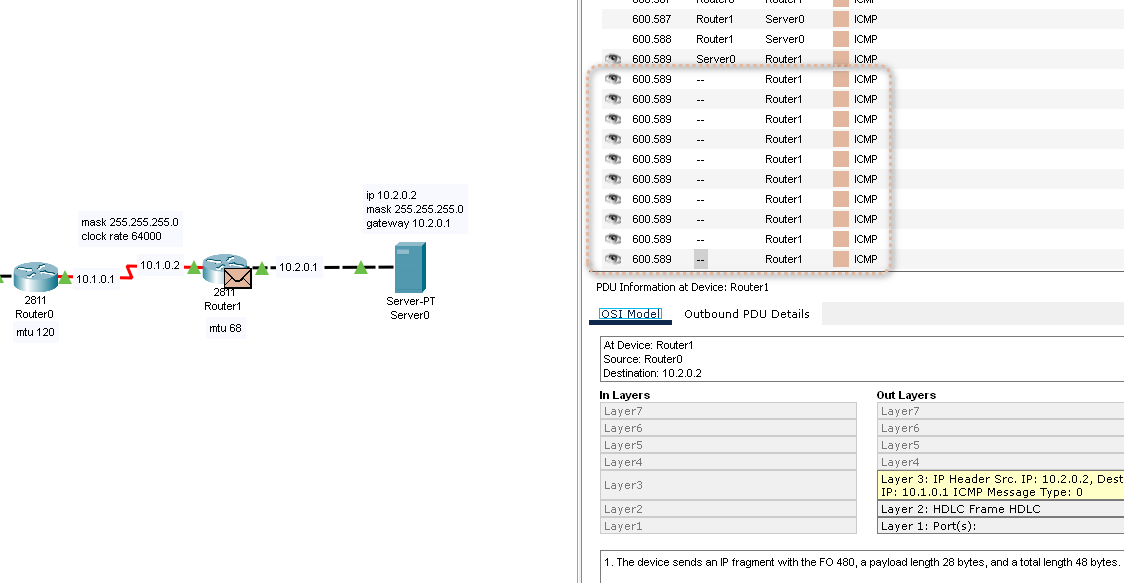
*Рис. 9 – Інкапсуляція в HDLC-кадри при початку виконанні відправки ICMP-пакету*

Після отримання пакетів маршрутизатором Router1, він починає перетворювати HDLC-кадри на Ethernet, а на Server0 відбувається кінцева збірка.



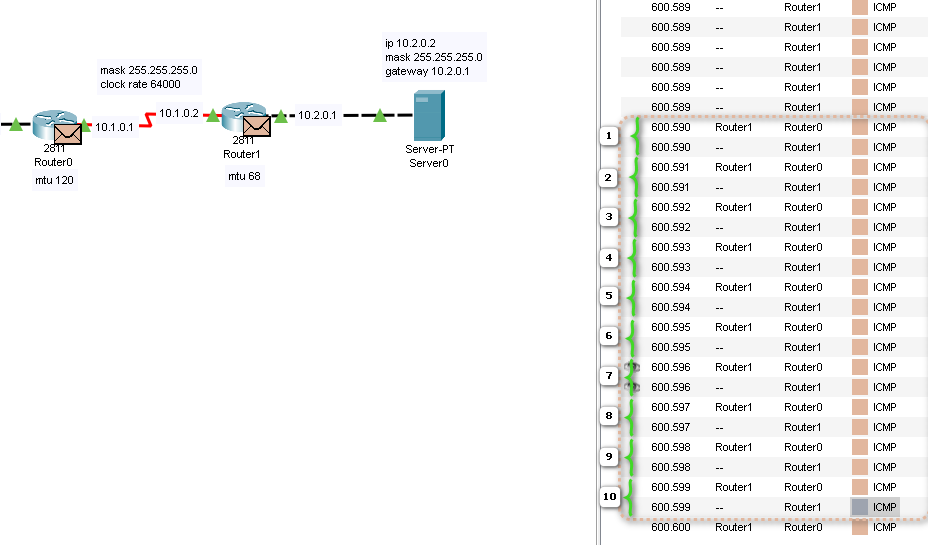
*Рис. 10 – Збірка пакетів на Server0*

Після отримання початкової дейтаграми на Server0, сервер формує відповідь яку надсилає спочатку на маршрутизатор Router1, що має MTU 68 байтів, через що відбувається фрагментація на 9 пакетів по 68 байтів та 1 пакет на 48 байтів, зображене на рис. 11.



*Рис. 11 – Відповідь від сервера Server0*

Й в кінці маршрутизатор Router1 передає HDLC-кадри на Router0 відповідну кількість разів до кількості фрагментів, що й зображене на рис. 12.



*Рис. 12 – Збірка пакетів на Router0*

***Висновок***

У ході виконання лабораторної роботи було досліджено процес фрагментації IP-дейтаграм при передачі пакетів через маршрутизатори, використовуючи послідовні інтерфейси. Було проаналізовано, як зміна MTU на різних інтерфейсах впливає на розбиття IP-пакетів та їх інкапсуляцію в протоколи HDLC і Ethernet.

Основні висновки:

1. **Фрагментація виникає через різницю в MTU** – коли розмір пакета перевищує допустимий для певного інтерфейсу, він ділиться на менші фрагменти, які передаються окремо.
2. **Маршрутизатори виконують інкапсуляцію та деінкапсуляцію** – IP-фрагменти інкапсулюються в HDLC-кадри на послідовних інтерфейсах, а потім перетворюються на Ethernet-кадри при передачі в інші сегменти мережі.
3. **Зворотна фрагментація при отриманні відповіді** – при отриманні відповіді від сервера пакети можуть знову піддаватися фрагментації, якщо їх розмір перевищує MTU вихідного інтерфейсу маршрутизатора.
4. **Фрагментація може впливати на продуктивність** – обробка фрагментованих пакетів збільшує навантаження на маршрутизатори та створює додаткові затримки в мережі.

В результаті виконаної роботи було закріплено навички налаштування та аналізу роботи послідовних інтерфейсів маршрутизаторів, а також досліджено принципи фрагментації IP-дейтаграм при передачі трафіку в мережі.