Форма № Н-9.02

Національний університет «Полтавська політехніка імені Юрія Кондратюка»
(повне найменування вищого навчального закладу)

Навчально-науковий інститут інформаційних технологій і робототехніки
(повне найменування інституту, назва факультету (відділення))

Кафедра автоматики, електроніки та телекомунікацій
(повна назва кафедри (предметної, циклової комісії))

Пояснювальна записка

до кваліфікаційної роботи

бакалавр
(освітньо-кваліфікаційний рівень)

на тему **Дослідження протоколів стеку TCP\IP**

Виконав: студент 4 курсу, групи 401-ТТ
спеціальності 172 «Телекомунікації та
(шифр і назва напряму підготовки, спеціальності)
радіотехніка

Сидорчук Б.М.
(прізвище та ініціали)
Керівник Жученко О.С.
(прізвище та ініціали)
Рецензент Єрмілова Н.В.
(прізвище та ініціали)

Полтава - 2024 рік

Національний університет «Полтавська політехніка імені Юрія Кондратюка»
Інститут Навчально-науковий інститут інформаційних технологій та робототехніки
Кафедра Автоматики, електроніки та телекомунікацій
Ступінь вищої освіти Бакалавр
Спеціальність 172 «Телекомунікації та радіотехніка»

ЗАТВЕРДЖУЮ

Завідувач кафедри автоматики, електроніки та телекомунікацій

_____О.В. Шефер
«01» квітня 2024 р.

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРУ СТУДЕНТУ

Сидорчуку Богдану Миколайовичу

1. Тема роботи «Дослідження протоколів стеку TCP\IP»
керівник роботи Жученко Олександр Сергійович, к.т.н., доцент
затверджена наказом вищого навчального закладу від 2.12.2024 року № 1481/1-ос

2. Строк подання студентом проекту (роботи) 10.06.2024 р.

3. Вихідні дані до проекту (роботи) Усі пакети даних, які проходять через мережу, включає дані : FTP-запитів, SMTP-запитів, UDP-пакетів, TCP-запитів і т.д..

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Дослідження загальних принципів аналізу протоколів с за допомогою програмного забезпечення. Дослідження протоколів стеку TCP/IP. Розрахунок корисно пропускної здатності тракту мереж с комутацією пакетів.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових плакатів):
   1) Титульний слайд;
   2) Мета роботи;
   3) Огляд стеку протоколів TCP/IP;
   4) Протокол-аналізатор Wireshark;
   5) Дослідження протоколу FTP;
   6) Дослідження протоколу SMTP;
   7) Дослідження протоколу TCP;
   8) Дослідження протоколу UDP;
   9) Дослідження протоколу IP;
   10) Дослідження протоколу Etnernet;
   11) Висновки.

6. Дата видачі завдання 01.04.2024 р.

КАЛЕНДАРНИЙ ПЛАН

Пор. №	Назва етапів кваліфікаційної роботи бакалавра	Термін виконання етапів роботи			Примітка (плакати)
1	Огляд стеку протоколів TCP/IP	25.04.24	I	20%	Пл. 1
2	Дослідження загальних принципів аналізу протоколів с за допомогою програмного додатку wireshark	10.05.24		40%	Пл. 2
3	Дослідження протоколів стеку TCP/IP	23.05.24	II	60%	Пл. 4
4	Розрахунок корисно пропускної здатності тракту мереж с комутацією пакетів	01.06.24		80 %	Пл. 5
5	Оформлення кваліфікаційної роботи бакалавра	10.06.24	III	100%	Пл. 6

Студент _____ Сидорчук Б.М
(підпис) (прізвище та ініціали)

Керівник роботи _____ Жученко О.С.
(підпис) (прізвище та ініціали)

РЕФЕРАТ

Кваліфікаційна робота бакалавра: 46 сторінок, 21 рисунок, 2 додатка, 10 джерел.

Об’єкт дослідження: протоколи стеку TCP/IP.

Мета роботи: дослідження та аналіз принципу роботи протоколів стеку TCP/IP.

Методи: Аналіз існуючої наукової та технічної літератури, проведення тестувань з використанням спеціалізованого програмного забезпечення.

Ключові слова: протокол, мережа, трафік.

ABSTRACTB

Bachelor's qualification work**:** 46 pages, 21 pictures, 10 sources, 2 addition.

Object of research: protocols of the TCP/IP stack.

Purpose: research and analysis of the principle of operation of TCP/IP stack protocols.

Methods: Analysis of existing scientific and technical literature, testing using specialized software.

Keywords: protocol, network, traffic.

ЗМІСТ

ВСТУП

В сучасному світі інформаційних технологій, де мережеві комунікації стали невід'ємною частиною повсякденного життя, стек протоколів TCP/IP відіграє ключову роль у забезпеченні надійної та ефективної передачі даних. З моменту свого створення, TCP/IP зазнав значних змін та удосконалень, що дозволило йому стати основним набором протоколів для Інтернету та багатьох інших мереж. Однак, швидкий розвиток технологій, зростання обсягів передаваних даних, а також постійно зростаючі вимоги до безпеки та швидкості передачі інформації вимагають подальших досліджень та вдосконалень цього протоколу.

Актуальність даної роботи обумовлена необхідністю глибокого аналізу сучасного стану стеку протоколів TCP/IP, виявлення його слабких місць та потенційних загроз. Сьогодні все більше уваги приділяється кібербезпеці, що вимагає впровадження нових методів захисту даних на всіх рівнях мережевих комунікацій. Крім того, розвиток нових технологій вимагає від TCP/IP адаптації до умов великої кількості одночасних підключень та обробки великих обсягів даних.

У роботі описується загальний огляд сімейства протоколів TCP/IP, основні принципи їх роботи.

РОЗДІЛ 1
Огляд структури стеку протоколів TCP/IP

1.1 Модель TCP/IP

Модель TCP/IP, що розшифровується як Transmission Control Protocol/Internet Protocol, являє собою концептуальну модель, яка детально описує, як саме комп'ютери і різні пристрої взаємодіють та обмінюються даними у великій глобальній мережі Інтернет. Ця модель складається з чотирьох основних рівнів (таблиця 1.1): прикладного рівня (application), транспортного рівня (transport), міжмережевого рівня (internet) та канального рівня (Network Access Layer). Кожен з цих рівнів виконує свої унікальні функції та операції, які забезпечують ефективну передачу даних через мережу, дозволяючи комп'ютерам спілкуватися один з одним.

Таблиця 1.1 – Модель TCP/IP

Модель OSI		Модель TCP/IP	
Прикладний	7	4	Прикладний
Представницький	6		
Сеансовий	5		
Транспортний	4	3	Транспортний
Мережевий	3	2	Мережевий
Канальний	2	1	Канальний
Фізичний	1		

1.2 Прикладний рівень

На прикладному рівні стеку TCP/IP здійснюється комунікація між різноманітними додатками, які працюють на різних пристроях, що знаходяться у мережі. Цей рівень включає в себе велику кількість різноманітних протоколів, які дозволяють програмам ефективно обмінюватися даними. Основні протоколи

прикладного рівня включають такі протоколи, як HTTP, HTTPS, FTP, SMTP, POP3, IMAP, DNS та інші.

Основна мета прикладного рівня полягає у забезпеченні інтерфейсу для взаємодії між різними додатками та сервісами, що працюють у мережі. Наприклад, HTTP використовується для передачі веб-сторінок, FTP - для передачі файлів, SMTP - для відправлення електронної пошти, а DNS - для перетворення доменних імен на IP-адреси.

Програми, що працюють на прикладному рівні, користуються послугами транспортного рівня, таким як TCP або UDP, для передачі даних через мережу. Вони використовуються для побудови з'єднань між різними пристроями, для керування потоком даних, а також для перевірки коректної доставки даних.

1.3 Транспортний рівень

Транспортний рівень стеку TCP/IP відповідає за передачу даних між двома кінцевими пунктами, які знаходяться у мережі. Основні протоколи, що активно використовуються на цьому рівні, - це TCP (Transmission Control Protocol) і UDP (User Datagram Protocol). Протокол TCP забезпечує надійну передачу даних, що робить його ідеальним вибором для тих застосувань, де важлива точність і цілісність даних, таких як передача файлів чи веб-сторінок. З іншого боку, протокол UDP часто використовується для швидкої передачі даних у реальному часі, таких як відео або голосові дзвінки, де певні втрати даних є прийнятними, але важлива мінімальна затримка для забезпечення плавного та безперебійного потоку інформації.

1.4 Мережевий рівень

Головна функція мережевого рівня полягає у здійсненні процесу маршрутизації. Він відповідає за визначення найкоротшого і найефективнішого шляху для передачі даних від відправника до кінцевого отримувача. Це включає в себе визначення оптимального маршруту для передачі, розбиття даних на

окремі пакети, які можуть бути передані через різні мережні вузли, і пересилання цих пакетів через мережу до кінцевого пункту призначення.

Додатково, мережевий рівень відповідає за виконання функцій ідентифікації і адресації вузлів в мережі. Кожен вузол мережі має свою унікальну IP-адресу, яка використовується для ідентифікації його в мережі. Ці адреси необхідні для маршрутизації пакетів даних від відправника до кінцевого отримувача.

Популярні протоколи мережевого рівня включають в себе Internet Protocol (IP), Internet Control Message Protocol (ICMP), який використовується для обміну повідомленнями про помилки та стан мережі, і Address Resolution Protocol (ARP), що служить для вирішення MAC-адрес у IP-адрес.

1.5 Канальний рівень

Канальний рівень у моделі TCP/IP відповідає за передачу даних через фізичну мережу та фізичні з'єднання між пристроями. Цей рівень виконує функції, необхідні для прямої передачі бітів через мережеве з'єднання без будь-якої обробки або перетворення даних.

Основні завдання канального рівня включають управління фізичним з'єднанням, передачу даних по мережевому кабелю або бездротовому зв'язку, а також розв'язання проблем, які можуть виникнути під час передачі даних, такі як помилки бітів.

Розглянемо основні функції канального рівня.

1.Формування кадрів: Дані, які надходять з вищих рівнів, розбиваються на кадри для передачі через мережеве середовище. Кадр містить інформацію про адресування та управління потоком даних.

2.Керування доступом до мережі: Канальний рівень може використовувати різні протоколи для керування доступом до мережі, такі як CSMA/CD (Carrier Sense Multiple Access with Collision Detection) або CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance).

3.Виявлення та виправлення помилок: Канальний рівень забезпечує механізми для виявлення та виправлення помилок, що можуть виникнути під час передачі даних по мережі. Це може включати в себе використання CRC (циклічного надлишкового коду) для перевірки цілісності даних.

4.Фізичне кодування: Дані перетворюються в сигнали, які можуть бути передані через фізичне середовище. Це включає в себе такі операції, як модуляція, демодуляція, кодування та декодування.

1.6 Переваги TCP/IP

1. Надійність передачі даних: TCP (Transmission Control Protocol) забезпечує надійну передачу даних, враховуючи перевірку на помилки, повтори та відновлення з'єднання в разі втрати пакетів.
2. Відкритий стандарт: TCP/IP є відкритим стандартом, що означає, що його можуть використовувати різні розробники програмного забезпечення та виробники обладнання для створення сумісних продуктів.
3. Масштабованість: TCP/IP легко масштабується від малих локальних мереж до великих глобальних інтернет-мереж, що робить його ідеальним для будь-яких потреб у зв'язку.
4. Гнучкість: TCP/IP підтримує різні типи мережевих топологій та технологій зв'язку, включаючи бездротові мережі, DSL та інші.
5. Швидкість передачі даних: TCP/IP може бути оптимізований для високошвидкісного передавання даних, що робить його відмінним вибором для потреб сучасних мереж.
6. Відкритість для розробки інтерфейсів: TCP/IP надає розробникам можливість створювати свої власні протоколи та інтерфейси для реалізації різноманітних функцій мережі.

РОЗДІЛ 2
Дослідження загальних принципів аналізу протоколів с за допомогою програмного додатку Wireshark

2.1 Протокол-аналізатор Wireshark

Wireshark - це один із найпопулярніших інструментів для аналізу мережевого трафіку. Це програмне забезпечення з відкритим вихідним кодом, яке дозволяє користувачам переглядати та аналізувати пакети даних, що передаються через мережу. Wireshark підтримує сотні різних протоколів і дозволяє в реальному часі або збережені у файли захоплення трафіку переглядати детально кожен пакет, що проходить через мережевий інтерфейс.

Кожен протокольний блок даних перехоплюється й аналізується програмою, відповідно до встановлених стандартів, таких як RFC. Важливо зазначити, що аналізатор трафіку може спостерігати тільки те, що проходить через мережу, до якої підключений комп'ютер. Використання комутаторів може забезпечити додатковий захист від небажаного перехоплення даних, оскільки вони керують потоком інформації між різними сегментами мережі.

Комутація пакетів - це метод передавання даних у мережах, де інформація розбивається на невеликі блоки, які називаються пакетами. Кожен пакет містить частину даних, а також метаінформацію, таку як адреса відправника, адреса одержувача та інші службові дані. Пакети можуть передаватися різними маршрутами через мережу та збиратися в оригінальне повідомлення на приймальному кінці.

Розглянемо методи перехоплення трафіку.

1. Прослуховування мережевого інтерфейсу. Це найпоширеніший спосіб перехоплення трафіку за допомогою Wireshark. Програма може слухати трафік, який проходить через конкретний мережевий інтерфейс вашого пристрою.
2. Підключенням сніфера у розрив каналу. Цей метод вимагає фізичного доступу до кабелів або безпровідних каналів, щоб вставити пристрій (сніфер) у місце розриву зв'язку.

3. Відгалуженням. Цей метод полягає в перенаправленні трафіку через проміжний пристрій (зазвичай маршрутизатор або комутатор), який має вбудовану підтримку аналізу пакетів. Wireshark потім може перехоплювати трафік, який проходить через цей пристрій.
4. Через аналіз побічних електромагнітних випромінювань. Цей метод вимагає спеціалізованого обладнання для перехоплення електромагнітних сигналів, які випромінюються від пристроїв під час передачі даних.
5. Через атаку на канальному чи мережевому рівні, що призводить до перенаправлення трафіку жертви або всього трафіку сегмента на сніфер з подальшим поверненням трафіку на належну адресу.

2.1 Завантаження та встановлення програми Wireshark

Завантаження програми здійснюється із офіційного сайту за адресою: www.wireshark.org. Зовнішній вигляд сайту Wireshark представлений на рисунку (2.1).

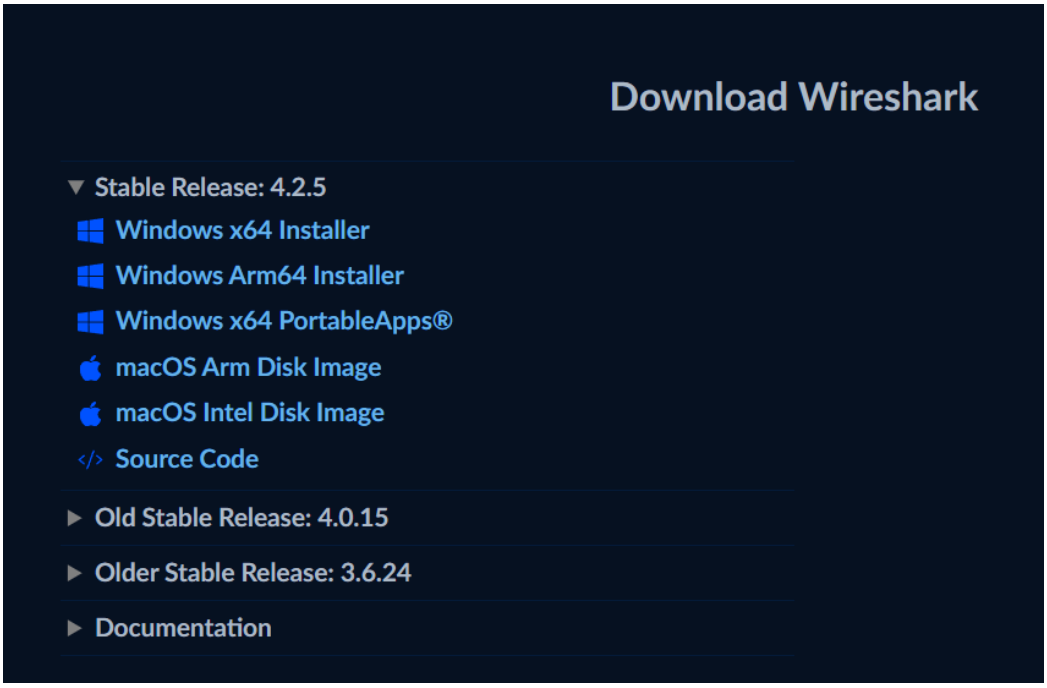


Рисунок 2.1 - Вигляд вікна "завантаження програми" Wireshark

Виберіть версію для своєї операційної системи (наприклад, Windows, macOS, або Linux). Після вибору вашої операційної системи, виберіть потрібний

пакет для завантаження. Зазвичай доступні 32-бітні і 64-бітні версії. Якщо ви не впевнені, яка версія вам потрібна, використовуйте 64-бітну версію. Клацніть на потрібну версію для розпочатку завантаження встановлювача Wireshark.

Після завершення завантаження встановлювача, відкрийте його. Якщо ви працюєте в операційній системі Windows, це зазвичай буде файл з розширенням ".exe". У macOS це може бути файл з розширенням ".dmg".

Запустіть встановлювач Wireshark, який ви завантажили на попередньому кроці. Відкриється майстер встановлення програми Wireshark. Натисніть кнопку "Next" (Далі) (рис. 2.2). Виконайте всі інструкції з інсталяції. Коли відкриється вікно License Agreement (Ліцензійна угода), натисніть кнопку "Noted" (підтвердити). Вид вікна «Ліцензійної угоди» подано на рис. 2.3. При виборі компонентів залиште стандартні налаштування та натисніть кнопку «Next» (Далі) (рис. 2.4).

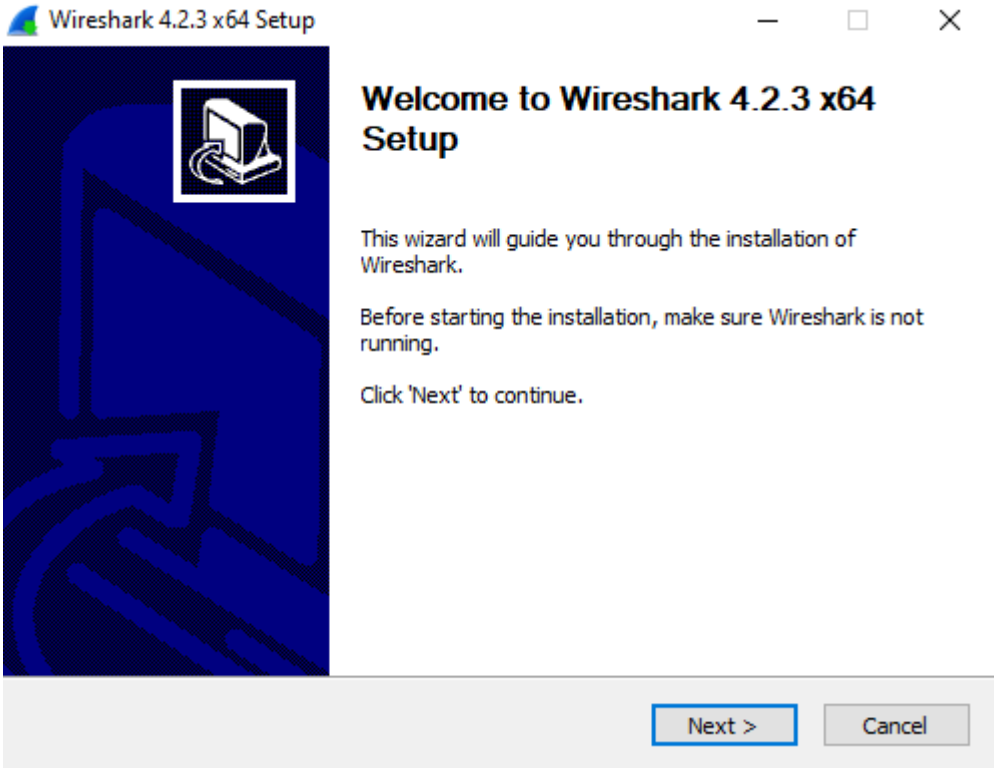


Рисунок 2.2 - Майстер встановлення програми

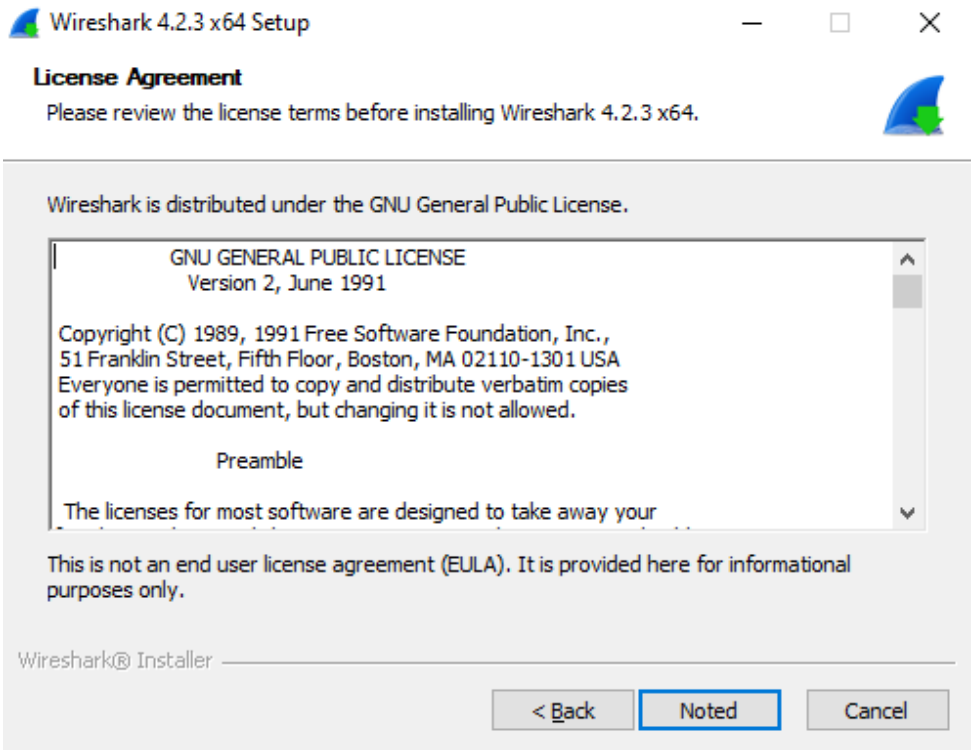


Рисунок 2.3 - Вид вікна «Ліцензійної угоди»

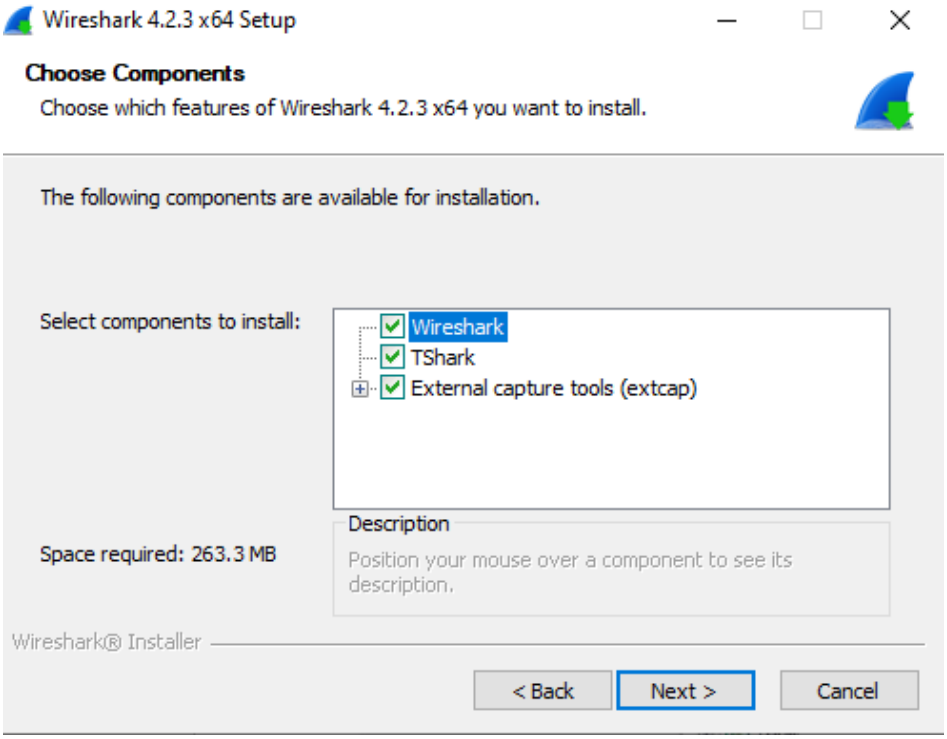


Рисунок 2.4 - Вигляд вікна програми Wireshark «Вибір компонентів»

Виберіть де розмістити бажані ярлики та натисніть кнопку "Next" (Далі) (рис. 2.5).

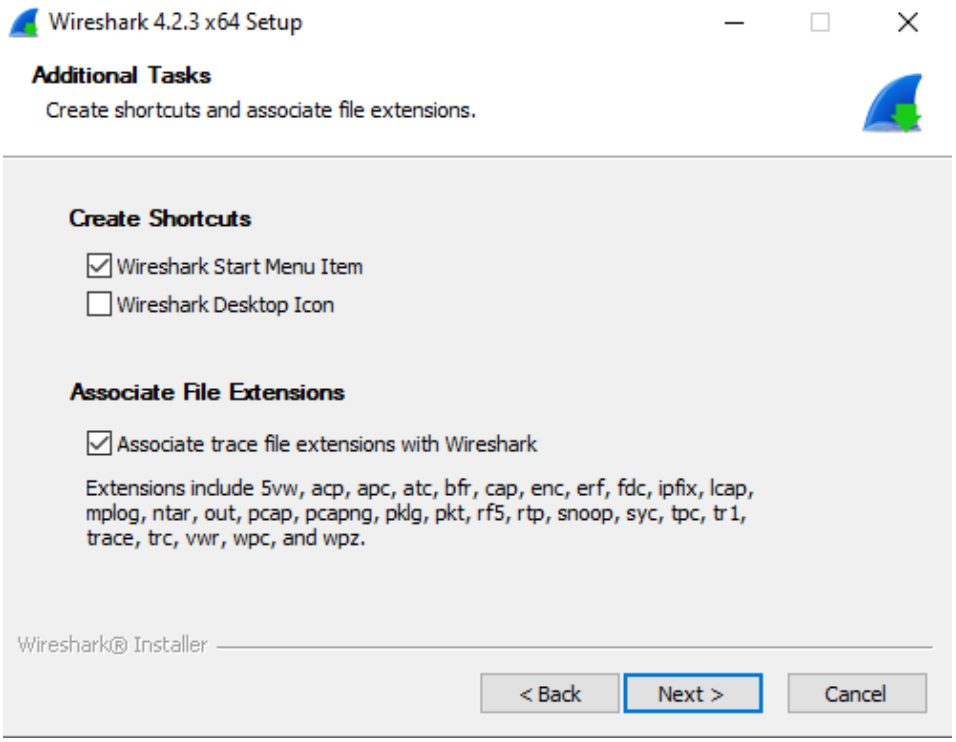


Рисунок 2.5 - Вигляд вікна програми

Якщо дисковий простір обмежений, директорію установки можна змінити, або залиште адресу, вказану за умовчанням (рис. 2.6). Після цього розпочнеться встановлення програми Wireshark. Статус встановлення відображатиметься в окремому вікні. Після завершення встановлення натисніть кнопку «Next» (Далі).

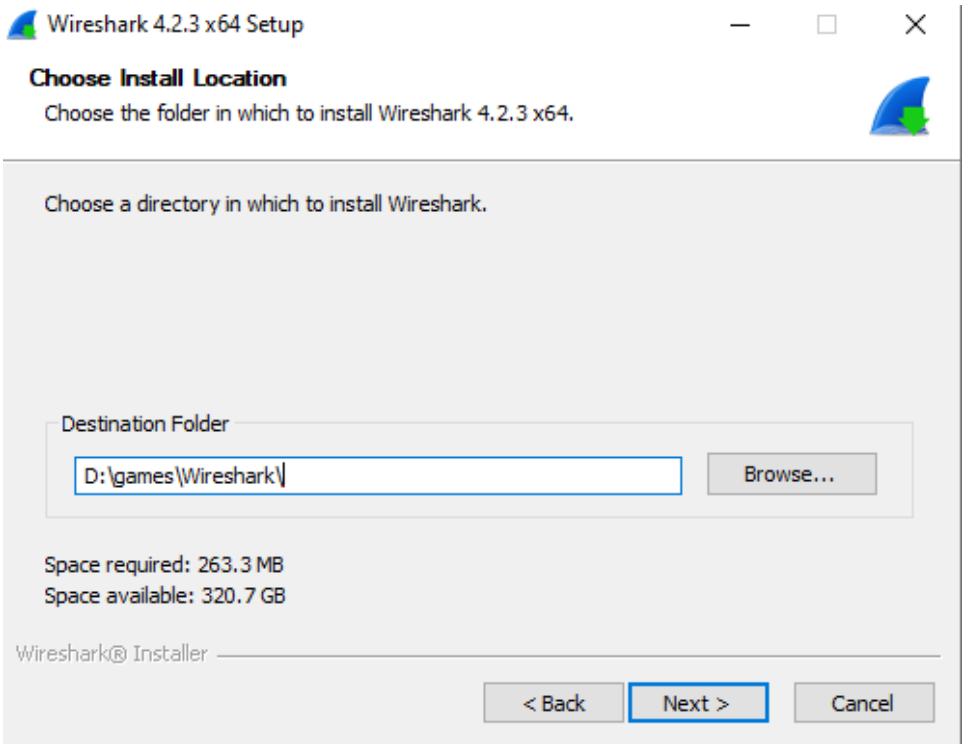


Рисунок 2.6 – Адреса встановлення програми

2.2 Налаштування програми Wireshark та запуск захоплення трафіку

Запустимо програму. Нас відразу зустрічає стартове меню (рис. 2.7), на якому можна побачити доступні для захоплення інтерфейси комп'ютера, посібники від розробників програми та безліч інших речей.

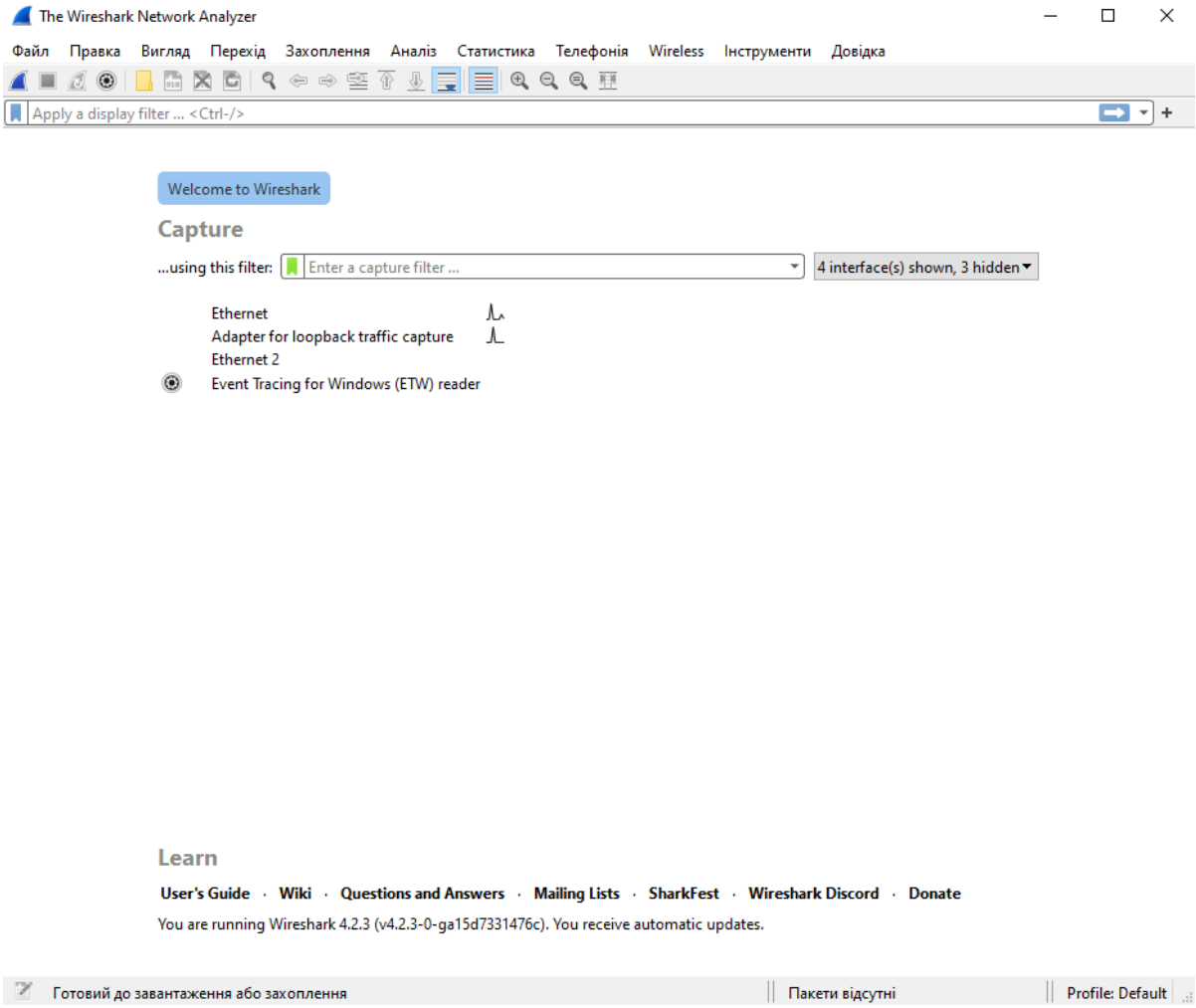


Рисунок 2.7 - Стартове меню

Розглянемо пункти меню.

1. Файл (File) - Цей пункт меню містить опції для роботи з файлами, такі як відкриття, збереження, експорт, імпорт та закриття захоплених пакетів.
2. Правка (Edit) - Тут зазвичай розташовані опції для редагування захоплених пакетів, такі як фільтрація, пошук та форматування.
3. Вигляд (View) - В цьому меню можна змінити відображення різних панелей та вікон Wireshark, налаштувати розмір шрифту та інші параметри відображення.

4. Захоплення (Capture) - Тут знаходяться опції для початку, припинення та керування процесом захоплення мережевих пакетів.
5. Аналіз (Analyze) - Цей пункт меню містить різноманітні інструменти для аналізу захоплених пакетів, включаючи статистику, фільтрацію та інші аналітичні функції.
6. Статистика (Statistics) - Тут можна переглянути різні статистичні дані про мережевий трафік, такі як загальна кількість пакетів, статистика протоколів тощо.
7. Інструменти (Tools) - В цьому меню містяться додаткові інструменти та утиліти, які можуть бути корисними для аналізу мережі або роботи з захопленими пакетами.
8. Довідка (Help) - Тут розташовані посилання на довідкову інформацію, документацію та інші ресурси, які можуть допомогти користувачам з різними питаннями щодо використання Wireshark.

Тут вам потрібно обрати мережевий інтерфейс, який підключений до Інтернету. Мережевий інтерфейс - це програмне забезпечення, яке співпрацює з мережним драйвером та IP-рівнем, забезпечуючи доступ до всіх наявних мережних адаптерів, трафік яких ми будемо аналізувати. Зазвичай у програмі Wireshark можна обрати мережевий інтерфейс для бездротового (Wi-Fi) або кабельного (Ethernet) з'єднання. Виберіть "Ethernet" для захоплення, а потім натисніть "Start". Якщо ви обрали правильний інтерфейс, ви побачите наступне. (рис. 2.8).

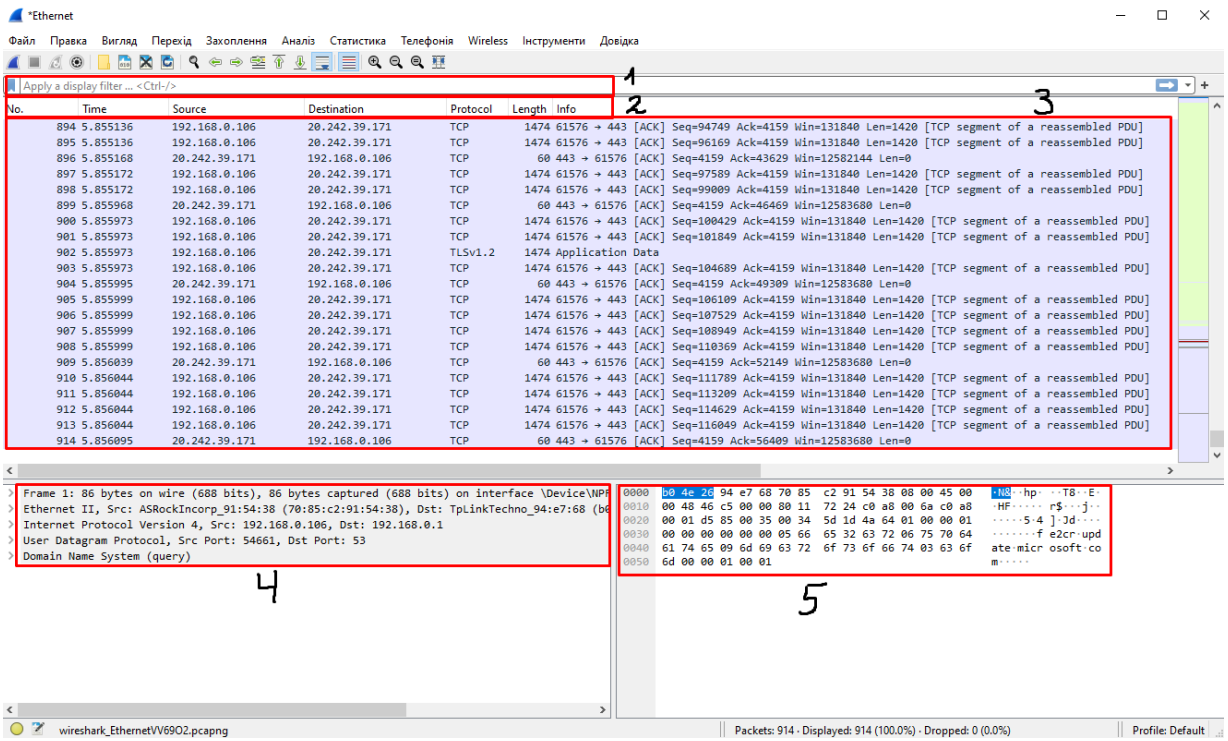


Рисунок 2.8 - Огляд початку захоплення трафіку

Розглянемо докладніше це вікно за пунктами, вказаними на ньому.

1. Панель фільтрів, що дозволяє користувачам застосовувати різні фільтри до трафіку мережі.
2. Панель найменувань, що поділяє інформацію з пункту 3 на номер, час від початку захоплення трафіку, джерело та адресат, а також протокол, розмір пакета і невелику інформацію про мережевий пакет.
3. Панель пакетів, що відображає список всіх пакетів, які були захоплені Wireshark.
4. Панель рівнів, показує структуру кожного пакета на основі моделі OSI.
5. Панель метаданих, що представляє дані у шістнадцятковому коді та символах.

Тепер можна побачити пакети даних, що проходять по мережі, а також деяку інформацію про них: адреси відправника та одержувача, протоколи та вміст пакета.

При проведенні дослідження трафіку може виникнути проблема з пошуком потрібного пакета. Для вирішення цієї проблеми в програмі Wireshark існує можливість фільтрації. У спеціальному полі «Filter» (рис. 2.9) можна ввести необхідні команди або скористатися підказками.

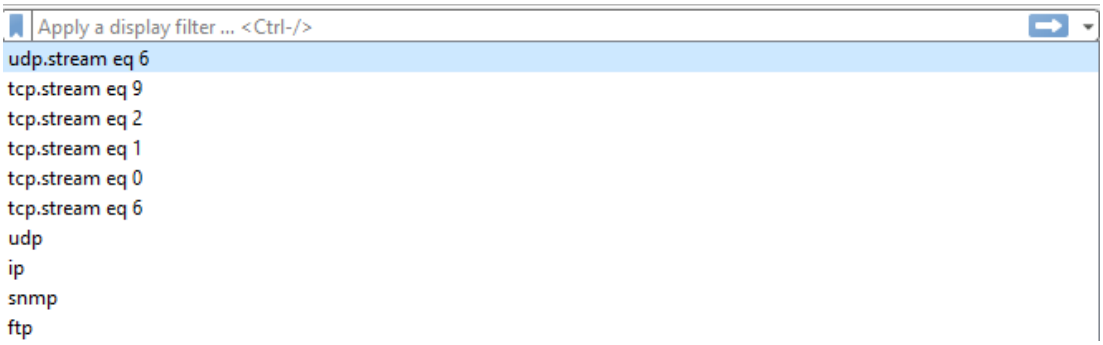


Рисунок 2.9 – Огляд панелі фільтрів

Найчастіше використовується фільтрація за IP-адресами, за номерами портів та протоколами. Фільтрування за IP-адресою дозволяє нам переглядати всі пакети, що надходять від кого-небудь або ті, що йдуть будь-кому. Наприклад, відберемо всі пакети, що надходять від IP-адреси 192.168.0.106 за допомогою введення у фільтрі «ip.src == x.x.x.x» (рис. 2.10).

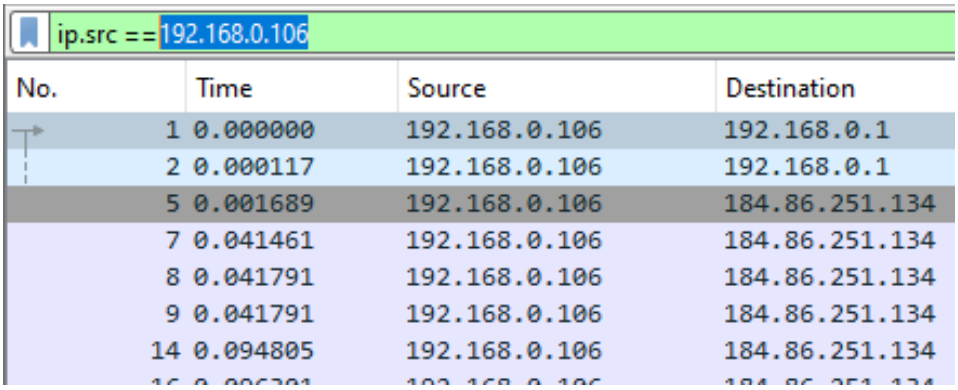


Рисунок 2.10 – Огляд команди ip.src

Також можна відфільтрувати трафік мережі IP-адресою одержувача пакетів за допомогою команди «ip.dst == x.x.x.x» (рис. 2.11).

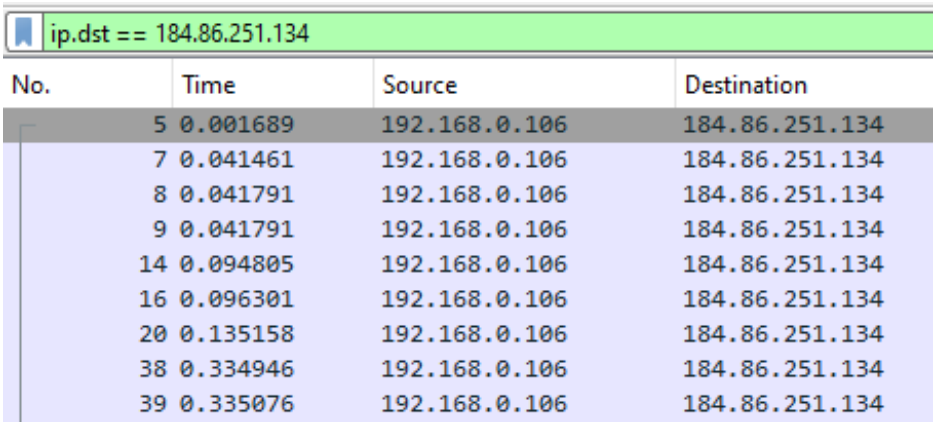


Рисунок 2.11 – Огляд команди ip.dst

Крім того, можна побачити пакети незалежно від напрямку трафіку за допомогою «ip.addr == x.x.x.x.» (рис. 2.11).

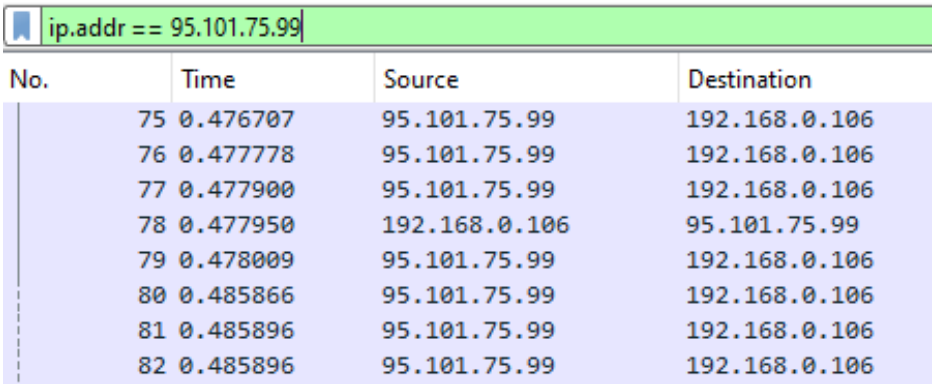


Рисунок 2.11 – Огляд команди ip.addr

І, найголовніше, для фільтрації трафіку за пакетами протоколів необхідно просто ввести назву протоколу (рис. 2.12).

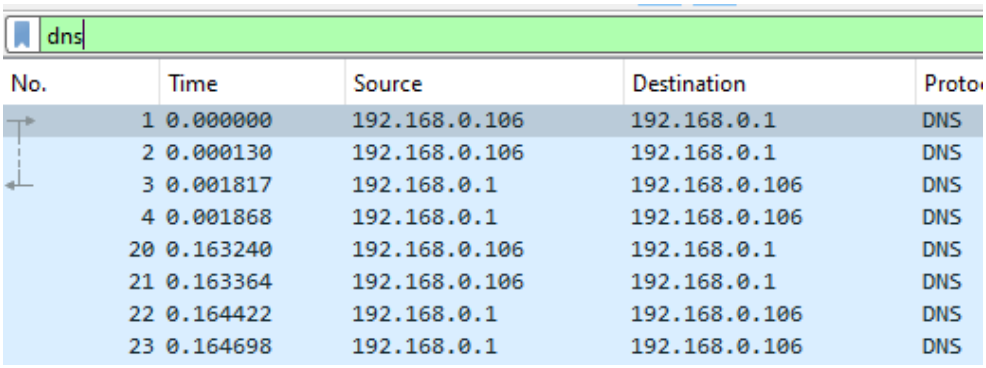


Рисунок 2.12 – Огляд фільтрації за протоколом

Тепер можна розпочати аналіз мережевого трафіку.

РОЗДІЛ 3
Дослідження протоколів стеку TCP/IP

3.1 Дослідження протоколу FTP

File Transfer Protocol (протокол передачі файлів) або просто FTP - мережевий протокол, призначений для передачі файлів в комп'ютерних мережах. Протокол FTP дозволяє підключатися до серверів FTP, переглядати традиційні каталоги і завантажувати файли з сервера або на сервер, крім цього можливий режим передачі файлів між серверами.

Протокол не шифрується, при аутентифікації передає логін та пароль відкритим текстом. Якщо зловмисник знаходиться в одному сегменті мережі з користувачем FTP, то, використовуючи сніфер (наприклад, Wireshark), він може перехопити логін і пароль користувача, або, за наявності спеціального програмного забезпечення, отримувати файли, що передаються по FTP без авторизації. Щоб запобігти перехопленню трафіку, необхідно використовувати протокол шифрування даних SSL.

За замовчуванням протокол FTP працює на портах 20 (пересилання даних) та 21 (пересилання команд). FTP відрізняється від решти протоколів TCP/IP тим, що команди (табл. 3.1) можуть передаватися одночасно з передачею даних у реальному часі; в інших протоколів такої можливості.

Таблиця 3.1 – Команди протоколу FTP

Команди	Призначення
USER	вказати ім'я користувача
PASS	вказати пароль
LIST	перегляд вмісту каталогу
CWD	зміна поточного каталогу
PETR	передати файл із сервера на клієнт
STOR	передати файл з клієнта на сервер
TYPE	встановити режим передачі
DELE	видалити файл
MKD	створити каталог
RMD	видалити каталог
PASV	використовувати пасивний режим
QUIT	вихід та розрив з'єднання

Кожен запит (рис. 3.1) є командою, в якій можуть бути аргументи. Відповіді (рис. 3.2) включають код запиту і запитувані дані.

```
▼ File Transfer Protocol (FTP)
  ▼ USER anonymous\r\n
      Request command: USER
      Request arg: anonymous
  [Current working directory: ]
```

Рисунок 3.1 - Приклад запиту

```
▼ File Transfer Protocol (FTP)
  ▼ 331 Guest login ok, type your name as password.\r\n
      Response code: User name okay, need password (331)
      Response arg: Guest login ok, type your name as password.
  [Current working directory: ]
```

Рисунок 3.2 – Приклад відповіді

У Wireshark трафік можна ідентифікувати за допомогою фільтра ftp (рис. 3.3).

```
220-
220 6bone.informatik.uni-leipzig.de FTP server (NetBSD-ftpd 20041119) ready.
USER anonymous
331 Guest login ok, type your name as password.
PASS IEUser@
230 Guest login ok, access restrictions apply.
opts utf8 on
502 Unknown command 'utf8'.
syst
215 UNIX Type: L8 Version: NetBSD-ftpd 20041119
site help
214-
```

Рисунок 3.3 - Приклад FTP-трафіку, відфільтрованого за допомогою Wireshark

Поетапно проаналізуємо запити та відповіді FTP сервера.

1. 220-220 6bone.informatik.uni-leipzig.de FTP server (NetBSD-ftpd 20041119) ready.
   Сервер відповідає кодом 220, що означає, що готовий до з'єднання. У цій відповіді також міститься ім'я сервера та версія програмного забезпечення.
2. USER anonymous331 Guest login ok, type your name as password.
   Клієнт намагається увійти під ім'ям користувача "anonymous". Сервер відповідає коду 331, що означає, що користувач може продовжити процес входу, надавши пароль.
3. PASS IEUser@ 230 Guest login ok, access restrictions apply.
   Клієнт надсилає пароль "IEUser@". Сервер відповідає кодом 230, що означає успішний вхід під гостьовим обліковим записом, але з певними обмеженнями доступу.
4. opts utf8 on 502 Unknown command 'utf8'.
   Клієнт намагається включити підтримку UTF-8 за допомогою команди opts utf8 on. Сервер відповідає кодом 502, що означає, що команда не розпізнана чи не підтримується.
5. Syst 215 UNIX Type: L8 Version: NetBSD-ftpd 20041119

Клієнт запитує інформацію про тип системи за допомогою команди "syst". Сервер відповідає кодом 215, надаючи інформацію про систему, вказуючи, що це UNIX-система з версією NetBSD-ftpd 20041119.

6. site help 214-

Клієнт запитує довідкову інформацію про доступні команди сервера за допомогою команди "site help". Відповідь починається з коду 214, що означає, що буде надана довідкова інформація.

У сукупності, обмін запитами та відповідями показує стандартний процес аутентифікації та обміну інформацією між FTP клієнтом та сервером. Сервер підтримує базові команди, але не розпізнає деякі додаткові команди, такі як налаштування UTF-8.

3.2 Дослідження протоколу SMTP

Simple Mail Transfer Protocol (SMTP) - це один із ключових протоколів для передачі електронної пошти. Він призначений для пересилання повідомлень від сервера відправника до сервера отримувача.

Принцип роботи SMTP базується на простому обміні даними між серверами, що дозволяє ефективно доставляти електронні листи з одного кінця світу на інший. SMTP виконує важливу роль у процесі відправлення електронних листів, але він не обробляє вхідні повідомлення. Це означає, що його основна функція полягає у відправці та подальшій доставці електронних листів до адресатів. Коли ви надсилаєте електронний лист, ваш поштовий клієнт використовує SMTP для передачі повідомлення на поштовий сервер, який потім пересилає його до сервера отримувача, де воно стає доступним для прочитання.

Особливо часто SMTP використовують для масових та транзакційних розсилок. Завдяки своїй простоті та надійності, SMTP залишається одним з основних протоколів для відправки електронної пошти.

Надсилання листів по SMTP відбувається через отримання відповідей на команду (табл. 3.2).

Таблиця 3.2 – Команди протоколу SMTP

Команда SMTP	Призначення
AUTH	Вказує механізм ідентифікації для SMTP-сервера.
DATA	Означає початок вмісту поштового повідомлення.
EHLO	Активує розширення SMTP.
EXPN	Запитує одержувача підтвердження ідентифікації списку розсилки.
HELO	Визначає відправника SMTP повідомлення.
HELP	Запитує у отримувача довідкову інформацію для відправника.
MAIL	Починає поштову транзакцію для доставки поштового повідомлення до одного або кількох одержувачів.
NOOP	Запитує у отримувача допустиму відповідь.
QUIT	Вказує, що одержувач повинен надіслати відповідь і потім закрити канал передачі.
RCPT	Визначає одержувача поштового повідомлення.
RSET	Завершує поштову транзакцію.
SAML	Доставляє пошту на одну або кілька робочих станцій або одержувачам, якщо користувач неактивний.
SEND	Доставляє пошту одну чи кілька робочих станцій.
SOML	Доставляє пошту на одну або кілька робочих станцій або одержувачам, якщо користувач неактивний.
STARTTLS	Звертається до сервера SMTP із командою запустити узгодження (SSL) або TLS із клієнтом SMTP.
TURN	Вказує, що одержувач повинен або надати допустиму відповідь і перейти в режим надсилання SMTP, або відмовитися і залишитися в режимі SMTP.
VRFY	Запитує одержувача підтвердження ідентифікації користувача.

Розглянемо кроки відправлення листа через SMTP.

1. Сервер відправника отримує необхідну інформацію та шукає сервер на стороні одержувача для передачі листа: за адресою електронної пошти одержувача визначає поштового провайдера та запитує IP-адресу SMTP-сервера одержувача.
2. Сервер встановлює з'єднання через порт 25 і передає лист серверу одержувача. Якщо сервер одержувача не відповідає, виконується ще кілька спроб встановити з'єднання.
3. Якщо відповіді немає, сервер відправника повертає помилку відправлення.

Для роботи з протоколом SMTP стандартно застосовують 25 портів. Але щоб уникнути спам-розсилок, провайдери можуть закрити до нього доступ. Тоді його замінюють додатковими портами:

1. 465 - застосовують для створення захищеного SSL-з'єднання;
2. 587 - використовують для запобігання вихідному спаму за рахунок обов'язкової автентифікації відправника.

Поетапно проаналізуємо запити та відповіді FTP сервера (рис. 3.4).

```
220-xc90.websitewelcome.com ESMTP Exim 4.69 #1 Mon, 05 Oct 2009 01:05:54 -0500
220-We do not authorize the use of this system to transport unsolicited,
220 and/or bulk e-mail.
EHLO GP
250-xc90.websitewelcome.com Hello GP [122.162.143.157]
250-SIZE 52428800
250-PIPELINING
250-AUTH PLAIN LOGIN
250-STARTTLS
250 HELP
AUTH LOGIN
334 VXNlcm5hbWU6
Z3VycGFydGFwQHBhdHJpb3RzLmlu
334 UGFzc3dvcmQ6
cHVuamFiQDEyMw==
235 Authentication succeeded
MAIL FROM: <gurpartap@patriots.in>
250 OK
RCPT TO: <raj_deol2002in@yahoo.co.in>
250 Accepted
DATA
354 Enter message, ending with "." on a line by itself
From: "Gurpartap Singh" <gurpartap@patriots.in>
To: <raj_deol2002in@yahoo.co.in>
Subject: SMTP
Date: Mon, 5 Oct 2009 11:36:07 +0530
Message-ID: <000301ca4581$ef9e57f0$cedb07d0$@in>
MIME-Version: 1.0
Content-Type: multipart/mixed;
          boundary="----=_NextPart_000_0004_01CA45B0.095693F0"
X-Mailer: Microsoft Office Outlook 12.0
Thread-Index: AcpFgem9BvjjZEDeR1Kh8i+hUyVo0A==
Content-Language: en-us
x-cr-hashedpuzzle: SeA= AAR2 ADaH BpiO C4G1 D1gW FNB1 FPkR Fn+W HFCP HnYJ JO7s K
osha1_v1;7;{CAA37F59-1850-45C7-8540-AA27696B5398};ZwB1AHIAcABhAHIAdABhAHAAQABwAG
x-cr-puzzleid: {CAA37F59-1850-45C7-8540-AA27696B5398}

This is a multipart message in MIME format.
```

Рисунок 3.4 - Приклад SMTP-трафіку, відфільтрованого за допомогою Wireshark

1. 220-xc90.websitewelcome.com ESMTP Exim 4.69 #1 Mon, 05 Oct 2009 01:05:54 -0500

   220-We do not authorize the use of this system to transport unsolicited,

   220 and/or bulk e-mail.

   Сервер вітає клієнта, повідомляючи, що він використовує Exim 4.69. Він також повідомляє, що не дозволяє використовувати цю систему для передачі небажаної або масової пошти.
2. EHLO GP

   250-xc90.websitewelcome.com Hello GP [122.162.143.157]

250-SIZE 52428800

250-PIPELINING

250-AUTH PLAIN LOGIN

250-STARTTLS

250 HELP

3. Клієнт надсилає команду EHLO з ім'ям хоста "GP". Сервер відповідає, що він готовий приймати з'єднання від цього клієнта, та перераховує підтримувані можливості: максимальний розмір повідомлення (52428800 байт), підтримка командного конвеєра (PIPELINING), методи автентифікації (PLAIN, LOGIN), підтримка шифрування (STARTTLS), та можливість отримати допомогу (HELP).
4. AUTH LOGIN

   334 VXNlcm5hbWU6

   Клієнт запитує автентифікацію за допомогою методу LOGIN. Сервер відповідає, запитуючи ім'я користувача (у базовому64 форматі).
5. Z3VycGFydGFwQHBhdHJpb3RzLmlu

   334 UGFzc3dvcmQ6

   Клієнт надсилає ім'я користувача, закодоване у Base64 ("Z3VycGFydGFwQHBhdHJpb3RzLmlu" декодується як "gurpartap@patriots.in"). Сервер відповідає, запитуючи пароль (також у Base64).
6. cHVuamFiQDEyMw==

   235 Authentication succeeded

   Клієнт надсилає пароль, закодований у Base64 ("cHVuamFiQDEyMw==" декодується як "punjab@123"). Сервер відповідає, що автентифікація успішна.
7. MAIL FROM: <gurpartap@patriots.in>

   250 OK

   Клієнт вказує адресу відправника. Сервер підтверджує, що команда прийнята.
8. RCPT TO: <raj_deol2002in@yahoo.co.in>

   250 Accepted

   Клієнт вказує адресу отримувача. Сервер підтверджує, що команда прийнята.
9. DATA

354 Enter message, ending with "." on a line by itself

Клієнт запитує дозвіл на надсилання тіла повідомлення. Сервер відповідає, що готовий прийняти дані, вказуючи, що завершення даних повинно бути позначено окремим рядком з однією точкою.

10. From: "Gurpartap Singh" gurpartap@patriots.in

Відправник електронного листа.

11. To: raj_deol2002in@yahoo.co.in

Отримувач електронного листа.

12. Subject: SMTP

Тема повідомлення, короткий опис його змісту.

13. Date: Mon, 5 Oct 2009 11:36:07 +0530

Дата і час відправлення повідомлення з урахуванням часової зони (+0530).

14. Message-ID: 000301ca4581$ef9e57f0$cedb07d0$@in

Унікальний ідентифікатор повідомлення, використовуваний для відстеження.

15. MIME-Version: 1.0

Версія MIME, яка використовується для форматування листа (1.0).

16. Content-Type:multipart/mixed;boundary="----=_NextPart_000_0004_01CA45B0.095693F0"

Тип вмісту повідомлення. Вказує, що повідомлення є багаточастинним (multipart/mixed), і кожна частина розділена межовим рядком "----=_NextPart_000_0004_01CA45B0.095693F0".

17. X-Mailer: Microsoft Office Outlook 12.0

Програма для надсилання листа (Microsoft Office Outlook 12.0).

18. Thread-Index: AcpFgem9BvjjZEDeR1Kh8i+hUyVo0A==

Індекс нитки для відстеження обговорень або ланцюгів листів.

19. Content-Language: en-us

Мова вмісту повідомлення (англійська - США).

20. x-cr-hashedpuzzle і x-cr-puzzleid:

Заголовки, які можуть бути специфічними для поштового сервера або клієнта. Вони, використовуються для захисту або автентифікації повідомлення.

21. This is a multipart message in MIME format.

Це повідомлення, яке складається з кількох частин. Рядок повідомляє про те, що вміст має кілька частин.

Ця сесія показує типову процедуру відправки електронного листа через SMTP-сервер з автентифікацією. Клієнт вказує серверу свої облікові дані, після чого відправляє повідомлення. Сервер приймає дані і вказує, що повідомлення успішно передано.

3.3 Дослідження протоколу TCP

Протокол TCP (Transmission Control Protocol, протокол керування передачею) був розроблений для забезпечення надійного та точного обміну даними між комп'ютерами в одній мережі. Він є сполучною ланкою між додатками користувача та нижчими рівнями протоколів, такими як IP, які відповідають за маршрутизацію та адресацію пакетів даних.

В операційній системі TCP зазвичай реалізується як окремий системний модуль (драйвер), через який проходять всі виклики функцій протоколу. Інтерфейс між користувацьким додатком і TCP представлений бібліотекою викликів, яка схожа на бібліотеку системних викликів для роботи з файлами. Наприклад, можна відкрити або закрити з'єднання (аналогічно відкриттю або закриттю файлу) і відправити або прийняти дані через це з'єднання (як операції читання та запису файлу). Ці виклики можуть виконуватися в асинхронному режимі, що дозволяє не блокувати роботу програми під час очікування відповіді.

Кожна реалізація TCP може мати свої особливості, але всі вони повинні відповідати стандартам TCP. Користувач взаємодіє з TCP наступним чином: для передачі даних процес викликає відповідну функцію TCP, вказуючи на буфер з даними. TCP розбиває ці дані на сегменти та передає їх протоколу нижчого рівня, наприклад, IP. На приймаючій стороні TCP збирає дані з сегментів, перевіряє їх цілісність та передає їх у користувацький процес, повідомляючи відправника про успішну доставку.

Користувацький інтерфейс з TCP дозволяє виконувати команди, такі як відкрити (OPEN) або закрити (CLOSE) з’єднання, відправити (SEND) або прийняти (RECEIVE) дані, а також отримати статус з’єднання (STATUS). Ці команди аналогічні системним викликам для роботи з файлами.

У моделі мережевого взаємодії зв’язок між TCP і протоколами нижчого рівня не є специфікованим, за винятком того, що має бути механізм для асинхронної передачі даних між рівнями. Цей механізм реалізує інкапсуляцію даних вищого рівня у протоколи нижчого рівня через інтерфейс викликів між TCP та IP. В результаті, кожен TCP-пакет вкладається в "конверт" протоколу нижчого рівня, наприклад, IP, створюючи дейтаграму, що містить TCP-пакет з користувацькими даними.

Таким чином, TCP забезпечує надійний обмін даними, що включає перевірку цілісності, контроль помилок та підтвердження доставки, дозволяючи додаткам користувача працювати без турбот про втрати або пошкодження даних під час передачі. Це робить TCP основним протоколом для багатьох критично важливих мережевих сервісів та застосунків, де важлива надійність передачі даних.

Поетапно проаналізуємо пакет протоколу TCP (рис. 3.5).

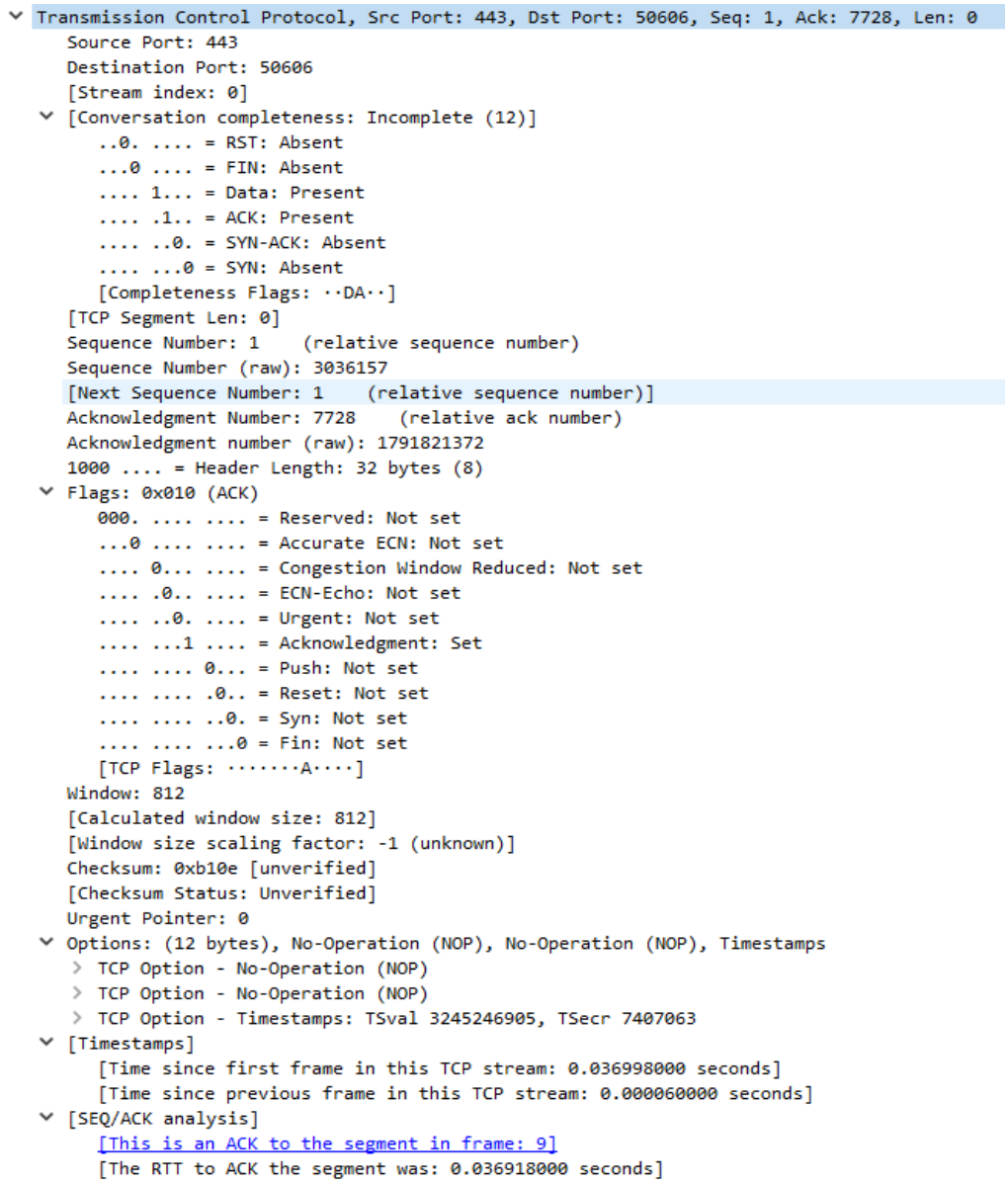


Рисунок 3.5 – Приклад пакету протоколу TCP

1. Заголовок TCP-пакету.
    1. Source Port (Порт джерела): 443 (це звичайно порт HTTPS, що означає, що з'єднання, ймовірно, використовує протокол HTTPS).
    2. Destination Port (Порт призначення): 50606 (це, ймовірно, випадковий високий порт на стороні клієнта).
    3. Sequence Number (Послідовний номер): 1 (це відносний послідовний номер, що вказує на початок передачі даних).

4. Acknowledgment Number (Номер підтвердження): 7728 (відносний номер, що підтверджує отримання даних від іншого хоста).

2. Статус з'єднання.

   Length (Довжина): 0 (цей пакет не містить даних, він тільки підтверджує отримання даних).

   Flags (Прапори):

   1. ACK (Підтвердження): установлений (пакет підтверджує отримання даних).
   2. SYN, FIN, RST: не встановлені (немає початку чи завершення з'єднання, немає скидання з'єднання).

3. Вікно.

   1. Window Size (Розмір вікна): 812 (розмір вікна TCP, що визначає кількість даних, які можуть бути надіслані без підтвердження).
   2. Checksum (Контрольна сума): 0xb10e (не перевірено, це стандартна частина пакету для виявлення помилок).

4. Опції.

   1. No-Operation (NOP): Два NOP, використовуються для вирівнювання.
   2. Timestamps (Мітки часу): TSval 3245246905, TSecr 7407063 (мітки часу для вимірювання часу проходження пакетів та RTT).

5. SEQ/ACK аналіз.

   RTT (Round Trip Time): 0.036918000 секунд (час, який потрібен для передачі пакету і отримання підтвердження).

Загальний висновок: цей TCP-пакет є підтвердженням (ACK) попередньо отриманих даних. Порт джерела 443 свідчить про те, що з'єднання йде через HTTPS. Пакет не містить даних (Length: 0), але підтверджує отримання сегменту з номером підтвердження 7728. RTT вказує на час, який потрібен для передачі даних між двома хостами.

3.4 Дослідження протоколу UDP

UDP (User Datagram Protocol) - це один з основних транспортних протоколів Інтернету, який використовується для передачі даних. Цей протокол був створений для забезпечення простого і швидкого способу передачі даних між пристроями в мережі.

Основна характеристика UDP полягає в тому, що це безз'єднувальний протокол, тобто не встановлює постійне з'єднання між відправником і отримувачем перед передачею даних. Це значно спрощує процес передачі, але водночас накладає певні обмеження і недоліки. UDP не встановлює з'єднання перед передачею даних, що зменшує затримки і збільшує швидкість передачі. Кожен пакет (датаграма) відправляється незалежно один від одного, що означає, що вони можуть прибувати у будь-якому порядку.

Цей протокол не гарантує доставку пакетів. Пакети можуть бути загублені, дубльовані або прибувати в неправильному порядку. Відсутність механізмів для повторної передачі втрачених пакетів робить його менш надійним порівняно з TCP.

Протокол не має механізмів для управління потоком даних або контролю завантаження мережі. Це може призвести до перевантаження мережі та втрати пакетів при великому навантаженні.

UDP часто використовується в ситуаціях, коли швидкість передачі даних і мінімальні затримки важливіші за надійність. Типові випадки використання включають стрімінг відео та аудіо, онлайн-ігри та DNS-запити. У реальному часі важливо доставляти дані швидко, навіть якщо деякі пакети втрачаються. Відповіді на запити DNS повинні бути швидкими, і втрачений пакет можна просто повторити.

UDP використовує заголовок, який складається з чотирьох основних полів: порт відправника, порт одержувача, довжина дейтаграми і контрольна сума. Ці поля забезпечують базову інформацію, необхідну для доставки даних.

Проаналізуємо кожен із них.

1. Порт відправника ідентифікує джерело повідомлення. Це 16-бітове число, яке вказує на конкретний порт на комп'ютері відправника. Порт використовується для того, щоб отримувач міг знати, звідки прийшло повідомлення і, при необхідності, відправити відповідь назад на той же порт. Якщо порт відправника не використовується, він може бути встановлений в нуль.
2. Порт одержувача також є 16-бітовим числом і вказує на порт, на який повинна бути доставлена дейтаграма на комп'ютері отримувача. Це забезпечує правильну маршрутизацію даних до відповідного застосунку або сервісу на стороні отримувача. Кожен порт відповідає певному процесу або застосунку, який прослуховує на цьому порту.
3. Довжина дейтаграми - це 16-бітове поле, яке вказує загальну довжину UDP-пакету, включаючи заголовок і дані. Мінімальна довжина UDP-пакету становить 8 байт (розмір заголовка). Це поле дозволяє отримувачу визначити, де закінчуються дані, що передаються, і які байти слід ігнорувати, якщо довжина повідомлення менша за довжину прийнятого пакету.
4. Контрольна сума використовується для перевірки цілісності даних. Це 16-бітове поле, яке дозволяє відправнику та отримувачу перевірити, чи не було пошкоджено дейтаграму під час передачі. Контрольна сума обчислюється на основі заголовка і даних дейтаграми перед її відправкою. Отримувач повторно обчислює контрольну суму на основі отриманих даних і порівнює її з переданою. Якщо контрольні суми не співпадають, дейтаграма вважається пошкодженою і може бути відкинута. В деяких випадках, якщо контрольна сума не використовується, це поле може бути заповнене нулями.

5.

Поетапно проаналізуємо пакет протоколу UDP (рис. 3.6).

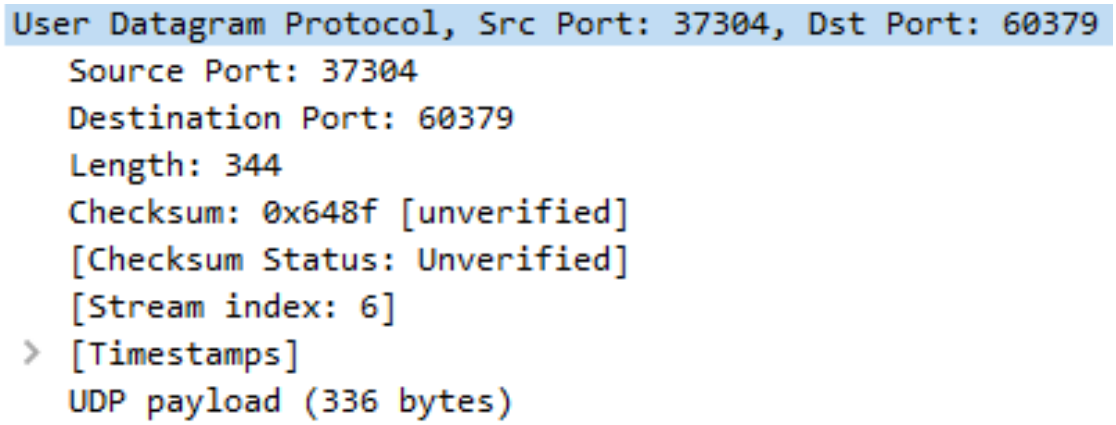


Рисунок 3.6 – приклад пакету протоколу UDP

1. Source Port (Порт відправника): 37304

   Це означає, що пакет був надісланий з додатка або служби, яка використовує порт 37304 на відправнику.
2. Destination Port (Порт призначення): 60379

   Це означає, що пакет призначений для додатка або служби, яка слухає порт 60379 на отримувачі.
3. Length (Довжина): 344

   Це загальна довжина пакета в байтах, включаючи заголовок та дані.
4. Checksum (Контрольна сума): 0x648f [unverified]

   Це означає, що достовірність даних в цьому пакеті може бути не підтверджена, оскільки контрольна сума не була перевірена.

Разом ці поля забезпечують базові функції для доставки даних через мережу з використанням UDP. Протокол залишається простим і ефективним завдяки мінімалістичному заголовку і відсутності додаткових механізмів, що забезпечують надійність, таких як повторна передача або підтвердження отримання.

3.5 Дослідження протоколу IP

Протокол IP (Internet Protocol) є основою мережевого спілкування в стеку TCP/IP, забезпечуючи адресацію та маршрутизацію пакетів даних між пристроями в мережі. Він працює на мережевому рівні та є відповідальним за доставку пакетів від відправника до одержувача.

IP визначає унікальні адреси для кожного пристрою в мережі, що дозволяє правильно ідентифікувати відправника та отримувача даних. Ці адреси можуть бути IPv4 або IPv6 формату.

1. IPv4 (Internet Protocol version 4).
    1. Адресація: 32-бітні адреси, що дозволяє мати близько 4,3 мільярда унікальних IP-адрес.
    2. Формат адреси: Представляється як чотири десяткових числа, розділених точками (наприклад, 192.168.0.1).
    3. Структура пакета: Включає заголовок (20-60 байт) і корисні дані (до 65,535 байт). Заголовок містить такі поля, як IP-адреси відправника і отримувача, час життя (TTL), протокол верхнього рівня (наприклад, TCP, UDP) і контрольна сума.
2. IPv6 (Internet Protocol version 6).
    1. Адресація: 128-бітні адреси, що дозволяє мати 2^128 унікальних IP-адрес, що практично усуває проблему вичерпання адрес, актуальну для IPv4.
    2. Формат адреси: Представляється як вісім груп шістнадцяткових чисел, розділених двокрапками (наприклад, 2001:0db8:85a3:0000:0000:8a2e:0370:7334).
    3. Структура пакета: Включає фіксований заголовок (40 байт) і корисні дані. Заголовок спрощений у порівнянні з IPv4 і не містить поля контрольної суми, що прискорює обробку пакетів.

Під час передачі даних IP розділяє інформацію на пакети, кожен з яких містить заголовок і корисне навантаження. Заголовок включає інформацію про джерело та місце призначення, а також інші метадані, необхідні для

маршрутизації. Пакети можуть проходити через різні маршрутизатори, які вирішують, яким шляхом має йти кожен пакет, базуючись на поточних умовах мережі.

Оскільки IP є безз'єднувальним протоколом, він не гарантує доставку пакетів і не відстежує їх. Це означає, що пакети можуть бути втрачені, пошкоджені або доставлені в неправильному порядку. Ці питання вирішуються іншими протоколами в стеку TCP/IP, такими як TCP (Transmission Control Protocol), який забезпечує надійну передачу даних, використовуючи механізми встановлення з'єднання, підтвердження отримання та повторної передачі втрачених пакетів.

Поетапно проаналізуємо пакет протоколу IP (рис. 3.7).

```
Internet Protocol Version 4, Src: 2.21.89.11, Dst: 192.168.0.106
   0100 .... = Version: 4
   .... 0101 = Header Length: 20 bytes (5)
 ⌄ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
      0000 00.. = Differentiated Services Codepoint: Default (0)
      .... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
   Total Length: 1500
   Identification: 0xb231 (45617)
 ⌄ 010. .... = Flags: 0x2, Don't fragment
      0... .... = Reserved bit: Not set
      .1.. .... = Don't fragment: Set
      ..0. .... = More fragments: Not set
   ...0 0000 0000 0000 = Fragment Offset: 0
   Time to Live: 59
   Protocol: TCP (6)
   Header Checksum: 0x6bb8 [validation disabled]
   [Header checksum status: Unverified]
   Source Address: 2.21.89.11
   Destination Address: 192.168.0.106
```

Рисунок 3.7 – приклад пакету протоколу IP.

1. Version (Версія): 4 - це номер версії IP-протоколу, у даному випадку, це IPv4.
2. Header Length (Довжина заголовка): 20 байтів (5 слів): це розмір заголовка IP-пакета у словах (кожне слово - 4 байти), або у байтах. У даному випадку, заголовок має розмір 20 байт.
3. Differentiated Services Field (Поле відмінених послуг): 0x00 - це поле вказує на рівень обслуговування (QoS) та експериментальний ступінь передачі. У цьому

випадку, значення 0x00 показує, що ніякий особливий рівень обслуговування не встановлено.

4. Total Length (Загальна довжина): 1500 байт - це загальна довжина пакета (заголовок + дані) у байтах.
5. Identification (Ідентифікація): 0xb231 (45617) - це унікальний ідентифікатор фрагмента, який допомагає відновлювати фрагментовані пакети при їхній передачі через мережі.
6. Flags (Прапорці): 0x2, Don't fragment - це показник фрагментації пакета. У даному випадку, фрагментувати пакет не дозволено (встановлений прапорець "Don't fragment").
7. Fragment Offset (Зміщення фрагмента): 0 - це зміщення фрагмента у вихідному пакеті.
8. Time to Live (Час життя): 59 - це максимальна кількість маршрутизаторів, яку пакет може пройти перед видаленням або відхиленням. Кожен маршрутизатор, через який проходить пакет, зменшує це значення на одиницю.
9. Protocol (Протокол): TCP (6) - це номер протоколу, який використовується для транспортування даних в середині IP-пакета. У цьому випадку, використовується TCP.
10. Header Checksum (Контрольна сума заголовка): 0x6bb8 - це контрольна сума заголовка IP-пакета, яка використовується для виявлення помилок у передачі заголовка.
11. Source Address (IP-адреса джерела): 2.21.89.11 - це IP-адреса відправника пакета.
12. Destination Address (IP-адреса призначення): 192.168.0.106 - це IP-адреса отримувача пакета.

Загально, цей пакет представляє собою стандартну IP-датаграму з даними, які передаються через мережу з вказанням основних параметрів, таких як адреси джерела та призначення, протокол транспорту та рівень обслуговування.

3.6 Дослідження протоколу Ethernet

Протокол Ethernet є одним з фундаментальних стандартів для передачі даних в локальних комп'ютерних мережах (LAN). Він забезпечує ефективний зв'язок між різними пристроями, такими як комп'ютери, принтери, маршрутизатори та інші, у межах одного офісу або будинку.

Ethernet має багато різних версій, які відрізняються швидкістю передачі даних та типом кабелю, що використовується. Найпоширеніші з них включають 10BASE-T, 100BASE-TX, 1000BASE-T і 10GBASE-T. Кожен з цих стандартів визначає максимальну швидкість передачі даних та типи кабелю, що підтримуються. Наприклад, 10BASE-T підтримує швидкість передачі даних до 10 мегабіт на секунду і використовує виту пару кабелю, тоді як 1000BASE-T може передавати дані зі швидкістю до 1 гігабіт на секунду за допомогою витої пари кабелю.

Принцип роботи протоколу Ethernet базується на технології CSMA/CD (Carrier Sense Multiple Access with Collision Detection). Пристрої, які хочуть передати дані, спочатку перевіряють, чи вільний канал передачі, і, якщо так, вони починають передачу. Однак, якщо кілька пристроїв намагаються передати дані одночасно, може виникнути колізія. У такому випадку, пристрої виявляють колізію і припиняють передачу на короткий час перед повторною спробою.

Цей механізм дозволяє ефективно керувати доступом до мережі і зменшує ймовірність виникнення конфліктів. Крім того, протокол Ethernet використовує адреси MAC (Media Access Control) для ідентифікації пристроїв у мережі, що дозволяє ефективно маршрутизувати дані до відповідних отримувачів.

Поетапно проаналізуємо пакет протоколу Ethernet (рис. 3.8).

```
Ethernet II, Src: TpLinkTechno_94:e7:68 (b0:4e:26:94:e7:68),
> Destination: ASRockIncorp_91:54:38 (70:85:c2:91:54:38)
> Source: TpLinkTechno_94:e7:68 (b0:4e:26:94:e7:68)
  Type: IPv4 (0x0800)
```

Рисунок 3.8 – приклад пакету протоколу Ethernet.

1. Source Address (IP-адреса джерела): 2.21.89.11 - це IP-адреса відправника пакета.Адреса джерела (Source Address): TpLinkTechno_94:e7:68 (b0:4e:26:94:e7:68)
2. Destination Address (IP-адреса призначення): 3.45.87.21 - це IP-адреса отримувача пакета.Адреса призначення (Destination Address): ASRockIncorp_91:54:38 (70:85:c2:91:54:38)
3. Тип пакету: IPv4 (0x0800) - це показує, що дані в цьому пакеті відносяться до протоколу IPv4.

Загалом, Ethernet виступає як низькорівневий протокол у стеку TCP/IP, забезпечуючи фізичну та канальну інфраструктуру для передачі даних у мережі.

Розділ 4 Розрахунок корисно-пропускної здатності тракту мереж з комутацією пакетів

Для 100-мегабітного Fast Ethernet інтервал між кадрами складає 0,96 мікросекунди, а для Gigabit Ethernet він становить 10 раз менше - 96 наносекунд. Цей інтервал точно відповідає часу передачі 12 байт або 96 біт. Якщо визначити одиницею вимірювання часового інтервалу час, потрібний для передачі одного біта - бітовий інтервал (bt), то міжкадровий інтервал складає 96 біт. Цей метод визначення часових інтервалів не залежить від швидкості передачі даних і часто використовується в стандарті Ethernet.

Пауза після зіткнення є випадковою і обирається за наступною формулою:

$$\Delta t = L \cdot \tau, \quad (4.1)$$

де τ - інтервал затримки, рівний 512 біт, що при швидкості 1000 Мбіт/с складає 0,512 мікросекунд.

L - ціле випадкове число, обране з діапазону $[0; 2^N]$,

N - номер повторної спроби передачі даного кадру.

Таким чином, після десятої спроби передачі кадру випадкова пауза може приймати значення:

$$\Delta t = [0; 1024] \cdot 512 \text{ біт} = 524\,288 \text{ біт}.$$

Для стандарту Fast Ethernet це відповідає часовому діапазону від 0 до 0,524 мілісекунд.

Просте відношення між часом, необхідним для передачі кадру мінімальної довжини, та затримкою сигналу при поширенні в мережі:

$$T_{min} \geq 2\tau, \quad (4.2)$$

де τ - час поширення сигналу по мережі Ethernet.

Оскільки частка службової інформації завжди однакова, то чим менше загальний розмір кадру, тим вищі "накладні витрати". Службова інформація в кадрах Ethernet становить 18 байт (без преамбули).

Сам розмір кадру змінюється від:

46 + 18 = 64 байт до 1500 + 18 = 1518 байт.

Таким чином, для кадру мінімальної довжини корисна інформація складає 46/64 = 71,88% від загальної передаваної інформації, а для кадру максимальної довжини - 1500/1518 = 98,81% від загальної інформації.

Для передачі кадру мінімального розміру, який разом з преамбулою має довжину 128 байт або 1024 біт, і якщо врахувати міжкадровий інтервал в 96 біт, то період наступності кадрів складає 1120 біт.

При швидкості передачі в 1000 Мбіт/с це відповідає часу 1,12 мікросекунди. Тоді частота наступності кадрів, тобто кількість кадрів, що проходять через мережу за 1 секунду, складатиме:

1 / 1,12 мкс = 892857 кадрів/с.

Для передачі кадру максимального розміру, який разом з преамбулою має довжину 1582 байта або 12656 біт, період наступності складає: 12656 біт + 96 біт = 12752 біт.

А частота кадрів при швидкості передачі 1000 Мбіт/с складатиме:

1 / 12,752 мкс = 78419 кадрів/с.

Для кадру мінімальної довжини корисна пропускна здатність становить 46 байт/кадр · 892857 кадрів/с = 328,571 Мбіт/с.

Для кадру максимального розміру корисна пропускна здатність мережі становить 1500 байт/кадр · 78419 кадрів/с = 941,028 Мбіт/с.

Таким чином, в мережі Gigabit Ethernet корисна пропускна здатність може змінюватися залежно від розміру переданих кадрів від 54,76 до 975,6 Мбіт/с, а частота кадрів змінюється в діапазоні від 8127 до 148 809 кадрів/с.

ВИСНОВКИ

У даній роботі було проведено дослідження стеку протоколів TCP/IP, що є основою сучасних комп'ютерних мереж. Починаючи з огляду моделі TCP/IP з її складовими рівнями, включаючи прикладний, транспортний, мережевий та канальний, було розглянуто основні принципи та функції кожного рівня.

Важливим етапом роботи було дослідження загальних принципів аналізу протоколів за допомогою програмного додатку Wireshark. Завантаження та встановлення програми Wireshark, а також налаштування для захоплення трафіку, були проведені для подальшого аналізу роботи різних протоколів стеку TCP/IP.

В розділі, присвяченому дослідженню протоколів стеку TCP/IP, були досліджені такі протоколи, як FTP, SMTP, TCP, UDP, IP та Ethernet.

У завершальному розділі роботи був проведений розрахунок корисної пропускної здатності тракту мережі з комутацією пакетів, що дозволило зрозуміти, як ефективно використовувати мережеві ресурси для передачі даних.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Офіційний сайт Wireshark [Електронний ресурс] – Режим доступу: http://www.wireshark.org/
2. IANA (Internet Assigned Numbers Authority) [Електронний ресурс] – Режим доступу: https://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml
3. Н.Оліфер, В.Оліфер “ Комп'ютерні мережі. Принципи, технології, протоколи”.
4. Д. Дэвіс, Т. Лі. “Microsoft Windows Server 2003 Протоколи і служби TCP/IP Технічне керівництво”.
5. TCP/IP. Для професіоналів. 3-те видання / Т. Паркер, К. Сіян -СПб.: Пітер, 2004.
6. Персональні комп'ютери в мережах TCP/IP/Крейг Хант; перек. з англ. - BHV-Київ, 1997.
7. Високопродуктивні мережі. Енциклопедія користувача / Марк А. Спортак та ін; перек. з англ. - Київ, ДіаСофт, 1998.
8. Мережі ЕОМ: протоколи, стандарти, інтерфейси/Ю. Блек; перек. з англ. - М: Мір, 1990.
9. Таненбаум, А. С., Везерол, Д. Дж. [2011]. Комп'ютерні мережі. Видавництво : Prentice Hall.
10. Комер, Д. Е. [2014]. Комп'ютерні мережі та Інтернет. Видавництво : Chapman and Hall.

Додаток А

CHAPTER 1
Overview of the TCP/IP Protocol Stack Structure

1.1 The TCP/IP Model

The TCP/IP model, which stands for Transmission Control Protocol/Internet Protocol, is a conceptual model that describes how computers and various devices interact and exchange data in the large global network, the Internet. This model consists of four main layers (Table 1.1): the application layer, the transport layer, the internet layer, and the network access layer. Each of these layers performs its unique functions and operations, ensuring efficient data transmission across the network, allowing computers to communicate with each other.

Table 1.1 – The TCP/IP Model

OSI Model		TCP/IP Model	
Application	7	4	Application
Presentation	6		
Session	5		
Transport	4	3	Transport
Network	3	2	Network
Data Link	2	1	Data Link
Physical	1		

1.2 Application Layer

At the application layer of the TCP/IP stack, communication occurs between various applications running on different devices within the network. This layer includes a wide range of protocols that enable applications to exchange data effectively.

The main application layer protocols include HTTP, HTTPS, FTP, SMTP, POP3, IMAP, DNS, and others.

The primary purpose of the application layer is to provide an interface for interaction between various applications and services operating within the network. For example, HTTP is used for transferring web pages, FTP for transferring files, SMTP for sending email, and DNS for converting domain names into IP addresses.

Applications operating at the application layer utilize the services of the transport layer, such as TCP or UDP, to transmit data across the network. These protocols are used to establish connections between different devices, manage data flow, and ensure correct data delivery.

1.3 Transport Layer

The transport layer of the TCP/IP stack is responsible for data transmission between two endpoints within the network. The primary protocols used at this layer are TCP (Transmission Control Protocol) and UDP (User Datagram Protocol). TCP provides reliable data transmission, making it ideal for applications where data accuracy and integrity are critical, such as file transfers or web pages. On the other hand, UDP is often used for fast data transmission in real-time applications, such as video or voice calls, where some data loss is acceptable but minimal delay is crucial to ensure a smooth and uninterrupted flow of information.

1.4 Internet Layer

The main function of the internet layer is to handle the process of routing. It is responsible for determining the shortest and most efficient path for data transmission from the sender to the final recipient. This includes identifying the optimal route for transmission, fragmenting data into individual packets that can be transmitted through various network nodes, and forwarding these packets through the network to their final destination.

Additionally, the internet layer is responsible for node identification and addressing within the network. Each network node has a unique IP address, used to identify it within the network. These addresses are necessary for routing data packets from the sender to the recipient.

Popular internet layer protocols include the Internet Protocol (IP), Internet Control Message Protocol (ICMP) for exchanging error and network status messages, and Address Resolution Protocol (ARP) for resolving MAC addresses to IP addresses.

1.5 Network Access Layer

The network access layer in the TCP/IP model is responsible for data transmission over the physical network and physical connections between devices. This layer performs functions necessary for direct bit transmission through the network connection without any processing or data transformation.

The main tasks of the network access layer include managing physical connections, transmitting data over network cables or wireless connections, and resolving issues that may arise during data transmission, such as bit errors.

Let's look at the main functions of the channel level.

1. Framing: Data received from higher layers is divided into frames for transmission over the network medium. A frame contains addressing and data flow control information.

2. Network Access Control: The network access layer can use various protocols to manage network access, such as CSMA/CD (Carrier Sense Multiple Access with Collision Detection) or CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance).

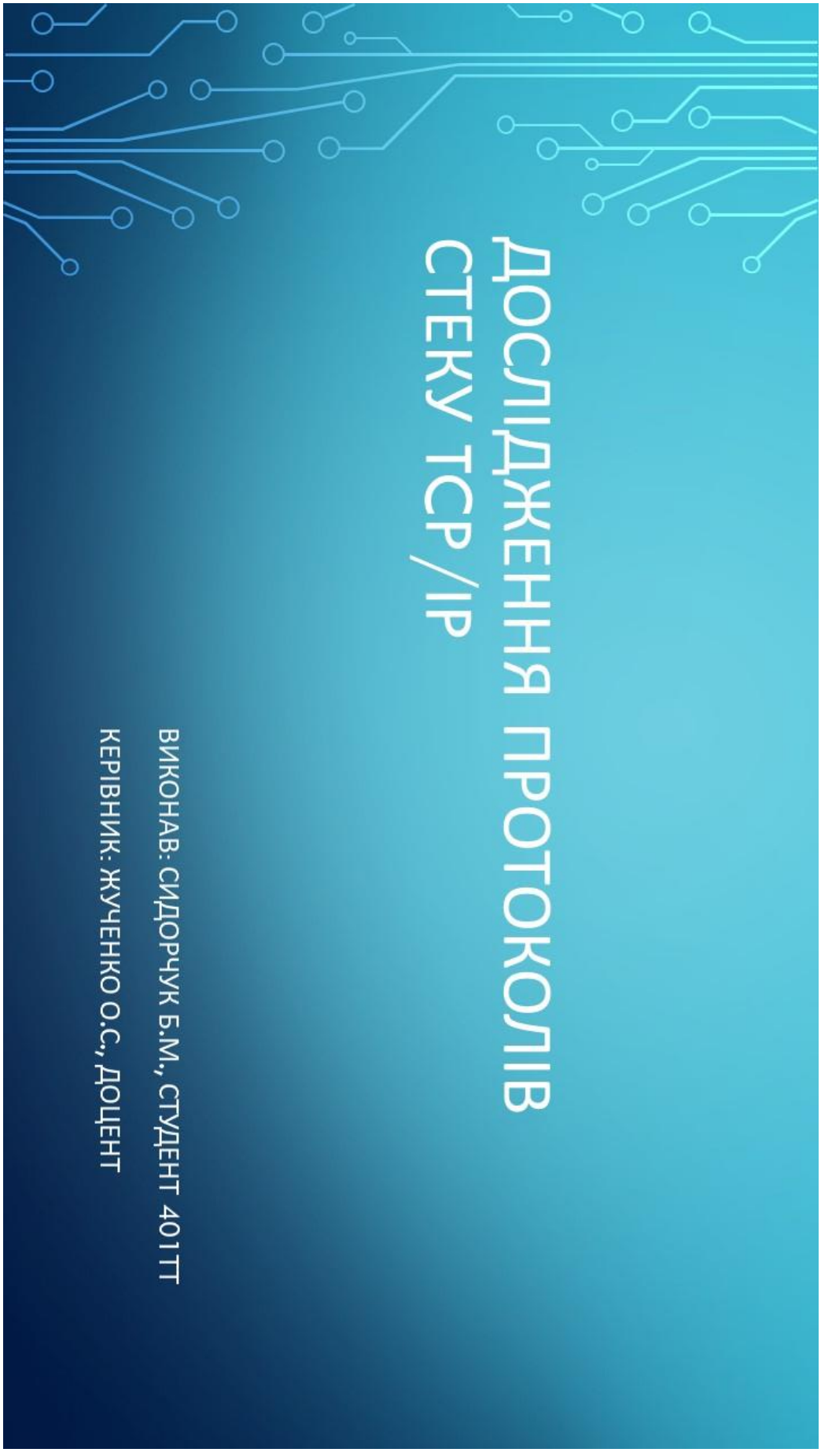
3. Error Detection and Correction: The network access layer provides mechanisms for detecting and correcting errors that may occur during data transmission. This can include using CRC (Cyclic Redundancy Check) for data integrity verification.

4. Physical Encoding: Data is converted into signals that can be transmitted through the physical medium. This includes operations like modulation, demodulation, encoding, and decoding.

1.6 Advantages of TCP/IP

1. Reliable Data Transmission: TCP (Transmission Control Protocol) ensures reliable data transmission through error checking, retransmissions, and connection recovery in case of packet loss.
2. Open Standard: TCP/IP is an open standard, meaning that it can be used by various software developers and hardware manufacturers to create compatible products.
3. Scalability: TCP/IP easily scales from small local networks to large global internet networks, making it ideal for any communication needs.
4. Flexibility: TCP/IP supports various network topologies and communication technologies, including wireless networks, Ethernet, DSL, and others.
5. High Data Transmission Speed: TCP/IP can be optimized for high-speed data transmission, making it an excellent choice for modern network needs.
6. Openness for Interface Development: TCP/IP provides developers with the opportunity to create their own protocols and interfaces to implement various network functions.

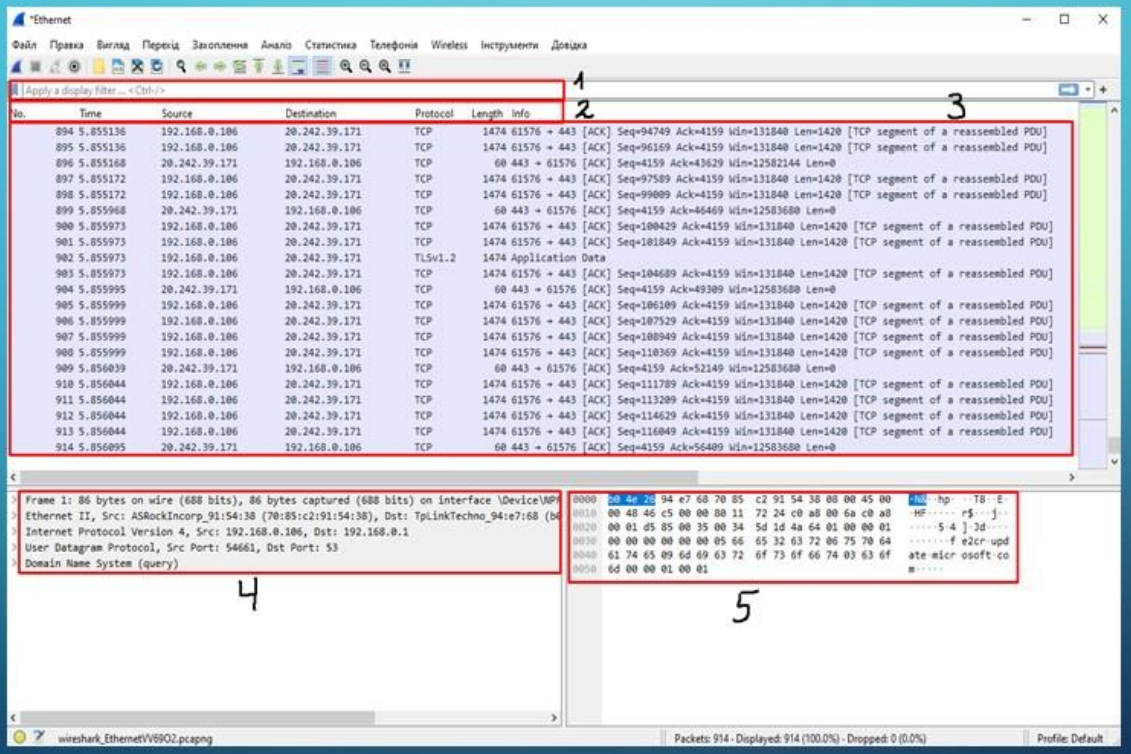
Додаток Б


ДОСЛІДЖЕННЯ ПРОТОКОЛІВ СТЕКУ TCP/IP
ВИКОНАВ: СИДОРЧУК Б.М., СТУДЕНТ 401ТТ
КЕРІВНИК: ЖУЧЕНКО О.С., ДОЦЕНТ

- Актуальність даної роботи обумовлена необхідністю глибокого аналізу сучасного стану стеку протоколів TCP/IP, оскільки він основою сучасних комп'ютерних мереж , та виявлення його слабких місць і потенційних загроз.
- Мета роботи: дослідження та аналіз принципу роботи протоколів стеку TCP/IP.
- Методи: Аналіз існуючої наукової та технічної літератури, проведення тестувань з використанням спеціалізованого програмного забезпечення.

Модель TCP/IP	
4	Прикладний(Application Layer)
3	Транспортний(Transport Layer)
2	Мережевий(Network Layer)
1	Канальний(Link Layer)

ПРОТОКОЛ-АНАЛІЗАТОР WIRESHARK



ПРОТОКОЛ FTP

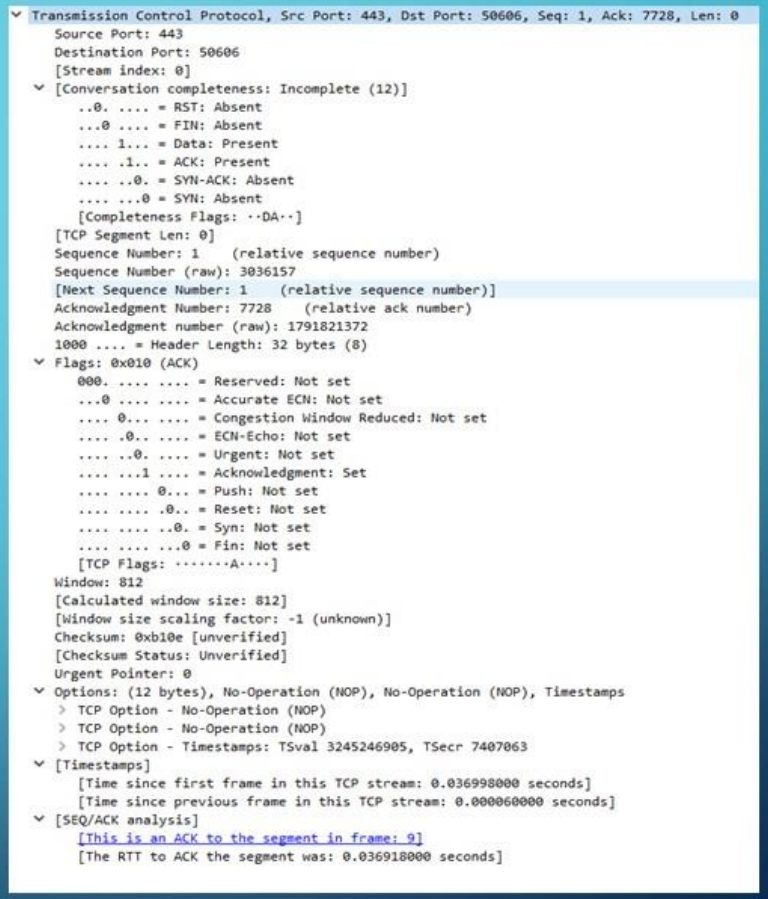
```
220-
220 6bone.informatik.uni-leipzig.de FTP server (NetBSD-ftpd 20041119) ready.
USER anonymous
331 Guest login ok, type your name as password.
PASS IEUser@
230 Guest login ok, access restrictions apply.
opts utf8 on
502 Unknown command 'utf8'.
syst
215 UNIX Type: L8 Version: NetBSD-ftpd 20041119
site help
214-
```

ПРОТОКОЛ SMTP

```
220-xc90.websitewelcome.com ESMTP Exim 4.69 #1 Mon, 05 Oct 2009 01:05:54 -0500
220-We do not authorize the use of this system to transport unsolicited,
220 and/or bulk e-mail.
EHLO GP
250-xc90.websitewelcome.com Hello GP [122.162.143.157]
250-SIZE 52428800
250-PIPELINING
250-AUTH PLAIN LOGIN
250-STARTTLS
250 HELP
AUTH LOGIN
334 VXNlcm5hbWU6
Z3VycGFydGFwQHBhdHJpb3RzLmlu
334 UGFzc3dvcmQ6
cHVuamFiQDEyMw==
235 Authentication succeeded
MAIL FROM: <gurpartap@patriots.in>
250 OK
RCPT TO: <raj_deol2002in@yahoo.co.in>
250 Accepted
DATA
354 Enter message, ending with "." on a line by itself
From: "Gurpartap Singh" <gurpartap@patriots.in>
To: <raj_deol2002in@yahoo.co.in>
Subject: SMTP
Date: Mon, 5 Oct 2009 11:36:07 +0530
Message-ID: <000301ca4581$ef9e57f0$cedb07d0$@in>
MIME-Version: 1.0
Content-Type: multipart/mixed;
        boundary="----=_NextPart_000_0004_01CA4580.095693F0"
X-Mailer: Microsoft Office Outlook 12.0
Thread-Index: AcpFgem9BvjjZEDeR1Kh8i+hUyVo0A==
Content-Language: en-us
x-cr-hashedpuzzle: SeA= AAR2 ADaH BpiO C4G1 D1gW FNB1 FPkR Fn+W HFCP HnYJ JO7s K
osha1_v1;7;{CAA37F59-1850-45C7-8540-AA27696B5398};ZwB1AHIAcABhAHIAdABhAHAAQABwAG
x-cr-puzzleid: {CAA37F59-1850-45C7-8540-AA27696B5398}

This is a multipart message in MIME format.
```

ПРОТОКОЛ TCP



ПРОТОКОЛ UDP

```
User Datagram Protocol, Src Port: 37304, Dst Port: 60379
    Source Port: 37304
    Destination Port: 60379
    Length: 344
    Checksum: 0x648f [unverified]
    [Checksum Status: Unverified]
    [Stream index: 6]
  > [Timestamps]
    UDP payload (336 bytes)
```

ПРОТОКОЛ IP

```
Internet Protocol Version 4, Src: 2.21.89.11, Dst: 192.168.0.106
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
v Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    0000 00.. = Differentiated Services Codepoint: Default (0)
    .... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
  Total Length: 1500
  Identification: 0xb231 (45617)
v 010. .... = Flags: 0x2, Don't fragment
    0... .... = Reserved bit: Not set
    .1.. .... = Don't fragment: Set
    ..0. .... = More fragments: Not set
  ...0 0000 0000 0000 = Fragment Offset: 0
  Time to Live: 59
  Protocol: TCP (6)
  Header Checksum: 0x6bb8 [validation disabled]
  [Header checksum status: Unverified]
  Source Address: 2.21.89.11
  Destination Address: 192.168.0.106
```

ПРОТОКОЛ ETHERNET

```
Ethernet II, Src: TpLinkTechno_94:e7:68 (b0:4e:26:94:e7:68),
> Destination: ASRockIncorp_91:54:38 (70:85:c2:91:54:38)
> Source: TpLinkTechno_94:e7:68 (b0:4e:26:94:e7:68)
  Type: IPv4 (0x0800)
```

ВИСНОВКИ

Використання TCP/IP є важливим для забезпечення функціональності сучасного Інтернету та корпоративних мереж. TCP/IP відіграє важливу роль у забезпеченні зв'язку в мережах, надаючи основні принципи та структуру для передачі даних усіма комп'ютерним пристроям у світі.