МІНІСТЕРСТВО ВНУТРІШНІХ СПРАВ УКРАЇНИ
ЛЬВІВСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ ВНУТРІШНІХ СПРАВ

НАВЧАЛЬНО–НАУКОВИЙ ІНСТИТУТ УПРАВЛІННЯ, ПСИХОЛОГІЇ ТА БЕЗПЕКИ

Кафедра інформаційних технологій

ІНФОРМАЦІЙНА СИСТЕМА МОНІТОРИНГУ ТА АНАЛІЗУ МЕРЕЖЕВОГО ТРАФІКУ

Кваліфікаційна робота
здобувача вищої освіти
4 курсу заочної форми навчання
Віталія Данилюка

Науковий керівник:
доцент, кандидат технічних наук
Любомир ФЛУД

Рецензент:

_____
вчене звання, науковий ступінь

_____
(Ім’я ПРІЗВИЩЕ рецензента)

Кваліфікаційна робота допущена до захисту
«___» _____ 2026 р., протокол № ____

Завідувач кафедри інформаційних технологій
_____ Олег ЗАЧЕК
(підпис)

Львів
2026

АНОТАЦІЯ

Данилюк В. Інформаційна система моніторингу та аналізу мережевого трафіку. – Рукопис.

Дослідження на здобуття освітнього ступеня «бакалавр» за спеціальністю 126 «Інформаційні системи та технології». – Львівський державний університет внутрішніх справ, МВС України, Львів, 2026.

Робота присвячена розробці програмного забезпечення для моніторингу та аналізу мережевого трафіку в локальних комп'ютерних мережах. У ході дослідження розроблено додаток, який дозволяє здійснювати захоплення пакетів у реальному часі, їх фільтрацію, статистичний аналіз, візуалізацію навантаження та прогнозування за допомогою алгоритму EWMA. Графічний інтерфейс забезпечує гнучке налаштування фільтрів та параметрів відображення. Передбачено збереження даних у файл, побудову графіків.

Ключові слова: мережевий трафік, аналіз трафіку, пакетний сніфер, Npcap, SharpPcap, моніторинг мережі, EWMA, інформаційна безпека.

ABSTRACT

Danyliuk V. Information System for Network Traffic Monitoring and Analysis. – Manuscript.

Bachelor's qualification thesis submitted in partial fulfillment of the requirements for the degree of Bachelor in the specialty 126 «Information Systems and Technologies». – Lviv State University of Internal Affairs, Ministry of Internal Affairs of Ukraine, Lviv, 2026.

The thesis is devoted to the development of software for monitoring and analyzing network traffic in local computer networks. As a result of the research, an application has been developed that allows real-time packet capturing, filtering, statistical analysis, traffic load visualization, and forecasting using the Exponential Weighted Moving Average (EWMA) algorithm. The graphical user interface provides flexible filter configuration and data display settings. The system also supports saving captured data to files and generating graphs.

Keywords: network traffic, traffic analysis, packet sniffer, Npcap, SharpPcap, network monitoring, EWMA, information security.

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

API	Application Programming Interface
BPF	Berkeley Packet Filter
IP	Internet Protocol
ISDN	Integrated Services Digital Network
LAN	Local Area Network
EWMA	Exponential Weighted Exponential Weighted Moving Average (EWMA) ()
NDIS	Network Driver Interface Specification
NIC	Network Interface Controller
PCAP	Packet Capture

ЗМІСТ

ВСТУП

В умовах стрімкого розвитку інформаційно-комунікаційних технологій комп’ютерні мережі відіграють ключову роль в забезпеченні ефективного функціонування інформаційних систем, бізнес-процесів та державного управління. Зростання обсягів даних, що передаються мережами, збільшення складності мережевої інфраструктури та постійна загроза кібербезпеці зумовлюють необхідність постійного моніторингу та аналізу мережевого трафіку. Володіння інструментами, які дозволяють у реальному часі контролювати стан мережі, виявляти аномалії, оцінювати навантаження та прогнозувати його зростання, стає важливою складовою забезпечення стабільності, продуктивності та безпеки інформаційних систем.

Значну допомогу в управлінні комп’ютерними мережами може надати спеціалізоване програмне забезпечення для моніторингу та аналізу мережевого трафіку. Такий застосунок здатен стати зручним і потужним інструментом для адміністраторів мереж, розробників мережевих додатків та фахівців з інформаційної безпеки. Необхідність застосування таких систем обумовлюється глобальною цифровізацією суспільства, зростанням кількості підключених пристроїв, поширенням хмарних технологій та віддаленої роботи. Подібні інструменти можуть використовуватися як в корпоративному середовищі, так і в освітніх закладах, державних установах та невеликих локальних мережах.

Розроблений програмний продукт поєднує в собі функції захоплення пакетів у реальному часі, їх аналізу, візуалізації навантаження, застосування фільтрів, побудови графіків та збереження даних. Він призначений для ефективного контролю локальної комп’ютерної мережі, оперативного виявлення проблем та прогнозування мережевого навантаження.

Актуальність роботи – зростаюча потреба в сучасних, гнучких і доступних засобах моніторингу та аналізу мережевого трафіку, які

поєднують високу функціональність з простотою використання.

Аналіз останніх досліджень і публікацій. Питанням розробки систем аналізу мережевого трафіку присвячено значну кількість наукових праць як вітчизняних, так і зарубіжних дослідників. Серед них варто відзначити роботи, присвячені технологіям захоплення пакетів (Npcap), інструментам аналізу та алгоритмам обробки даних [1–9]. Разом з тим, багато існуючих рішень є або надто складними та ресурсоємними, або комерційними з обмеженою функціональністю. Тому подальше удосконалення програмних засобів моніторингу локальних мереж з акцентом на зручний інтерфейс, ефективність та інтеграцію сучасних технологій (C#, WPF) залишається актуальним напрямом.

Метою кваліфікаційної роботи є проєктування та розробка інформаційної системи моніторингу та аналізу мережевого трафіку.

Для досягнення поставленої мети необхідно вирішити такі **завдання**:

- провести аналіз предметної області моніторингу мережевого трафіку в локальних мережах та огляд сучасних інструментів;
- обґрунтувати вибір методів та технологій захоплення пакетів (C# + WPF + Npcap + SharpPcap);
- спроєктувати архітектуру програмного комплексу з урахуванням вимог продуктивності, модульності та безпеки;
- реалізувати модулі захоплення, фільтрації, візуалізації та аналізу трафіку;
- забезпечити можливість збереження даних, побудови графіків та прогнозування навантаження;
- розробити зручний графічний інтерфейс та передбачити можливість роботи в режимі командного рядка;
- провести аналіз ризиків безпеки та запропонувати технічні та організаційні заходи захисту.

Об’єктом дослідження є мережева інфраструктура локальних комп’ютерних мереж та процеси передавання, оброблення та контролю

мережевого трафіку.

Предметом дослідження виступають моделі, методи та технології збору, оброблення, аналізу даних мережевого трафіку у локальній комп’ютерній мережі.

Методи дослідження – бібліометричний метод дозволив проаналізувати наявні наукові праці у даній сфері діяльності, методи системного аналізу, проектування та програмної реалізації. У роботі застосовано інструментарій мови програмування C#, технологію Windows Presentation Foundation (WPF) та бібліотеку libpcap (Npcap).

Програмне забезпечення реалізовано з урахуванням принципів модульності та розширюваності, що дозволяє в майбутньому додавати нові функції аналізу без суттєвої зміни базового коду. Продуктивність забезпечується ефективною обробкою пакетів на рівні ядра та оптимізованою візуалізацією даних.

Структура роботи. Кваліфікаційна робота складається із вступу, трьох розділів, висновків, списку використаних джерел. Обсяг основного тексту роботи складає 40 сторінок, 14 рисунків, 2 таблиці, 20 бібліографічних джерел. Загальний обсяг роботи – 49 сторінок.

РОЗДІЛ 1
АНАЛІТИЧНИЙ ОГЛЯД МЕРЕЖЕВОГО ТРАФІКУ

1.1. Поняття та огляд існуючих рішень

Мережевий трафік – це сукупність цифрових даних, які передаються через комп’ютерні мережі. Він включає всі види цифрової інформації, що переміщуються між пристроями: текстові повідомлення, мультимедійні файли, пакети команд, запити до серверів тощо. Аналіз мережевого трафіку є важливим для забезпечення ефективної роботи мережі, оптимізації пропускної здатності, кібербезпеки та діагностики проблем.

Організовуючи системи контролю мережевих ресурсів потрібно вибирати відповідне програмне забезпечення. На сучасному ринку існує велика кількість інструментів для збору та аналізу статистики трафіку. Але, більшість із цих інструментів орієнтована на операційні системи сімейства UNIX. Більшість програм для операційної системи Windows часто мають дуже високу вартість та обмеженими можливостями адаптації до потреб конкретного користувача.

Існуючі системи моніторингу мережевої активності переважно використовують наступні технології: перехоплення пакетів даних, аналіз інформації, отриманої від зовнішніх програмних, а також обробка мережевих потоків за допомогою спеціальних драйверів (рис. 1.1). Кожен із цих підходів має особливості, переваги та обмеження.

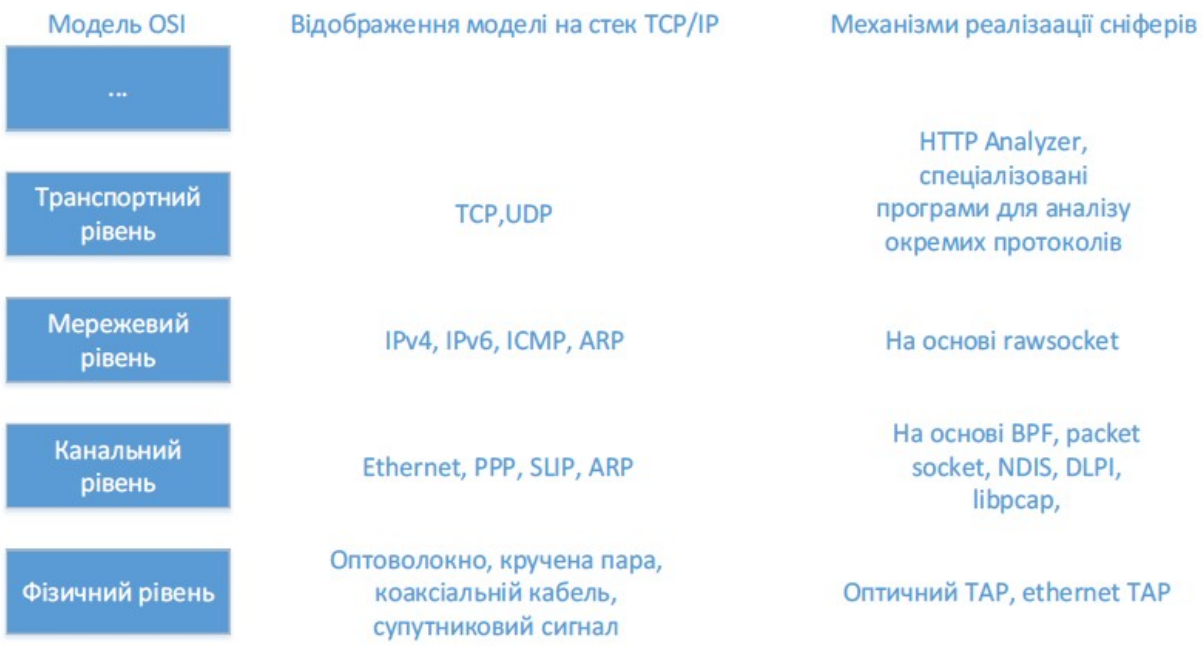

Рис.1.1. Реалізація сніферів відносно моделі OSI [1]

Метод перехоплення пакетів ґрунтується на переведенні мережевого адаптера в режим прийому всього трафіку, який проходить через мережу. У звичайному режимі пристрій обробляє лише пакети, адресовані безпосередньо йому, але у режимі прослуховування можливо отримання повної інформації про мережевий обмін. Основою таких додатків є мережеві драйвери та бібліотеки захоплення пакетів. Недоліком підходу є значне навантаження на комп'ютер під час обміну даними, що може призвести до втрати частини пакетів. Цей метод забезпечує лише відстеження об'єму трафіку, але не має можмивості для його обмеження чи керування [1].

Другий підхід використовує дані отриманих від стороннього програмного забезпечення або мережевого обладнання. Перевагою другого підхіду є збереження початкової інформації у незмінному вигляді, що дозволяє повторно проводити аналіз із використанням інших критеріїв або фільтрів. Також у деяких випадках, наприклад при використанні проксі-серверів, облік може охоплювати не весь фактичний обсяг мережевого трафіку [1].

Третій підхід реалізується на рівні драйвера, який здійснює обробку

мережевих потоків в операційній системі. Додатки засновані на драйвері мають високу точністю вимірювань та продуктивність. Але за умов значного навантаження можливе пропускання окремих пакетів. Варто пам’ятати, що драйвер працює в режимі ядра і будь-які помилки в його роботі можуть негативно вплинути на стабільність усієї системи.

У бакалаврській роботі будемо використовувати підхід, що базується на перехопленні мережевих пакетів через переведення мережевого інтерфейсу в режим прослуховування. Передавання інформації до користувацького застосунку здійснюється за допомогою архітектури Npcap. Використання цієї програми забезпечить незалежність від конкретного мережевого обладнання та сумісність із операційними системами сімейства Microsoft Windows.

1.2. Загальний огляд сучасних інструментів для аналізу трафіку

Аналіз мережевого трафіку є важливим завданням у сфері інформаційної безпеки, адміністрування мереж та оптимізації роботи інформаційних систем. Завдяки сучасним інструментам можна контролювати потоки даних, виявляти загрози, знаходити вузькі місця в інфраструктурі та покращувати продуктивність мережі.

Варто зазначити, що існують ключові сучасні інструменти для аналізу трафіку, які використовуються для різних цілей – від базового моніторингу до детекції аномалій та кіберзагроз.

Сучасні інструменти для аналізу мережевого трафіку можна розподілити на кілька категорій залежно від їх функціоналу та сфери застосування [2]:

– Моніторингові системи – використовуються для постійного відстеження трафіку та візуалізації даних у реальному часі.

– Системи виявлення та запобігання вторгненням IDS/IP – аналізують трафік для пошуку підозрілих активностей.

– Пакетні аналізатори Sniffers – захоплюють і детально аналізують

мережеві пакети.

– Системи аналізу поведінки мережевого трафіку NBA – Network Behavior Analysis – виявляють аномалії та потенційні загрози.

– Програми для діагностики та тестування мережі – використовуються для перевірки пропускної здатності, затримок, якості з'єднання.

1.2.1. Microsoft Defender for Endpoint (MDE)

Мережевий аналізатор та система захисту кінцевих точок Microsoft Defender for Endpoint (MDE) є сучасним enterprise-рішенням від Microsoft, яке значно виходить за рамки класичного моніторингу трафіку, поєднуючи функції Next-Generation Antivirus (NGAV)**,** Endpoint Detection and Response (EDR), управління поверхнею атаки, автоматизоване розслідування та реагування на інциденти.

Продукт створено компанією Microsoft як частина екосистеми Microsoft Defender XDR. Він призначений для захисту кінцевих пристроїв (Windows, macOS, Linux, Android, iOS, IoT) від складних загроз, включаючи ransomware, APT, zero-day атаки тощо.

MDE дозволяє здійснювати безперервний моніторинг поведінки пристроїв у реальному часі, збирати телеметрію (включаючи мережеву активність), виявляти підозрілу активність, перехоплювати та аналізувати мережеві з'єднання, а також автоматично реагувати на інциденти. Зібрані дані зберігаються в хмарі Microsoft для подальшого глибокого аналізу, hunting'у та forensic'у. Накопичені дані можуть використовуватися для виправлення помилок у локальних і розподілених мережах, а також взаєсодіють з більшістю мережевого обладнання які використовують протокол TCP/IP та інші сучасні мережеві технології.

Перевагами Defender for Endpoint є:

Комплексний захист і видимість**:** MDE поєднує традиційний моніторинг мережевого трафіку з розширеними можливостями EDR. Він збирає сигнали поведінки безпосередньо з операційної системи, аналізує

мережеві з’єднання, процеси, зміни файлів, реєстру тощо. Це дозволяє виявляти складні атаки, які не помітні класичним сніферам.

Network Protection та аналіз трафіку**:** Вбудована функція Network Protection блокує з’єднання з шкідливими доменами IP (C2-серверами, фішинговими сайтами). Підтримуються web content filtering, custom indicators, а також пасивний/активний device discovery для виявлення unmanaged пристроїв у мережі шляхом аналізу трафіку.

Автоматизоване розслідування та реагування (Automated Investigation and Response – AIR)**:** Система самостійно аналізує alerts, будує attack timeline, виконує remediation actions (ізоляція пристрою, блокування процесів тощо). Це значно прискорює response порівняно з ручним аналізом у традиційних Network Monitor.

Advanced Hunting та Threat Intelligence**:** Доступ до потужної мови запиту Kusto (KQL) для проактивного пошуку загроз. Використовується глобальна threat intelligence Microsoft.

Управління поверхнею атаки (Attack Surface Reduction – ASR)**:** Правила для зменшення векторів атаки, device control (USB), firewall, vulnerability management.

Інтеграція та централізоване управління**:** Повна інтеграція з Microsoft Intune, Sentinel, Defender XDR, Azure тощо. Єдиний портал security.microsoft.com для управління всіма пристроями.

Підтримка крос-платформенності**:** Працює на Windows, macOS, Linux, мобільних пристроях та IoT.

Приклад графічного інтерфейсу Microsoft Defender for Endpoint зображено на рис. 1.2:

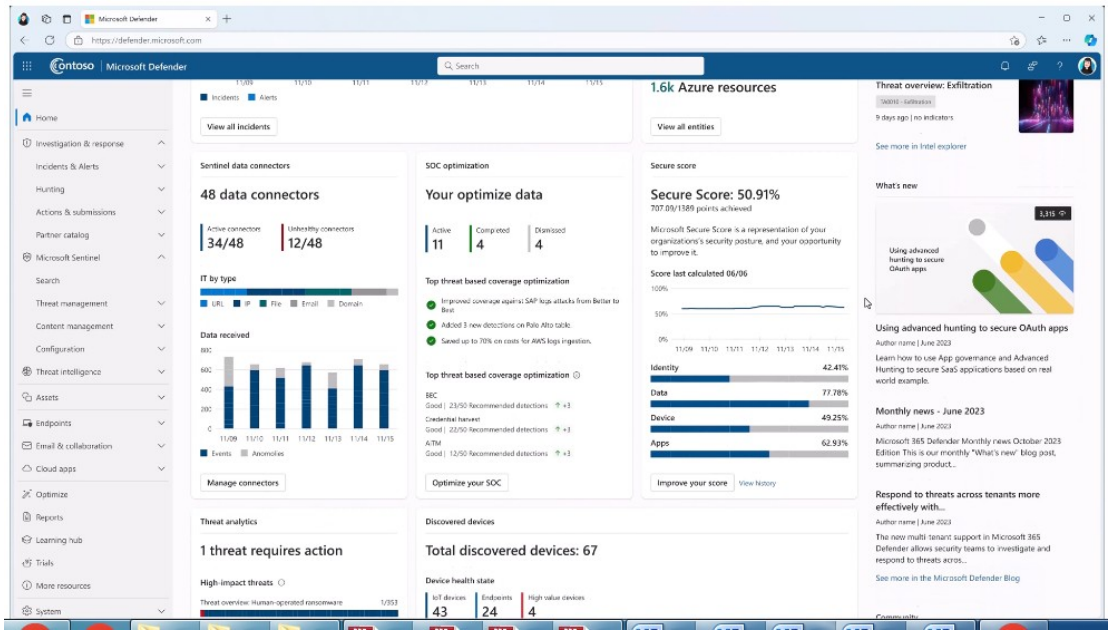

Рис.1.2. Приклад програми Microsoft Defender

Розглянутий продукт має наступні недоліки:

Хмарно-орієнтований: для повної функціональності потрібне підключення до хмари Microsoft; в офлайн-режимі можливості суттєво обмежені.

Складність розгортання та налаштування**:** вимагає onboarding пристроїв, правильної конфігурації політики, розуміння EDR-концепцій. Не такий «plug-and-play», як старий Network Monitor.

Високе навантаження на ресурси**:** на слабких пристроях сенсори та постійний моніторинг можуть впливати на продуктивність.

Ліцензування**:** повна функціональність (Plan 2) є платною; Plan 1 – базовий захист. Не підходить для дуже маленьких організацій без бюджету.

Microsoft Defender for Endpoint є еволюційним наступником простих інструментів на кшталт Network Monitor.

1.2.2. SolarWinds NetFlow Traffic Analyzer / NPM

SolarWinds NetFlow Traffic Analyzer (NTA) є додатковим модулем до основного продукту SolarWinds Network Performance Monitor (NPM). Разом вони утворюють потужний комплекс для моніторингу продуктивності мережі

та детального аналізу трафіку. NPM відповідає за загальний моніторинг стану мережевих пристроїв, а NTA спеціалізується на аналізі потоків даних за допомогою технологій NetFlow, sFlow, IPFIX та J-Flow.

Даний програмний комплекс працює під управлінням операційних систем сімейства Windows Server та підтримує розгортання як у on-premise середовищі, так і в гібридних інфраструктурах.

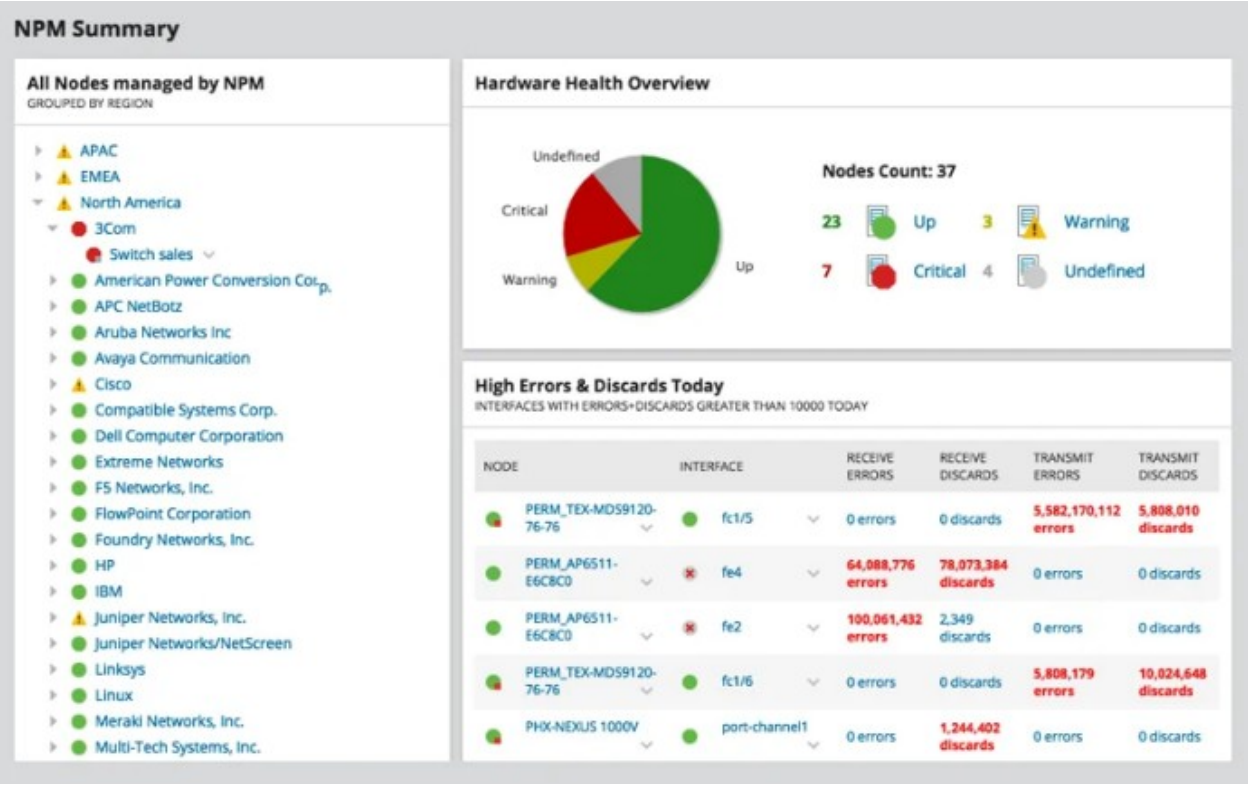

Рис. 1.3. Приклад інтерфейсу SolarWinds NPM з модулем NTA

Інтерфейс SolarWinds NPM та NTA є сучасним, графічно насиченим і орієнтованим на зручність щоденного використання. Він включає інтерактивні дашборди, динамічні графіки, теплові карти та можливість глибокої деталізації. Серед ключових функцій варто виділити:

- аналіз трафіку в реальному часі та в історичній перспективі;
- виявлення топ-споживачів каналу (Top Talkers / Top Listeners);
- класифікацію трафіку за додатками, протоколами, користувачами та IP-адресами;
- моніторинг якості обслуговування (QoS);
- побудову детальних звітів та гістограм;

- візуалізацію шляхів проходження пакетів (NetPath);
- систему інтелектуальних сповіщень та аномалій.

Однією з найсильніших сторін продукту є глибока інтеграція даних SNMP і NetFlow, що дозволяє отримувати комплексну картину стану мережі – від завантаження інтерфейсів до конкретних додатків, які споживають найбільше ресурсів. Також підтримується створення кастомних звітів і тривимірний аналіз тенденцій навантаження.

До переваг SolarWinds NTA / NPM можна віднести потужну систему візуалізації, зручність масштабування, велику кількість готових шаблонів для різних виробників обладнання та можливість довгострокового зберігання даних для ретроспективного аналізу.

До недоліків програми відноситься:

- висока вартість ліцензій, особливо при великій кількості моніторинг-елементів;
- обов’язкова наявність модуля NPM для роботи NTA;
- значні апаратні вимоги до сервера при обробці великих обсягів потокових даних;
- складність первинного налаштування та розгортання для невеликих організацій;
- залежність від Windows-середовища (відсутність нативної підтримки Linux).

SolarWinds NetFlow Traffic Analyzer / NPM є одним з найпоширеніших комерційних рішень для професійного моніторингу мереж середнього та великого масштабу станом на 2026 рік [1-5].

1.2.3. CommView

CommView це спеціалізований програмний засіб для аналізу мережевого трафіку, який розповсюджується за комерційною основою. Для ознайомлення з можливостями продуктурозробником надається пробна версія, функціональність якої обмежена. Програма призначена для контролю

мережевої активності як у локальних мережах та Інтернеті.

CommView перехоплює пакети даних, що проходять через мережевий адаптер комп'ютера, їх подальший аналіз і відображення результатів у зрозумілій для користувача формі. Програма орієнтована на використання в середовищі операційних систем Windows та підтримує багато мережевих протоколів, тому користувач отримує детальну інформацію про структуру, вміст і параметри перехоплених пакетів.

Для вибіркового аналізу трафіку в CommView є механізм фільтрації даних. Користувач може створювати правила для відбору пакетів за заданими критеріями під час їх захоплення,але тут неможливо застосувати нові фільтри до вже збережених результатів моніторингу (рис. 1.4).

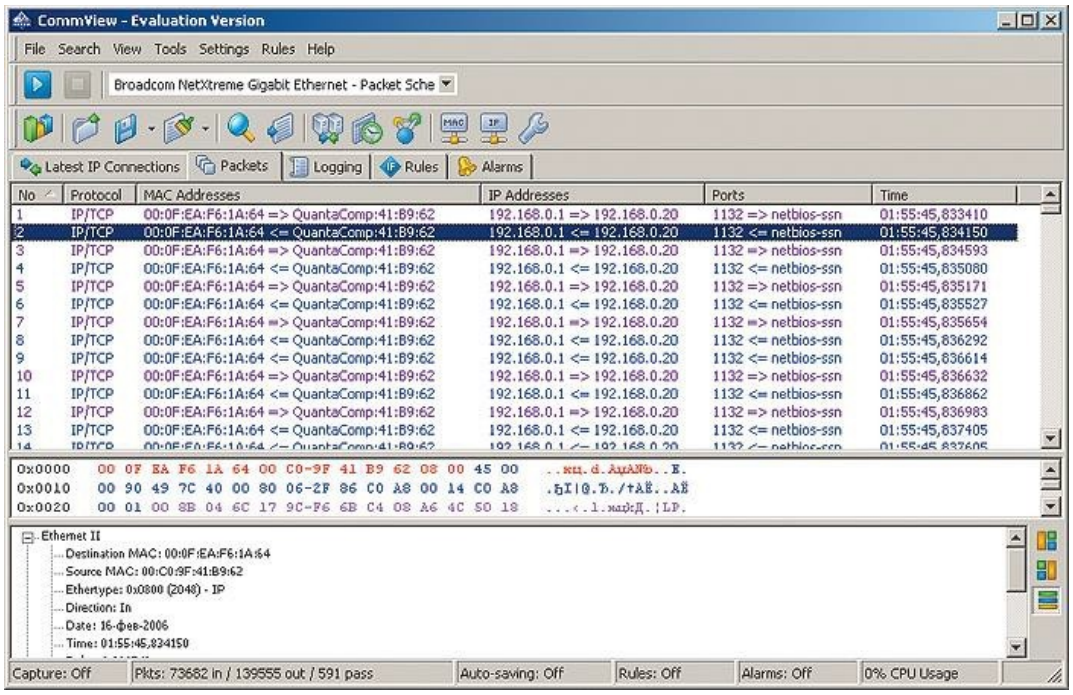

Рис.1.4. Приклад програми CommView

Попри значні функціональні можливості, пакетний аналізатор CommView має і недоліки такі як: висока вартість ліцензії, відсутність підтримки окремих декодерів протоколів Windows, зокрема Kerberos та RDP, потрібно використовувати різні версії програми для роботи з дротовими та бездротовими мережами [5].

1.2.4. PRTG Network Monitor

PRTG Network Monitor є одним з найпопулярніших комерційних рішень для комплексного моніторингу інфраструктури та мережевого трафіку. Програмне забезпечення розроблено німецькою компанією Paessler і активно використовується як у невеликих компаніях, так і в середніх та великих підприємствах.

PRTG поширюється виключно на комерційній основі. Для ознайомлення доступна повнофункціональна 30-денна пробна версія, після закінчення якої функціональність суттєво обмежується. Даний продукт призначений для постійного моніторингу локальних, розподілених та гібридних мереж. Він здатний збирати дані про трафік за допомогою протоколів NetFlow, sFlow, IPFIX, SNMP, а також виконувати активний моніторинг доступності пристроїв, пропускної здатності каналів і якості сервісів.

PRTG підтримує операційні системи сімейства Windows Server. Відмінною особливістю продукту є сенсорний підхід до моніторингу — кожен параметр, який потрібно контролювати, активується як окремий сенсор. Завдяки цьому система є дуже гнучкою та дозволяє точно налаштовувати обсяг збираних даних.

Інтерфейс програми є веб-орієнтованим і доступний через будь-який сучасний браузер. Головне вікно представляє собою інтерактивний дашборд з великою кількістю віджетів, графіків та карт. Користувач може створювати власні дашборди, налаштовувати сповіщення та генерувати детальні звіти. PRTG підтримує можливість глибокого аналізу трафіку, побудови топологій мережі, виявлення топ-споживачів каналу (Top Talkers) та моніторингу додатків.

До переваг PRTG Network Monitor можна віднести зручність розгортання, велику кількість готових сенсорів (понад 250 типів), можливість віддаленого моніторингу, детальну систему сповіщень та зрозумілий

інтерфейс. Продукт також підтримує кластеризацію та розподілене розгортання для великих мереж.

До недоліків даного рішення відноситься:

- висока вартість ліцензій (ліцензування відбувається за кількістю сенсорів);
- значні апаратні вимоги до сервера при великій кількості сенсорів;
- менша глибина декодування окремих пакетів порівняно з класичними сніферами типу Wireshark;
- залежність від Windows Server для основного сервера моніторингу;
- необхідність платної технічної підтримки для отримання оновлень та повної функціональності.

1.2.5. Zeek

Zeek є потужним відкритим інструментом для мережевого аналізу та моніторингу, який активно використовується в академічному середовищі, дослідницьких установах та центрах реагування на комп'ютерні інциденти (SOC). На відміну від класичних сніферів, Zeek не є звичайним пакетним аналізатором, а швидше семантичним аналізатором мережевого трафіку.

Zeek поширюється на умовах відкритої ліцензії BSD і є повністю безкоштовним. Він призначений для глибокого аналізу мережевого трафіку в реальному часі або з попередньо збережених файлів. Zeek здатний перехоплювати пакети, виконувати їх семантичну обробку та генерувати структуровані логи, які містять багату інформацію про з'єднання, файли, сертифікати, HTTP-сесії, DNS-запити та багато іншого.

Програма працює переважно в режимі командного рядка і не має вбудованого графічного інтерфейсу. Zeek використовує власну скриптову мову для написання правил аналізу, що робить його надзвичайно гнучким інструментом. Завдяки цьому фахівці можуть створювати власні скрипти для виявлення специфічних загроз або аномалій.

Основними перевагами Zeek є висока продуктивність навіть на

високошвидкісних каналах, глибокий семантичний аналіз (незалежно від шифрування на транспортному рівні), можливість інтеграції з іншими системами безпеки (SIEM, Elasticsearch, Splunk) та активна спільнота розробників.

До недоліків даного інструменту належить:

- відсутність графічного інтерфейсу (необхідно використовувати сторонні рішення, такі як Zeek-osquery, Kibana тощо);
- висока складність освоєння (потрібні хороші знання мереж та програмування);
- значні вимоги до обчислювальних ресурсів при аналізі великих обсягів трафіку;
- відсутність готових «коробкових» рішень для невеликих організацій;
- складність первинного налаштування та конфігурування.

Zeek є одним з найпотужніших інструментів для професійного та дослідницького аналізу мережевого трафіку і часто використовується поряд з Wireshark: перший — для автоматизованого моніторингу та виявлення аномалій, другий — для глибокого ручного аналізу конкретних сесій.

1.2.6. Wireshark

Одним з найпопулярніших аналізаторів пакетів у комп'ютерній мережі є Wireshark. Аналізатор Wireshark дозволяє виконувати захоплення пакетів з мережі, до якої комп'ютер підключено та проводити детальний аналіз його вмісту. Як правило аналізатор використовується:

- для рішення проблем із мережею;
- для оцінки рівня мережної безпеки;
- для вивчення мережевих протоколів та виявлення хибного їх використання;
- для діагностування роботи мережевого програмного забезпечення, що розробляється;
- для виявлення типів протоколів, що використовуються в локальній

мережі;

- для виявлення прихованого мережевого трафіку.

Основними відмінностями Wireshark є:

- кросплатформна реалізація;
- захоплення пакетів, що надходять до мережевого адаптеру;
- повна деталізація інформації про протоколи, що використано у пакетах;
- збереження даних пакету, який було захоплено, та можливість подальшого аналізу цих даних;
- імпортування та експортування пакетів у різні відомі формати для інших програм-аналізаторів;
- фільтрування пакетів за багатьма критеріями;
- пошук пакетів за багатьма критеріями;
- забарвлення рядків із захопленими пакетами відповідно до фільтрів;
- створення різноманітних статистик;
- та інше.

Захоплення кадрів можна виконати через меню Capture–Options або за допомогою команд Capture-Start та Capture-Stop. Через вкладку Options можна зробити додаткові налаштування, щодо процесу захоплення та показу захоплених даних під час його виконання.

У верхній частині вікна аналізатору Wireshark показуються захоплені кадри з мережі з вказівкою їх основних параметрів: мережевої адреси відправника та отримувача, типу кадру, довжини у байтах, часу отримання після початку процесу захоплення. У центральній частині показується ієрархічна структура мережевих протоколів, що використовуються на відповідних рівнях передавання даних. Якщо розгорнути певний рівень, то можна побачити опис полів його заголовку. У нижній частині вікна - шістнадцятирічний дамп кадру (байти кадру, які представлено у шістнадцятковому форматі)[1-5].

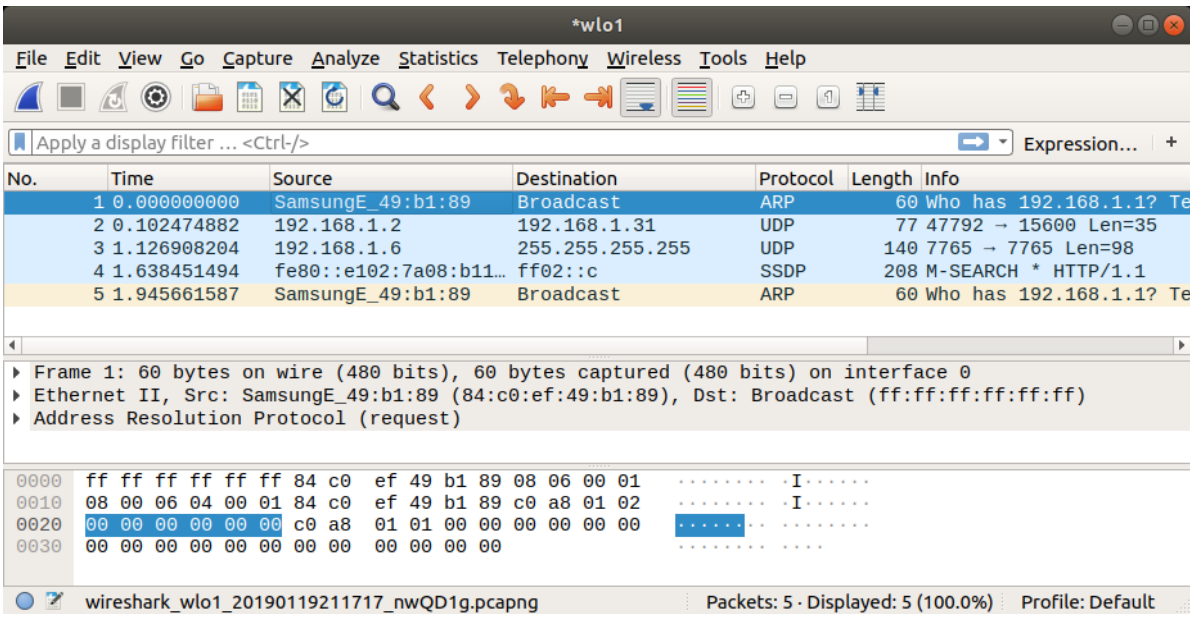

Рис. 1.5. Вигляд вікна програми Wireshark

Wireshark має деякі недоліки: складність створення та налаштування правил фільтрації, інтерфейс програми містить велику кількість елементів керування та службової інформації, менша продуктивність роботи порівняно з іншими програмами.

1.3. Безпека та етичні аспекти перехоплення мережевого трафіку

Перехоплення та аналіз мережевого трафіку є потужним інструментом діагностики та дослідження, проте цей процес пов’язаний зі значними правовими, етичними та технічними ризиками. Оскільки мережевий трафік часто містить персональні дані, конфіденційну інформацію та комерційну таємницю, неконтрольоване використання засобів захоплення пакетів може призвести до порушення законодавства та етичних норм.

1.3.1. Правові аспекти

В Україні діяльність з перехоплення мережевого трафіку регулюється низкою нормативно-правових актів. Згідно зі Законом України «Про захист персональних даних» (від 01.06.2010 № 2297-VI) [6], обробка персональних даних, до яких належать IP-адреси, MAC-адреси, вміст пакетів (логіни, паролі, cookies тощо), допускається лише за наявності правової підстави

(згода суб’єкта даних, необхідність виконання договору або виконання обов’язків, покладених на володільця даних).

Кримінальний кодекс України (ст. 163, 182) передбачає відповідальність за незаконне втручання в роботу електронно-обчислювальних машин, систем та мереж, а також за порушення недоторканності приватного життя. Несанкціоноване перехоплення трафіку в чужій мережі може бути кваліфіковане як злочин [7].

На міжнародному рівні важливе значення має Загальний регламент захисту даних (GDPR) Європейського Союзу, а також Конвенція про кіберзлочинність (Будапештська конвенція), ратифікована Україною. Згідно з цими документами, перехоплення комунікацій без чіткої правової підстави є неприпустимим.

У контексті академічних досліджень перехоплення трафіку дозволене лише в межах власної лабораторної або навчальної мережі.

1.3.2. Етичні аспекти

Етичне використання інструментів моніторингу трафіку базується на таких ключових принципах:

- Принцип згоди – усі учасники мережі, трафік яких може бути перехоплений, повинні бути поінформовані та надати згоду.
- Принцип мінімального втручання – збирати лише ті дані, які необхідні для досягнення цілей дослідження.
- Принцип конфіденційності та анонімізації – отримані дані повинні бути захищені від несанкціонованого доступу, а персональна інформація анонімізована або псевдонімізована.
- Принцип відповідальності – дослідник несе повну відповідальність за можливу шкоду, спричинену витоком даних або неправильним використанням інструменту.

Особливо важливо дотримуватися етичних норм.

1.3.3. Технічні заходи безпеки

Для забезпечення безпеки процесу перехоплення та зберігання даних у розробленій системі рекомендовано впровадити такі заходи:

- Запуск програми з мінімально необхідними правами (Principle of Least Privilege).
- Шифрування файлів зі збереженими дампами трафіку (наприклад, за допомогою AES-256).
- Обмеження часу зберігання захоплених даних.
- Ведення детального журналу всіх дій користувача програми.
- Автоматичне видалення чутливих даних (наприклад, payload з паролями) після завершення аналізу.
- Використання фільтрів для виключення захоплення конфіденційного трафіку (HTTPS, SSH тощо) під час демонстрації.

Дотримання правових та етичних норм є невід’ємною частиною професійної компетентності спеціаліста.

РОЗДІЛ 2

ВИБІР МЕТОДУ РІШЕННЯ

2.1. Технологія захоплення трафіку Npcap

Для реалізації захоплення мережевого трафіку в сучасних умовах найбільш доцільним є використання бібліотеки Npcap. Npcap є офіційним форком Npcap, який активно підтримується, сумісний з Windows 10/11 та відповідає сучасним вимогам безпеки операційної системи.

Архітектура рішення базується на використанні бібліотеки SharpPcap – керованій .NET-обгортці над libpcap/Npcap. SharpPcap дозволяє працювати з пакетами на високому рівні, забезпечуючи зручний API для перехоплення, фільтрації та обробки мережних пакетів.

На рис. 2.1 наведено спрощену структуру стека захоплення пакетів.

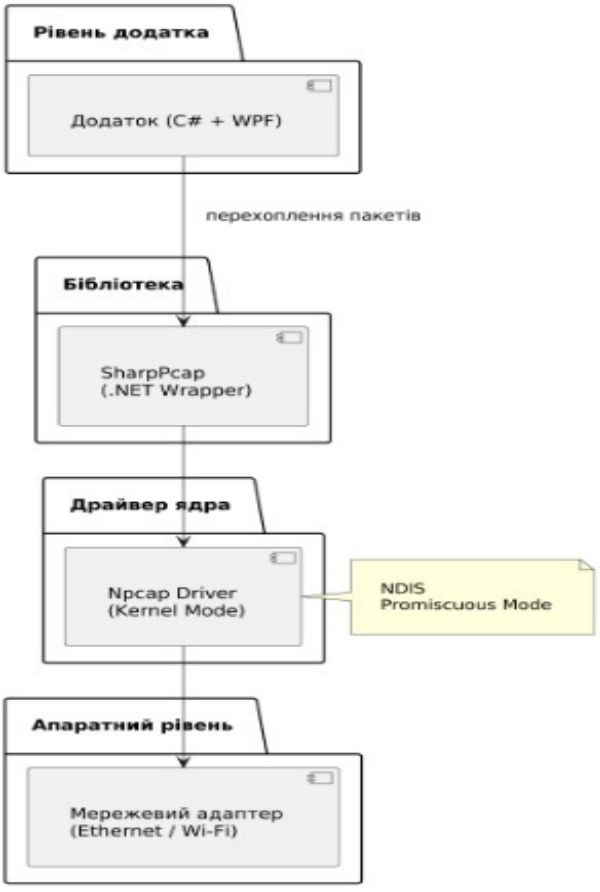

Рис. 2.1 Структура стека захоплення пакетів

Npcap працює на рівні ядра операційної системи та взаємодіє з мережевим адаптером через NDIS (Network Driver Interface Specification). Бібліотека дозволяє перевести мережевий інтерфейс у режим promiscuous (змішаний режим), у результаті мережевий адаптер отримує можливість обробляти весь потік переданих даних у сегменті мережі, включаючи пакети для інших пристроїв [10].

- Основними перевагами використання SharpPcap + Npcap порівняно є:
- Підтримка сучасних версій Windows;
- Краща сумісність з 64-бітними системами;
- Висока продуктивність і стабільність;
- Підтримка сучасних механізмів фільтрації (BPF-фільтри);
- Покращена безпека драйвера.

Таким чином, вибір технології Npcap + SharpPcap дозволяє створити сучасне, ефективне та безпечне рішення для моніторингу мережевого трафіку.

2.1.1 Драйвери

Драйвер захоплення пакетів Npcap (npcap.sys) є NDIS 6 Lightweight Filter драйвером, забезпечує значно кращу сумісність, продуктивність і безпеку.

NDIS (Network Driver Interface Specification) – це специфікація Microsoft, яка визначає стандартну архітектуру для мережевих драйверів Windows. У сучасній реалізації розрізняють такі основні типи драйверів:

NDIS Miniport Drivers – драйвери мережевих адаптерів (Ethernet, Wi-Fi тощо);

NDIS Protocol Drivers – драйвери протоколів верхнього рівня;

NDIS Lightweight Filter Drivers – проміжні фільтруючі драйвери (до яких належить Npcap) [10].

Npcap працює як Lightweight Filter Driver, що дозволяє йому ефективно

перехоплювати пакети на рівні ядра, мінімально впливаючи на продуктивність системи. Драйвер підтримує два основних режими роботи:

Normal mode – захоплення лише тих пакетів, які адресовані даному комп'ютеру.

Promiscuous mode (змішаний режим) – захоплення всіх пакетів, що проходять через мережевий інтерфейс.

Для нормальної роботи драйвера захоплення пакетів необхідна взаємодія як з драйвером мережевого адаптера, так і з додатком користувача. Архітектура Npcap дозволяє працювати з більшістю сучасних мережевих адаптерів (Ethernet, Wi-Fi), а також з віртуальними інтерфейсами (VMware, Hyper-V, VirtualBox тощо).

Переваги NDIS 6 Lightweight Filter драйвера Npcap:

- Висока продуктивність і низьке споживання ресурсів;
- Підтримка апаратного прискорення мережевих карт;
- Краща стабільність і безпека порівняно зі старими драйверами;
- Підтримка сучасних версій Windows (від Windows 7 до Windows 11);
- Можливість одночасної роботи з кількома додатками.

Таким чином, використання сучасного NDIS 6 Lightweight Filter драйвера є ключовим фактором ефективної та надійної роботи розробленої інформаційної системи моніторингу мережевого трафіку.

2.1.2. Делегати

Однією з важливих технічних проблем при розробці системи моніторингу трафіку є необхідність оновлення елементів графічного інтерфейсу з потоку захоплення пакетів, який відрізняється від основного UI-потоку WPF. Для безпечного звернення до елементів керування з іншого потоку в .NET використовуються делегати.

Делегат – це типобезпечний вказівник на метод, який дозволяє інкапсулювати посилання на функцію та передавати її як параметр. У мові C#

делегати є основою реалізації подій та асинхронного програмування.

У розробленій системі делегати застосовуються для оновлення інтерфейсу в реальному часі при отриманні нових пакетів. Основний механізм реалізовано через метод Invoke або BeginInvoke об’єкта Dispatcher (у WPF):

C#

```
Application.Current.Dispatcher.Invoke(() => {
});
```

Також використовуються вбудовані делегати Action та Func <T>, що спрощує код. Для обробки події прибуття нового пакета (OnPacketArrival) застосовується делегат типу PacketArrivalEventHandler.

Переваги використання делегатів у даній системі:

- Забезпечення потокобезпечності при роботі з UI;
- Можливість асинхронної обробки подій;
- Гнучкість та розширюваність коду;
- Підтримка багатопоточної архітектури програми.

У сучасних версіях .NET делегати часто поєднуються з механізмами async/await та Task, що дозволяє створювати більш читабельний та ефективний код порівняно з класичними делегатами попередніх версій.

Під час реалізації інформаційної системи важливо забезпечити доступ до елементів інтерфейсу з потоку, відмінного від того, у якому відбувається перехоплення пакетів. Для цього створюється спеціальний асинхронний метод, який виконує взаємодію з компонентами форми без призупинення роботи програми. До такого методу передаються делегат та необхідні параметри.

Делегати являють собою об’єкти, що містять посилання на методи та забезпечують зручні засоби роботи з ними. Усі делегати є екземплярами відповідних типів і можуть застосовуватися разом із сучасними механізмами async/await, Dispatcher.InvokeAsync, IProgress<T> або ObservableCollection.

Особливістю делегатів є можливість об’єднання кількох методів у

єдиний ланцюжок викликів. Завдяки цьому один делегат може послідовно запускати декілька функцій. Для зовнішнього коду він сприймається як один об’єкт, хоча фактично містить кілька методів (рис. 2.2).

Такий механізм широко використовується під час реалізації подій, оскільки дозволяє підключати кілька обробників без застосування додаткових інструментів. По суті, делегат є об’єктом, який приховує в собі адреси функцій. Його особливість в підтримці з боку середовища виконання та компілятора, що забезпечує спеціальні засоби для роботи з такими об’єктами [6].

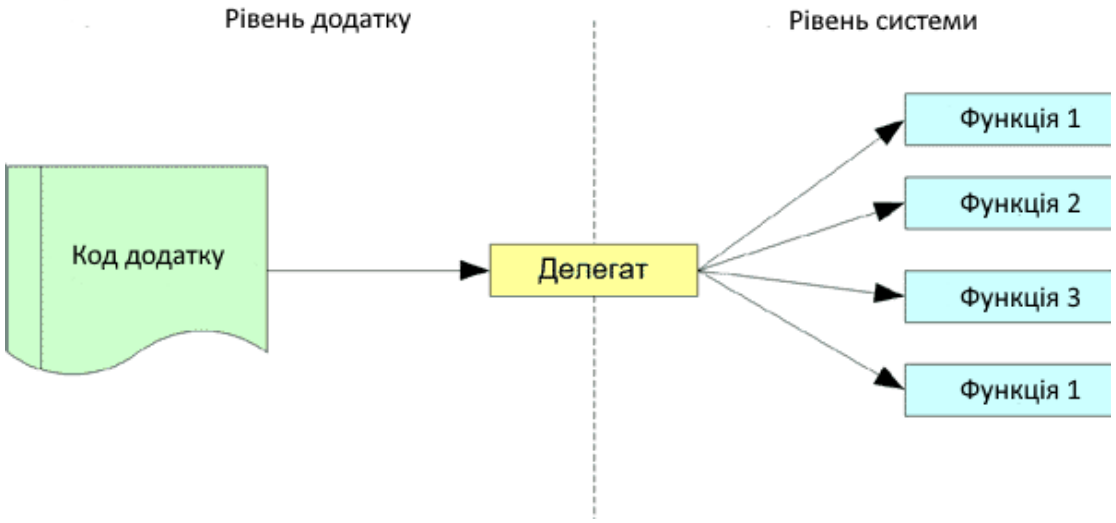

Рис. 2.2. Структура застосування делегатів

У середовищі .NET усі методи поділяються на статичні (static) та екземплярні (instance). Якщо делегат пов’язаний зі статичним методом, для його виклику достатньо адреси функції та параметрів. У випадку використання екземплярних методів необхідно також зберігати інформацію про об’єкт, якому належить цей метод. Таке посилання записується під час створення делегата та залишається незмінним протягом усього періоду його існування [11].

Незалежно від того, чи працює делегат зі статичною функцією, чи з методом певного об’єкта, спосіб його виклику залишається однаковим. Усі необхідні механізми реалізуються засобами самого делегата та середовища

виконання, що значно спрощує організацію подій і дозволяє прив’язувати до них різні делегати.

Отже, правильне використання делегатів і засобів синхронізації потоків є важливою умовою стабільної та надійної роботи інформаційної системи в умовах інтенсивного мережевого навантаження.

2.2. Метод аналізу трафіку

Для відображення зміни навантаження на мережу та побудови графічних залежностей було обрано метод ковзаючих середніх. Його суть полягає в тому, що значення функції в кожній точці визначається як середнє значення певної кількості попередніх вимірювань. Такий підхід забезпечує згладжування різких коливань і дозволяє простежувати закономірності та циклічність змін.

Розрізняють три основні типи ковзаючих середніх:

– прості;

– зважені;

– експоненціальні.

Проста ковзаюча середня визначається як середнє арифметичне значення вибраних елементів часового ряду та обчислюється за формулою (2.1):

$$SMA_{t=}\ \frac{1}{n}\sum_{i=0}^{n-1} p_{t-i} = \frac{p_t+p_{t-1}+\cdots p_{t-i}\cdots+p_{t-n+2}+p_{t-n+1}}{n} \quad (2.1)$$

де SMA_t – значення функції у момент *t*;

n – кількість значень вихідної функції для розрахунку змінного середнього;

p_{t-} – значення вихідної функції в точці *t-i*.

До недоліків простої ковзаючої середньої належать:

– однакова вага для всіх елементів;

– повторний вплив кожного значення на результат.

У деяких випадках доцільно надавати новішим значенням більшу важливість.

Зважена ковзаюча середня відрізняється тим, що окремі значення мають різні вагові коефіцієнти, які утворюють арифметичну прогресію. Завдяки цьому останні значення ряду мають більший вплив на результат, ніж попередні.

Для прогресії з початковим значенням і кроком, що дорівнює одиниці, формула набуває вигляду (2.2):

$$WMA_t = \frac{2}{n\times(n+1)} \sum_{i=0}^{n-1} (n-i) \times p_{t-i} \quad (2.2)$$

де WMA_t – значення в точці *t*;

n – значення функції у момент;

p_{t-} – значення часового ряду, віддалене від поточного на *i* інтервалів.

Знаменник функції дорівнює сумі членів арифметичної прогресії з початковим членом і кроком рівними 1: $n\times(n+1)$:

$$\frac{n\times(n+1)}{2} \quad (2.3)$$

Експоненціальна ковзаюча середня є різновидом зваженої середньої, у якій вплив попередніх значень зменшується за експоненціальним законом і ніколи не стає нульовим. Її математичний вираз має вигляд:

$$EMA_t = \alpha \times p_t + (1-\alpha) \times EMA_{t-1} \quad (2.4)$$

де EMA_t – значення у поточний момент часу;

EMA_{t-1} – попереднє значення експоненціальної середньої *t-1*,

p_t – поточне значення досліджуваної величини *t*;

α – коефіцієнт згладжування, який змінюється в межах від 0 до 1.

Значення α може задаватися довільно або визначатися через період усереднення:

$$\alpha = \frac{2}{n+1} \quad (2.5)$$

Під час аналізу мережевого трафіку на невеликих часових інтервалах найбільш доцільним є використання експоненціальної ковзаючої середньої, оскільки вона швидше реагує на зміни інтенсивності трафіку. Для довготривалих досліджень більш придатною є проста ковзаюча середня.

2.3. Засоби розробки програмного забезпечення

Для створення програмного продукту було обрано мову програмування C# та технологію Windows Presentation Foundation (WPF). Така комбінація забезпечує високу швидкодію застосунку, сучасний зовнішній вигляд інтерфейсу та зручність процесу розроблення. Порівняно з іншими мовами програмування, C# надає широкі можливості для роботи з мережевими з'єднаннями, мережевими адаптерами та засобами аналізу трафіку. Достатня кількість бібліотек і готових компонентів дає можливість взаємодіяти з різноманітними типами мережевого обладнання та обробляти різні види трафіку. Мова C# має інструменти для створення графічних застосунків.

Мова C# є однією з основних мов платформи .NET. Вона побудована на принципах об'єктно-орієнтованого програмування та дозволяє створювати програмні компоненти, які можуть використовуватися повторно в різних типах програмного забезпечення. У процесі розроблення C# було враховано досвід застосування таких мов, як C++ і Java, що дало змогу усунути низку недоліків, притаманних попереднім технологіям. Однією з характерних особливостей C# є те, що вона належить до керованих мов програмування, а виконання програм здійснюється за допомогою середовища .NET Common Language Runtime (CLR) [12].

У поєднанні з платформою .NET мова C# є найефективнішим засобом для створення мережевих застосунків, бо має розвинену стандартну бібліотеку та велику кількість пакетів NuGet. Технологія WPF дає можливість реалізовувати сучасні, масштабовані та гнучкі інтерфейси користувача, використовуючи мову розмітки XAML разом із програмним кодом на C#.

Засоби C# дозволяють створювати графічні програми різного рівня складності. Технологія WPF, яка прийшла на зміну Windows Forms, характеризується ширшими функціональними можливостями та використовується як підсистема побудови інтерфейсу користувача. Вона є складовою платформи .NET і забезпечує незалежність від фізичної роздільної здатності екрана завдяки використанню векторної системи візуалізації. Основою графічного механізму WPF є технологія DirectX, що дозволяє застосовувати апаратне прискорення та підвищувати ефективність використання ресурсів комп'ютера. Важливою перевагою є також можливість комбінування декларативної мови XAML із програмним кодом, написаним мовою C# [13].

Схематичне представлення архітектури WPF наведено на рисунку 2.3.

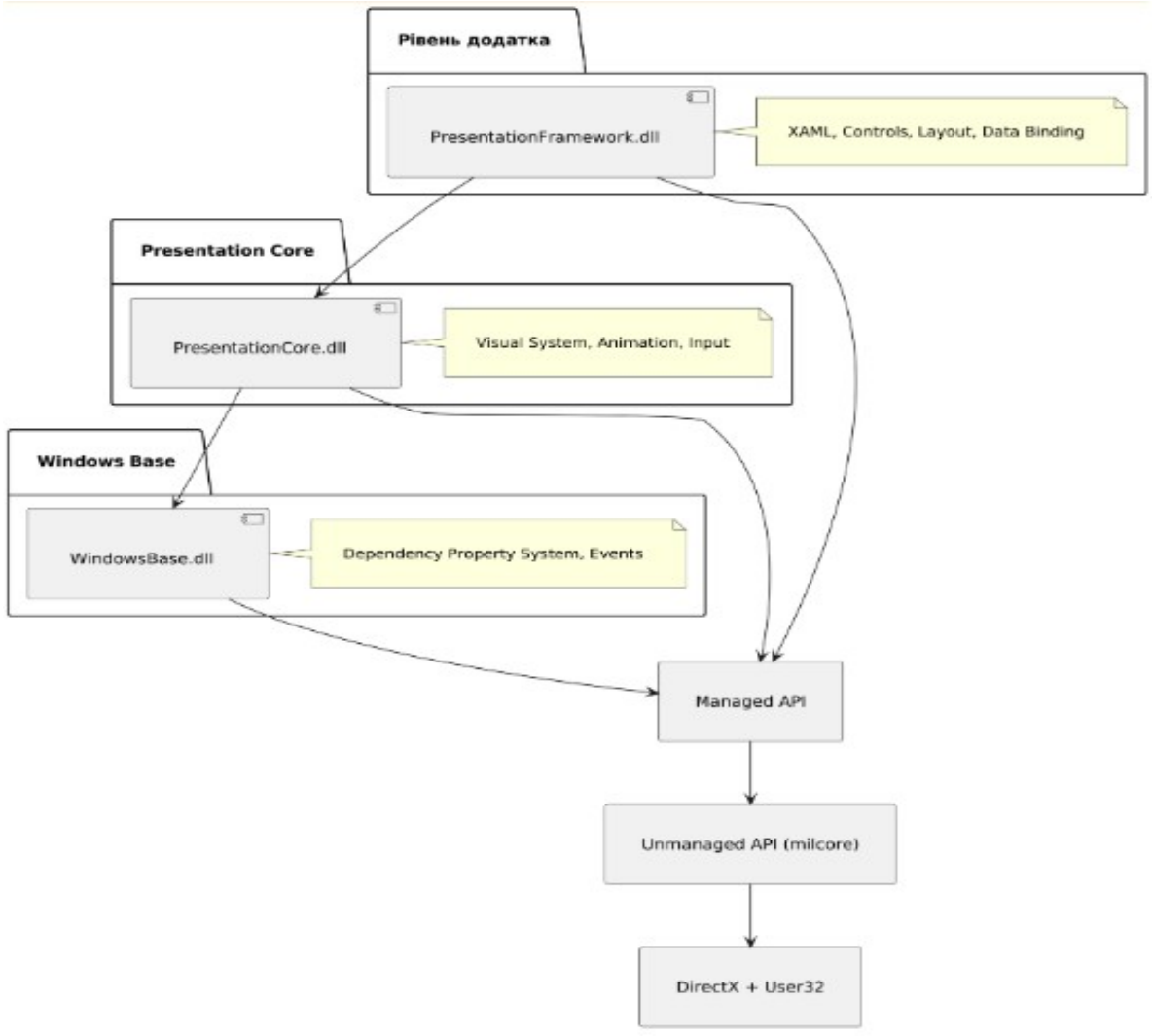

Рис. 2.3 – Архітектура WPF

Засобами розроблення обрано використати інтегроване середовище Microsoft Visual Studio та редактор Visual Studio Code. Ці програмні продукти забезпечують зручні інструменти для створення, тестування та налагодження програмного забезпечення. Microsoft Visual Studio надає широкі можливості для розробки застосунків на основі WPF, а також підтримує створення веб-орієнтованих проєктів. У свою чергу, Visual Studio Code є більш легким середовищем, яке не потребує значних системних ресурсів і може ефективно використовуватися для швидкого редагування та відлагодження програмного коду. Visual Studio підтримує підключення додаткових модулів і розширень, що дозволяє адаптувати середовище до конкретних завдань та значно спрощує процес розроблення програмного забезпечення [14].

РОЗДІЛ 3
ІНФОРМАЦІЙНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ

3.1. Функціональні можливості програмного комплексу

Першим етапом реалізації системи моніторингу та дослідження мережевого трафіку стало створення проєкту на основі технології WPF, яка забезпечує наявність графічного інтерфейсу користувача. Про використання консольного застосунку було прийнято рішення що це буде недоцільним, тому, що такий підхід не дозволяє організувати зручне керування та відображення результатів. Архітектура програмного комплексу складається з трьох основних компонентів:

- модуль взаємодії з користувачем;
- функціонально-логічний модуль;
- набір допоміжних бібліотек.

Інтерфейс реалізовано у вигляді графічного вікна, яке містить вкладки з основними робочими областями, засоби керування, параметри налаштування та області відображення інформації.

Логічний рівень забезпечує виконання процесів моніторингу та аналізу мережевого трафіку. Роботу цього модуля можна поділити на два послідовні етапи. Перший етап пов'язаний зі збором початкових даних, під час якого визначається кількість пакетів, обсяг переданих даних, перелік активних вузлів, їхні адреси та використовувані мережеві протоколи. Другий етап полягає в обробці отриманої інформації, її порівнянні з раніше накопиченими результатами та виявленні факторів, які можуть призводити до нестабільного функціонування або зниження продуктивності мережі.

Для забезпечення доступу до мережевих адаптерів та реалізації перехоплення пакетів використовуються сторонні програмні бібліотеки: SharpPcap та PacketDotNet. Бібліотека Pcap встановлюється разом із драйвером Npcap, тому окремого підключення не потребує. Додатково

використовується простір імен System.Net.NetworkInformation, який дозволяє отримувати відомості про доступні мережеві інтерфейси.

У процесі створення програмного забезпечення були реалізовані окремі класи та методи, які забезпечують функціонування алгоритмів обробки даних.

Програмний продукт реалізований у вигляді настільного застосунку PocketCapt. До його основних компонентів належать:

- – MainWindow – модуль головного вікна;
- – CaptureEngine – компонент керування процесом захоплення мережевих пакетів;
- – PacketProcessor – модуль декодування та обробки отриманих пакетів;
- – StatisticsManager – засіб накопичення та відображення статистичних даних;
- – GraphBuilder – компонент побудови графічних залежностей із використанням ковзного середнього;
- – FilterManager – підсистема фільтрації трафіку;
- – SettingsManager – модуль роботи з параметрами користувача.

У таблиці 3.1 наведено перелік основних методів, що використовуються в інформаційній системі.

Таблиця 3.1

Основні методи, які використовуються в інформаційній системі

Назва методу	Запуск процесу захоплення пакетів
`StartCapture()`	Зупинення захоплення
`StopCapture()`	Обробка нового пакета
`OnPacketArrival()`	Застосування фільтрів
`ApplyFilters()`	Побудова графіка ковзного середнього
`BuildMovingAverage()`	Збереження даних у файл
`SaveToFile()`	Отримання списку мережевих інтерфейсів
`GetNetworkDevices()`	Запуск процесу захоплення пакетів

На рисунку 3.1 наведено блок-схему, яка відображає загальний алгоритм роботи розробленої інформаційної системи.

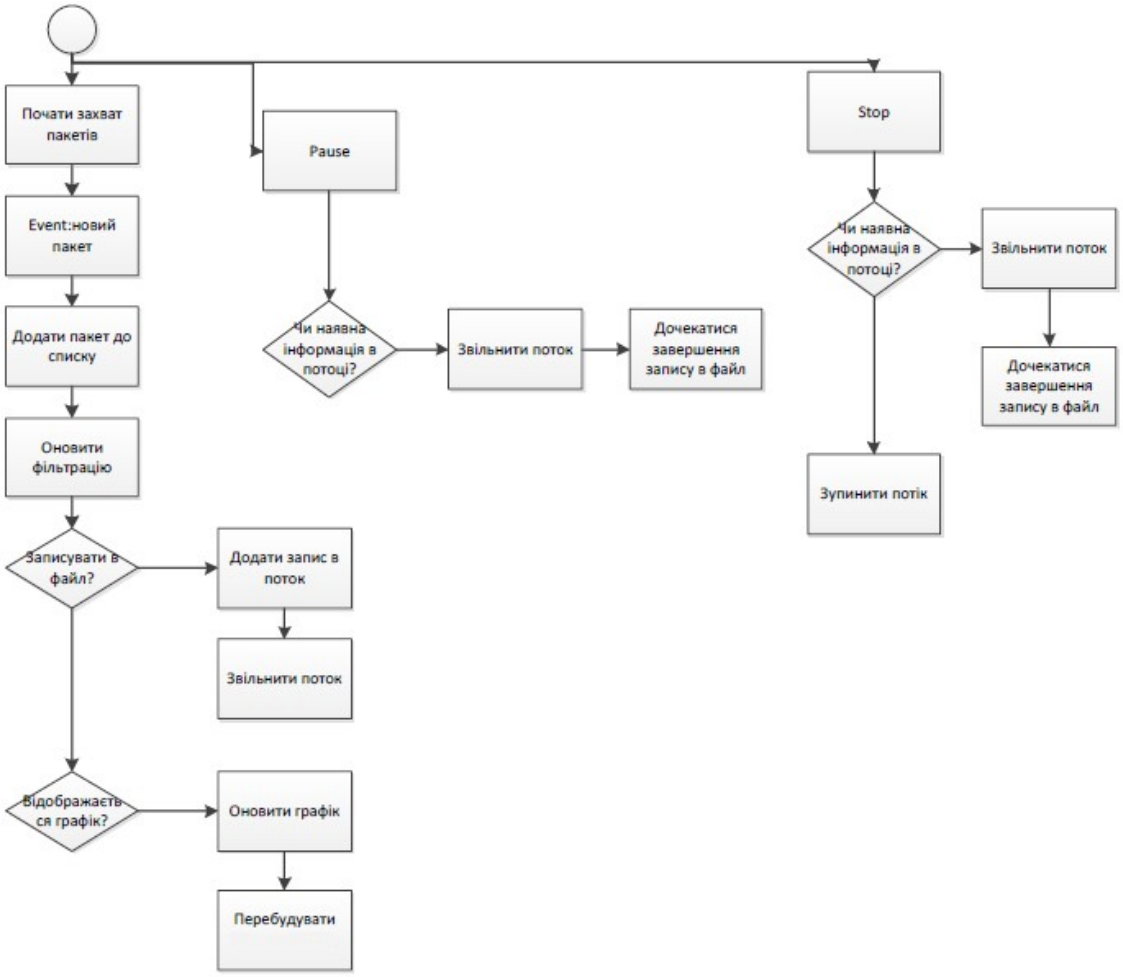

Рис. 3.1. Блок-схема алгоритму роботи програми

3.2. Опис програмного забезпечення

Перед початком використання інформаційної системи моніторингу та аналізу мережевого трафіку потрібно переконатися в наявності та коректному налаштуванні мережевих інтерфейсів, і у тому, що комп’ютер підключений до мережі та в ній присутні активні вузли, обмін даними між якими може бути проаналізований.

Для стабільного функціонування програми рекомендується використовувати комп’ютер із такими мінімальними характеристиками:

- Процесор: Intel Core i3 / AMD Ryzen 3 і вище;
- Оперативна пам’ять: не менше 8 ГБ;
- Вільне місце на диску: не менше 500 МБ;
- Операційна система: Windows 10 (версія 1809) або Windows 11.

Крім рекомендованих параметрів, є ряд обов'язкових вимог, без виконання яких робота застосунку буде неможливою:

- Встановлений драйвер Npcap (остання версія);
- .NET 6 / .NET 8.0 Desktop Runtime;
- Мережевий адаптер, який підтримує режим promiscuous;
- Запуск програми з правами адміністратора;
- Вимкнені мережеві екрани (Firewall) під час тестування (або додані відповідні правила).

Перед запуском інформаційної системи потрібно перевірити структуру каталогу, у якому знаходиться виконуваний файл програми (рис. 3.2).

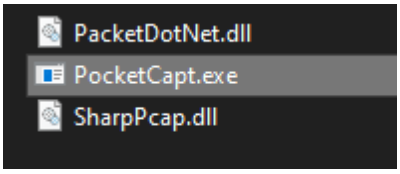

Рис. 3.2. Структура папки з програмою перед запуском

Після встановлення пакета Npcap та перевірки наявності всіх необхідних бібліотек можна запускати розроблений застосунок PocketCapt.

Після відкриття програми користувач потрапляє до вкладки «Interfaces» (рис. 3.3), яка призначена для вибору мережевого інтерфейсу, що буде використовуватися для захоплення трафіку. Після вибору пристрою у правій частині вікна відображаються його характеристики, а саме:

– системне ім'я інтерфейсу;

– назва, призначена користувачеві;

– MAC-адреса;

– IP-адреса.

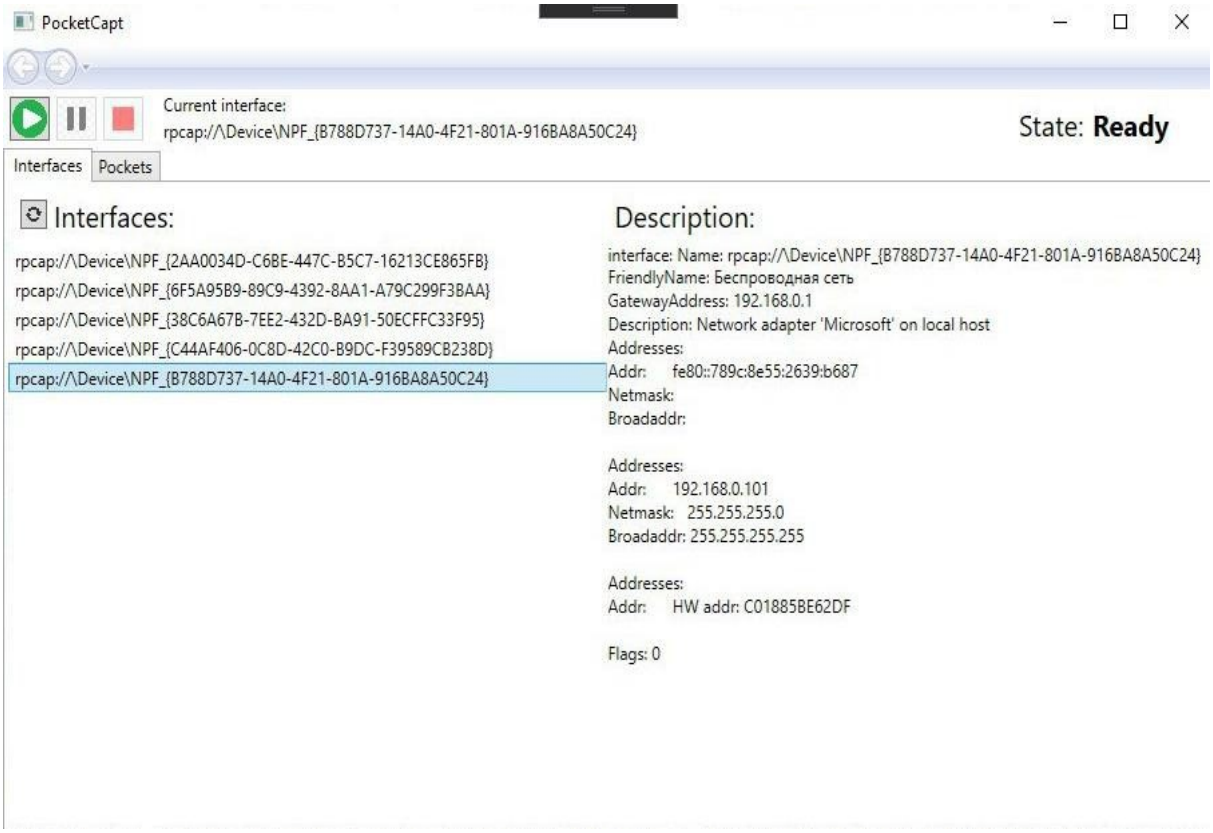

Рис. 3.3. Головне вікно додатку «PocketCapt»

Для початку перехоплення пакетів потрібно перейти на вкладку «Packets» (рис. 3.4), яка містить основні інструменти взаємодії з процесом перехоплення даних. Користувачу доступні такі елементи:

- – кнопки керування захопленням пакетів (Play, Pause, Stop);
- – індикатор поточного стану роботи системи (Ready, Pause, Running);
- – кнопка відкриття параметрів програми «Settings» (рис. 3.5);
- – кнопка переходу до вікна побудови графіків (рис. 3.6);
- – область налаштування фільтрів та вибору кольорових позначень;
- – інструменти для збереження отриманих даних.

Система фільтрації дозволяє задавати параметри пошуку, визначати значення критеріїв та призначати кольори для більш зручного візуального аналізу. Передбачено можливість створення додаткових правил. Якщо декілька фільтрів конфліктують між собою, пріоритет отримує правило, яке розташоване вище у списку. Для сортування інформації достатньо натиснути на заголовок відповідної колонки, а подвійне натискання на рядок відкриває

детальні відомості про вибраний пакет.

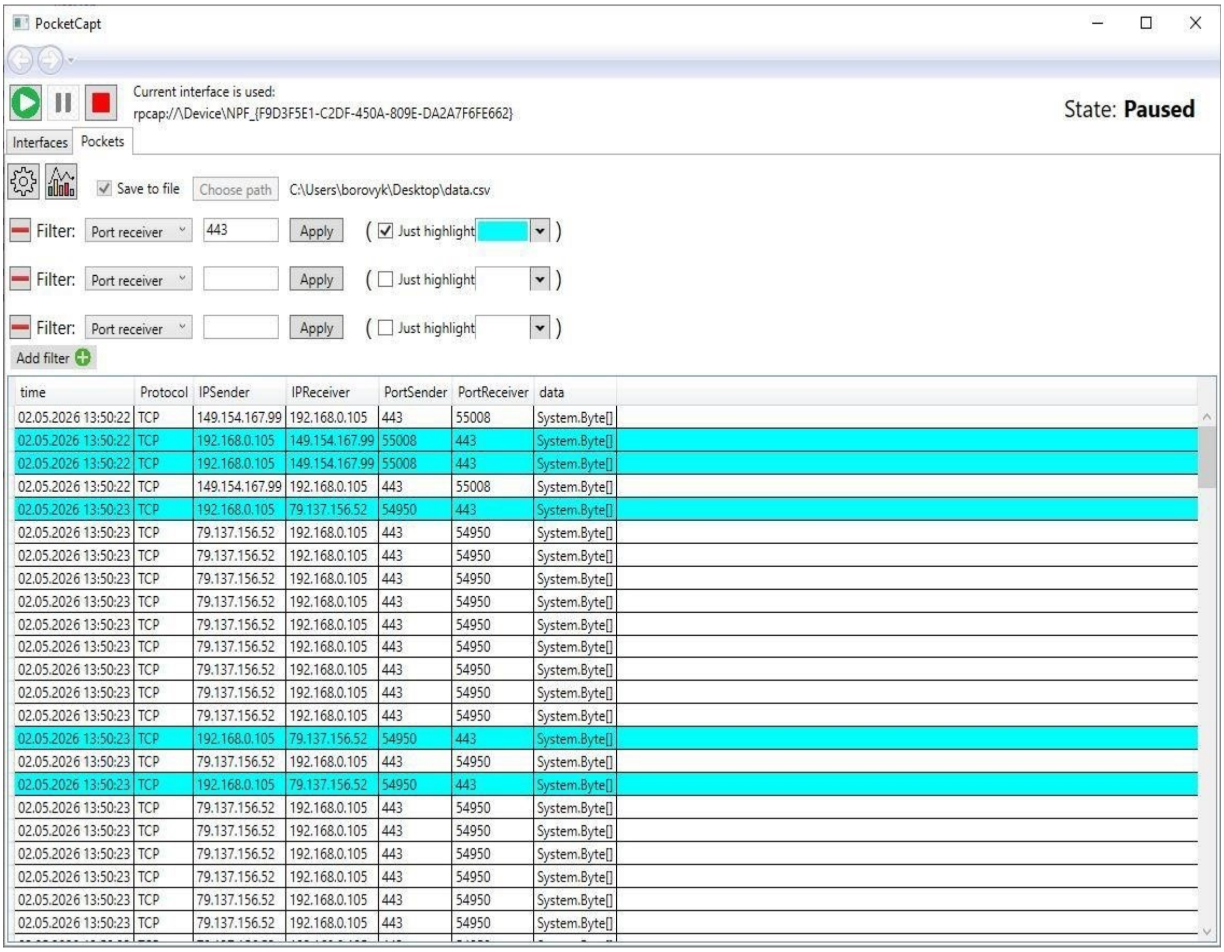

Рис. 3.4. Вкладка «Pockets»

Вікно налаштувань, яке відкривається після натискання кнопки «Settings», дозволяє визначити перелік стовпців, що відображатимуться у таблиці результатів.

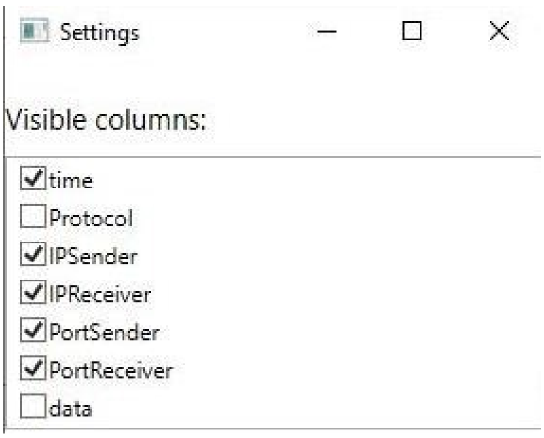

Рис. 3.5. Вікно налаштувань програми

Для запису результатів у файл необхідно активувати параметр «Save to file» та вказати директорію, у яку будуть збережені отримані дані.

На основі зібраної інформації програма дає змогу будувати графіки. За стандартними налаштуваннями відображається зміна розміру пакетів у часі. Користувач може змінювати параметри групування даних, задаючи необхідний часовий інтервал.

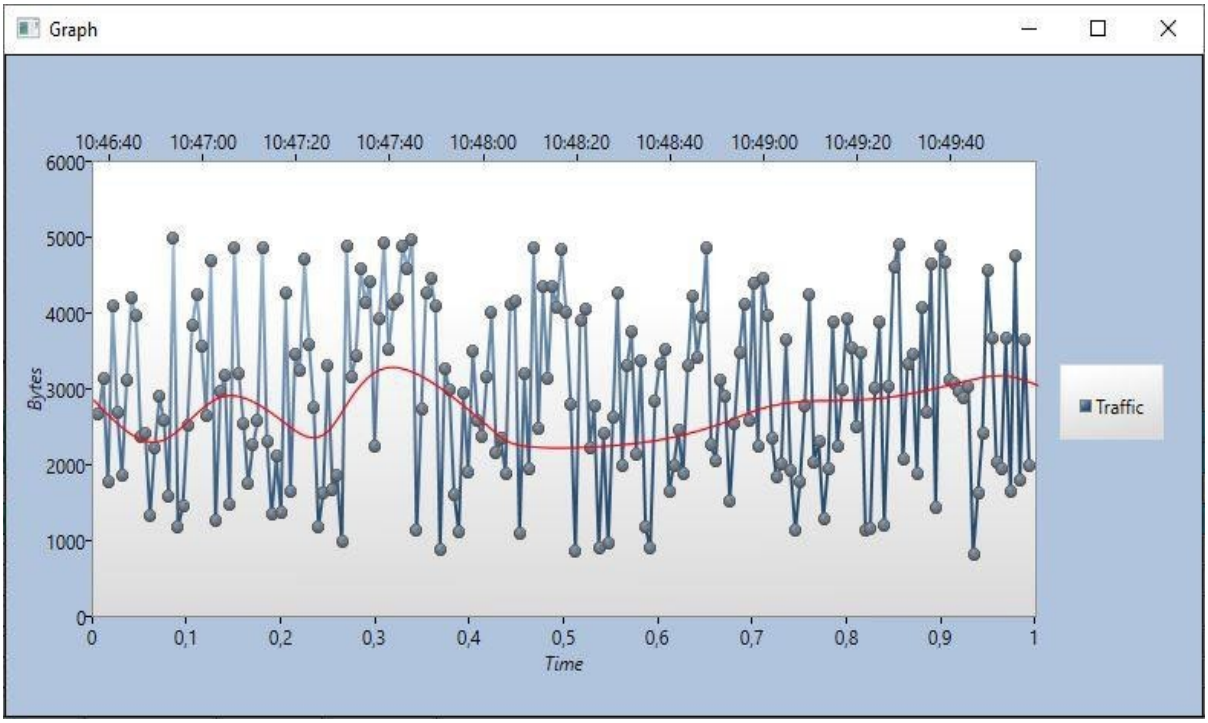

Рис. 3.6.Вікно відображення графіка

Для покращення сприйняття статистичних даних реалізовано підтримку методу Exponential Weighted Moving Average (EWMA), який дозволяє згладжувати випадкові коливання та відображати загальні тенденції зміни навантаження. Побудова графіка виконується в режимі реального часу, а отримані результати можуть бути збережені у форматі CSV.

Реалізований функціонал забезпечує отримання відомостей про поточний стан локальної мережі та дозволяє спростити пошук ділянок, які негативно впливають на її продуктивність.

3.3. Безпека розробленої системи

Одним із найважливіших аспектів при розробці інформаційної системи моніторингу та аналізу мережевого трафіку є забезпечення її інформаційної безпеки. Оскільки програмне забезпечення працює в режимі захоплення пакетів у promiscuous-режимі, воно має доступ до всього трафіку, що циркулює в локальній комп'ютерній мережі. Це робить систему потужним інструментом для діагностики та адміністрування, але водночас створює значні ризики зловживань, витоку конфіденційної інформації та порушення законодавчих норм.

У сучасних умовах (станом на 2026 рік) питання безпеки мережевих аналізаторів є особливо актуальним. Згідно з даними міжнародних звітів, значна частина кіберінцидентів пов'язана з внутрішніми загрозами, коли легітимні інструменти використовуються для несанкціонованого збору даних. Крім того, широке впровадження віддаленої роботи, хмарних сервісів та Інтернету речей суттєво збільшило обсяг чутливого трафіку, що передається в локальних мережах. У зв'язку з цим під час проектування та реалізації системи було приділено особливу увагу комплексному забезпеченню безпеки на технічному, програмному та організаційному рівнях.

3.3.1 Аналіз потенційних загроз

Під час розробки системи проведено детальний аналіз можливих ризиків. Основні загрози можна класифікувати наступним чином:

- Несанкціонований доступ до системи. Зловмисник, який отримав доступ до комп'ютера адміністратора, може використовувати аналізатор для перехоплення конфіденційного трафіку.
- Внутрішні загрози. Легітимний користувач (адміністратор, інженер) може використовувати систему для несанкціонованого моніторингу діяльності інших співробітників.
- Витік даних при збереженні. Файли захопленого трафіку (.pcap) часто містять персональні дані, комерційну таємницю, автентифікаційну інформацію.

- Перевантаження мережі. Інтенсивне захоплення пакетів може призвести до зниження продуктивності мережі або навіть часткової відмови в обслуговуванні.
- Атаки на програмне забезпечення. Експлуатація вразливостей у коді програми або використаних бібліотеках (наприклад, buffer overflow).
- Юридичні та нормативні ризики. Порушення вимог Закону України «Про захист персональних даних», GDPR (у разі роботи з даними громадян ЄС) та внутрішніх політик безпеки організації.

Для наочності основні загрози та рівні їхнього впливу представлено в таблиці 3.2.

Таблиця 3.2

Основні загрози безпеки системи

Загроза	**Ймовірність**	**Вплив**	**Рівень ризику**
Несанкціонований доступ	Середня	Високий	Критичний
Зловживання легітимним користувачем	Висока	Високий	Критичний
Витік збережених даних	Середня	Високий	Високий
Перевантаження мережі	Низька	Середній	Середній
Експлуатація вразливостей коду	Низька	Високий	Високий

3.3.2 Технічні та програмні механізми захисту

У розробленій системі реалізовано багаторівневий підхід до забезпечення безпеки:

1. Контроль доступу:

- Обов’язкова перевірка прав адміністратора при запуску програми.
- Автоматична відмова в роботі без підвищених привілеїв.
- Можливість інтеграції з системою автентифікації Windows (Active Directory).

2. Логування та аудит:

- Детальне фіксування всіх дій користувача у файлі application.log.
- Записуються: дата і час події, ім’я користувача, обраний мережевий інтерфейс, застосовані фільтри, обсяг захоплених даних, операції збереження.
- Захист лог-файлу від несанкціонованих змін.

3. Захист збережених даних:

- Підтримка двох режимів збереження: стандартний .pcap та зашифрований .pcap.aes.
- Алгоритм шифрування – AES-256-GCM.
- Ключ шифрування генерується динамічно на основі пароля користувача з використанням PBKDF2.
- Автоматичне видалення тимчасових файлів після завершення сесії (за бажанням користувача).

4. Обмеження функціональності:

- «Безпечний режим» – автоматичне виключення захоплення чутливих протоколів (HTTP, HTTPS з розшифровкою, FTP, SMTP, POP3, Telnet).
- Можливість задати максимальний обсяг даних для однієї сесії захоплення.
- Таймер автоматичної зупинки захоплення.

5. Захист на рівні коду:

- Використання сучасної бібліотеки SharpPcap разом з Npcap.
- Перевірка цілісності основних бібліотек перед запуском.
- Рекомендоване підписання виконуваного файлу цифровим сертифікатом.

3.3.3 Організаційні заходи та рекомендації з експлуатації

Технічних заходів захисту недостатньо без відповідних організаційних рішень. Для безпечного використання системи рекомендується:

- Використовувати програмне забезпечення виключно на спеціально виділених робочих станціях адміністраторів мереж та спеціалістів з інформаційної безпеки.

- Обмежувати фізичний доступ до таких робочих місць.
- Розробити та затвердити на рівні організації «Політику використання системи моніторингу мережевого трафіку».
- Проводити обов’язкове навчання користувачів системи з питань інформаційної безпеки.
- Після завершення аналізу обов’язково видаляти або переміщувати в захищене сховище всі файли захопленого трафіку.
- Регулярно переглядати та аналізувати лог-файли системи.
- Не використовувати систему в мережах, де обробляються дані особливих категорій (медична таємниця, державна таємниця, банківська таємниця) без додаткових погоджень і сертифікації.

ВИСНОВКИ

У кваліфікаційній роботі спроєктовано та розроблено десктопний додаток PocketCapt для моніторингу та аналізу мережевого трафіку в локальних комп'ютерних мережах.

Розроблений застосунок є гнучкою об'єктно-орієнтованою системою, створеною мовою програмування C# із використанням технології Windows Presentation Foundation для побудови графічного інтерфейсу користувача.

Виконуючи кваліфікаційну роботу запропоновано підхід до створення програмного засобу для операційних систем сімейства Windows, який забезпечує перехоплення та анадізу мережевого трафіку. У процесі виконання було вирішено такі завдання:

- проведено аналіз предметної області моніторингу мережевого трафіку в локальних мережах та огляд сучасних інструментів;
- обґрунтовано вибір методів та технологій захоплення пакетів (C# + WPF + Npcap + SharpPcap);
- спроєктовано архітектуру програмного комплексу з урахуванням вимог продуктивності, модульності та безпеки;
- реалізовано модулі захоплення, фільтрації, візуалізації та аналізу трафіку;
- забезпечено можливість збереження даних, побудови графіків та прогнозування навантаженн;
- розроблено зручний графічний інтерфейс та передбачено можливість роботи в режимі командного рядка.
- проведено аналіз ризиків безпеки та запропоновано технічні та організаційні заходи захисту.

Функціональні можливості спроєктованого програмного забезпечення забезпечують захоплення пакетів у реальному часі, гнучку фільтрацію, статистичний аналіз, візуалізацію навантаження та прогнозування тенденцій за допомогою EWMA. Це надає зручний інструмент для адміністраторів

мереж, розробників та фахівців з інформаційної безпеки в невеликих і середніх локальних мережах.

Результат тестування додатку PocketCapt показав коректність захоплення та обробки трафіку, стабільність роботи інтерфейсу та надійність алгоритмів аналізу. Поряд з цим, результати апробації виявили перспективи подальшого удосконалення: перехід на сучасний API SharpPcap 6.x, заміну потокових конструкцій на асинхронні, розширення підтримки протоколів, додавання anomaly detection на базі машинного навчання та повноцінну реалізацію CLI-режиму.

Подальший розвиток програмного продукту може бути пов’язаний із розширенням можливостей налаштування фільтрів, удосконаленням механізмів аналізу мережевого трафіку, реалізацією статистики використання мережевих ресурсів окремими застосунками та розширеним відображенням вмісту пакетів.

В результаті виконання проєкту отримано нові знання і практичні навички з розробки мережевих додатків, роботи з пакетним захопленням, візуалізації даних та забезпечення безпеки інформаційних систем.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Goyal P., Goyal A. Comparative study of two most popular packet sniffing tools-Tcpdump and Wireshark.2017 9th International Conference on Computational Intelligence and Communication Networks (CICN). IEEE, 2017. P. 77–81. URL: https://ieeexplore.ieee.org/document/8319360
2. Deep Learning for Network Traffic Monitoring and Analysis (NTMA): A Survey. Computer Communications. 2021. Vol. 170. P. 19–41. DOI: 10.1016/j.comcom.2021.01.021. URL: https://www.sciencedirect.com/science/article/pii/S014036642100021X
3. Hamid I.R.A. et al. Network monitoring system to detect unauthorized connection. Acta Electronica Malaysia. 2017. Vol. 1, No. 2. P. 11–14. URL: https://www.actaelectronicamalaysia.com/archives/2017/2/11-14
4. Network Traffic Analysis Using Deep Packet Inspection. Journal of Network and Computer Applications. 2019. Vol. 137. P. 35–50. DOI: 10.1016/j.jnca.2019.04.005
5. Програма аналізатор бездротових мереж CommView URL: https://texnogid.biz.ua/wi-fi/coomview-for-wifi.html
6. Закон України «Про захист персональних даних» від 01.06.2010 № 2297-VI (зі змінами та доповненнями). URL: https://zakon.rada.gov.ua/laws/show/2297-17#Text
7. Кримінальний кодекс України (ст. 163, 182) URL: https://zakon.rada.gov.ua/laws/show/2341-14#Text
8. Bhandari A. et al. Packet Sniffing and Network Traffic Analysis Using TCP-A New Approach. Advances in Electronics, Communication and Computing. Springer, Singapore, 2018. P. 93–101.
9. Nakov S., Kolev V. Fundamentals of Computer Programming with C#. Sofia: Bulgarian Academy of Sciences, Faber Publishing, 2013. 1132 p. URL: https://www.csharp-book.com/

10. NDIS_FILTER_ATTACH_PARAMETERS structure (ndis.h) URL: https://learn.microsoft.com/en-us/windows-hardware/drivers/ddi/ndis/ns-ndis-_ndis_filter_attach_parameters

11. Al-Bastami B.G., Abu Naser S.S. Design and Development of an Intelligent Tutoring System for C# Language. International Journal of Advanced Research in Computer Science and Software Engineering. 2017. Vol. 7, Issue 2. P. 1–8.

12. Perkins B., Hammer J.V., Reid J.D. C# 7 Programming with Visual Studio 2017. Wiley, 2018. 912 p.

13. Clark D. Beginning C# Object-Oriented Programming. 2nd Edition. Apress, 2013. 384 p.

14. Halvorsen H.-P. Introduction to Visual Studio and C#. – University College of Southeast Norway, 2016. 86 p. URL: https://www.halvorsen.blog/documents/teaching/materials/

15. NIST Special Publication 800-53 Rev. 5. Security and Privacy Controls for Information Systems and Organizations. Gaithersburg: National Institute of Standards and Technology, 2020 (updated 2023). 492 с. URL: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r5.pdf

16. Горбань О.В. Інформаційна безпека комп’ютерних систем і мереж: навчальний посібник. Київ: НАУ, 2022. 368 с.

17. Шевченко А.Ю., Кравченко О.М. Захист інформації в комп’ютерних мережах: підручник. Харків: ХНУРЕ, 2023. 412 с.

18. Stallings W. Cryptography and Network Security: Principles and Practice. 8th Edition. Pearson, 2020. 752 с.

19. Conklin W.A., White G.B. Principles of Computer Security: CompTIA Security+ and Beyond. 6th Edition. New York: McGraw-Hill Education, 2023. 976 с.

20. Ліпінський В.В. Кібербезпека та захист інформації: підручник. Львів: ЛНУ імені Івана Франка, 2024. 480 с.