

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра комп'ютерних наук

Бакалаврська дипломна робота на тему:

РОЗРОБКА ПРОГРАМНОГО МОДУЛЯ ПРОКСІ ФЕРМИ
З ДИНАМІЧНОЮ ЗМІНОЮ IP-АДРЕСИ

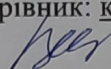
Виконав: студент (4) курсу, групи 22КН-мс
спеціальності 122 – Комп'ютерні науки

(шифр / назва напрямку підготовки, спеціальності)

 Богомазов А.А.

(прізвище та ініціали)

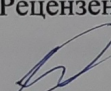
Керівник: к.т.н., доцент каф.КН

 Денисюк В.О.

(прізвище та ініціали)

« 07 » Червня 2024 р.

Рецензент: к.т.н, доц. каф. САІТ

 Козачко О.М.

(прізвище та ініціали)

« 07 » Червня 2024 р.

Допущено до захисту

Зав. кафедри Ірвинь А.А. 

« 07 » Червня 2024 р.

Вінниця ВНТУ - 2024 рік

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра комп'ютерних наук
Рівень вищої освіти перший (бакалаврський)
Галузь знань – 12 Інформаційні технології
Спеціальність – 122 Комп'ютерні науки
Освітньо-професійна програма – Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри КН

проф., д.т.н. Яровий А.А.

« 07 » 03 2024 р.

**ЗАВДАННЯ
НА БАКАЛАВРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ**
Богомазову Артуру Анатолійовичу

1. Тема роботи: Розробка програмного модуля проксі ферми з динамічною зміною IP-адреси.

Керівник роботи: Денисюк Віталій Олександрович, к.т.н., доцент
затверджені наказом вищого навчального закладу “ 11 ” 03 2024 року № 80

2. Строк подання студентом роботи 27.05.2024р.

3. Вихідні дані до роботи :

мова програмування – об'єктно-орієнтована;
середовище розробки - Visual Studio.

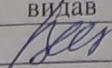
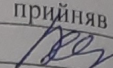
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):

- обґрунтування доцільності розробки модуля проксі ферми з динамічною зміною IP-адреси;
- аналіз переваг та недоліків існуючих проксі;
- створення структури програмного модуля проксі ферми з динамічною зміною IP-адреси;
- розробка алгоритму взаємодії користувача з додатком;
- визначити предмет, об'єкт, мету та задачі подальшого дослідження.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень):

- схема алгоритму роботи модуля для ідентифікації кольорових відтінків;
- діаграми класів та компонентів програмного забезпечення;
- загальний вигляд інтерфейсу користувача;
- приклад роботи програми.

6. Консультанти розділів проекту (роботи)

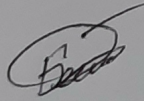
Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1-3	Денисюк В.О., к.т.н., доцент		

7. Дата видачі завдання 07.03.2024р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва та зміст етапу	Термін виконання		Примітка
		початок	закінчення	
	Аналіз предметної області задачі розробки програмного модуля проксі ферми з динамічною зміною IP-адреси	06.05.2024	09.05.2024	
	Розробка програмного модуля проксі ферми з динамічною зміною IP-адреси	10.05.2024	14.05.2024	
	Проектування програмного модуля проксі ферми з динамічною зміною IP-адреси	15.05.2024	21.05.2024	
	Реалізація програмного модуля проксі ферми з динамічною зміною IP-адреси	22.05.2024	29.05.2024	
	Розробка інструкції користувача	30.05.2024	03.06.2024	
	Оформлення матеріалів до захисту БКР	04.06.2024	06.06.2024	

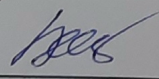
Студент


(підпис)

Богомазов А.А.

(прізвище та ініціали)

Керівник роботи


(підпис)

Денисюк В.О.

(прізвище та ініціали)

АНОТАЦІЯ

Бакалаврська дипломна робота складається з 81 сторінок формату А4, на яких є 26 рисунків, список використаних джерел містить 12 найменувань.

У роботі розглянуто розробку програмного модуля проксі ферми з динамічною зміною IP-адреси. Розв'язано питання аналізу галузі застосування проксі ферм, відомих методів розв'язання задач використання проксі ферми, обґрунтовано засоби розробки програмного модуля, розроблено діаграми класів програмного модулю, обґрунтовано вибір програмного засобу реалізації, створено програмну реалізацію та проведено тестування програмної реалізації модуля проксі ферми з динамічною зміною IP-адреси. Власна реалізація програмного модуля дозволяє збільшити швидкодію, продуктивність, надійність, зменшити витрат на експлуатацію та спростити керування проксі фермою з динамічною зміною IP-адреси за рахунок уникнення використання зовнішніх модулів та простої організації власного модуля. Одержані результати можливо використовувати сторонніми користувачами проксі ферм з динамічною зміною IP-адреси.

Ключові слова: мобільна проксі ферма, IP-адреса, смартфон, HTTP, 4G.

ABSTRACT

The bachelor thesis consists of 81 pages of A4 format, on which there are 26 figures, the list of used sources contains 12 names.

The paper deals with the development of a software module of a proxy farm with a dynamic change of the IP address. The issue of analysis of the branch of proxy farm implementation, known methods of solving the problems of proxy farm use was solved, the tools for developing the software module were justified, the class diagrams of the software module were developed, the choice of the software implementation tool was justified, the software implementation was created and the software implementation of the proxy farm module was tested with dynamic change of the IP address. Own implementation of the software module allows you to increase speed, performance, reliability, reduce operating costs and simplify the management of a proxy farm with a dynamic change of IP address by avoiding the use of external modules and simply organizing your own module. The obtained results can be used by third-party users of proxy farms with a dynamic change of IP address.

Keywords: mobile proxy farm, IP address, smartphone, HTTP, 4G.

ЗМІСТ

ВСТУП.....	3
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ВИКОРИСТАННЯ ПРОКСІ ФЕРМИ З ДИНАМІЧНОЮ ЗМІНОЮ ІР-АДРЕСИ	6
1.1 Аналіз галузі застосування проксі ферм	6
1.2 Аналіз відомих методів розв’язання задач використання проксі ферми	16
1.3 Постановка задачі.....	18
1.4 Висновок до розділу 1	21
2 РОЗРОБКА ПРОГРАМНОГО МОДУЛЯ ПРОКСІ ФЕРМИ З ДИНАМІЧНОЮ ЗМІНОЮ ІР-АДРЕСИ.....	22
2.1 Обґрунтування засобів розробки програмного модуля.....	22
2.2 Розробка діаграми класів.....	29
2.3 Розробка UML діаграми послідовності роботи проксі ферми	31
2.4 Висновок до розділу 2	33
3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО МОДУЛЯ ПРОКСІ ФЕРМИ З ДИНАМІЧНОЮ ЗМІНОЮ ІР-АДРЕСИ.....	36
3.1 Обґрунтування вибору програмного засобу реалізації.....	36
3.2 Створення програмної реалізації.....	50
3.3 Тестування програмної реалізації.....	60
3.4 Висновок до розділу 3.....	63
ВИСНОВКИ.....	64
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	66
ДОДАТОК А ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ.....	68
ДОДАТОК Б ІНСТРУКЦІЯ КОРИСТУВАЧА	69
ДОДАТОК В ЛІСТИНГ ПРОГРАМИ.....	72
ДОДАТОК Г ГРАФІЧНА ЧАСТИНА.....	76

ВСТУП

Актуальність теми дослідження полягає у тому, що мобільні проксі ферми стали невід'ємною частиною різних галузей людської діяльності, надаючи унікальні можливості для заробітку, маркетингу, спілкуванню тощо. Завдяки мобільним проксі фермам користувачі переміщуються в Інтернеті за допомогою мобільної IP-адреси. Відмінною особливістю мобільних мереж є те, що IP-адреса мобільного зв'язку є довіреною та може бути легко змінена, оскільки доступний великий пул цих надійних IP-адрес. В електронній комерції мобільні проксі можуть допомогти керувати кількома обліковими записами та збільшити кількість покупок. Користувачі можуть заробляти, керуючи кількома онлайн-магазинами та оптимізуючи їхні процеси, зокрема навчаючись купувати проксі-сервери електронної комерції, що може підвищити рівень конверсії та прибутки. Мобільні проксі забезпечують анонімність і унікальність IP-адрес, що важливо в афілійованому маркетингу. Користувачі можуть заробляти, спрямовуючи трафік на вигідні партнерські пропозиції, використовуючи кілька унікальних IP-адрес, щоб збільшити конверсію та дохід. Цільова аудиторія тут – це ті, хто розміщує рекламу у Facebook, Google, TikTok та інших соціальних мережах. В індустрії азартних ігор мобільні проксі дозволяють користувачам обійти обмеження на кількість облікових записів на одному сайті ставок. Це відкриває можливість шукати бонуси на сайті та заробляти на цьому гроші за допомогою проксі-сервера для ставок на спорт. Наприклад, користувачі можуть зареєструвати кілька акаунтів, на яких вони будуть використовувати безкоштовні ставки. У світі соціальних мереж мобільні проксі дозволяють керувати кількома обліковими записами з різних місць. Користувачі можуть заробляти, просуваючи свої бренди, надаючи маркетингові послуги в соціальних мережах або беручи участь у партнерських програмах. Мобільні проксі ідеально підходять для збирання даних із веб-сайтів, особливо якщо використовуються проксі для

веб-збирання. Користувачі можуть створювати та продавати бази даних, аналізувати ринки та надавати дані компаніям для прийняття бізнес-рішень. У кожній із цих галузей мобільні проксі-сервери відкривають двері для нових можливостей заробляння грошей і забезпечують анонімність і безпеку в Інтернеті. Використання мобільних проксі у ніші потокового передавання гарантує, що творці вмісту та глядачі можуть обходити геообмеження та безперешкодно отримувати доступ до вмісту за допомогою проксі-сервера потокового відео. Ці проксі надають справжні IP-адреси, що зменшує ймовірність виявлення та блокування. Основною метою використання мобільних проксі-серверів у цьому секторі є забезпечення безперебійного доступу до потокового вмісту незалежно від регіональних обмежень. Використання проксі-серверів для сканування SERP (сторінка результатів пошукової системи) дозволяє ефективно збирати дані рейтингу пошукових систем без ініціювання заборон або обмежень швидкості. Проксі забезпечують анонімні та різноманітні точки доступу, полегшуючи збір великих обсягів точної інформації SERP із різних географічних місць. Основною метою використання проксі-серверів для SERP є моніторинг і аналіз рейтингів веб-сайтів у різних пошукових системах без збоїв, а також увімкнення аналізу ключових слів у результатах пошуку.

Мета роботи - це підвищення швидкодії, збільшення продуктивності, підвищення надійності, зменшення витрат та спрощення керування проксі фермою з динамічною зміною IP-адреси.

Завдання роботи полягають у розв'язанні таких питань:

- аналіз галузі застосування проксі ферм;
- аналіз відомих методів розв'язання задач використання проксі ферми;
- обґрунтування засобів розробки програмного модуля;
- розробка діаграми класів;
- обґрунтування вибору програмного засобу реалізації;

- створення програмної реалізації;
- тестування програмної реалізації.

Об'єкт дослідження - процес створення програмного модуля проксі ферми з динамічною зміною IP-адреси

Предмет дослідження - алгоритми та засоби програмного модуля проксі ферми з динамічною зміною IP-адреси

Практична цінність одержаних результатів полягає в можливості використання іншими розробниками та користувачами проксі ферми з динамічною зміною IP-адреси.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ВИКОРИСТАННЯ ПРОКСІ ФЕРМИ З ДИНАМІЧНОЮ ЗМІНОЮ IP-АДРЕСИ

1.1 Аналіз галузі застосування проксі ферм

Мобільна проксі ферма (проксі-ферма, ферма мобільних проксі) - це звичайна мережа пристроїв мобільного зв'язку, яка використовується в арбітражі трафіку для виконання завдань, пов'язаних із обробкою та маніпулюванням трафіку.

Ферма мобільних проксі складається з великої кількості пристроїв, на кшталт смартфонів або планшетів. Їх застосовують для створення оригінальних IP-адрес і імітації реальної поведінки користувачів у мережі.

Основні задачі, що розв'язують ферми мобільних проксі, такі.

1. Генерація унікальних IP-адрес: кожен мобільний пристрій у фермі має свою власну IP-адресу, яка може бути використана при вході в мережу.
2. Маскування та зміна IP-адрес: ферма мобільних проксі дозволяє маскувати та змінювати IP-адреси пристроїв, що допомагає обходити обмеження та блокування на сторонніх сайтах та сервісах.
3. Емуляція реальної поведінки користувачів: айпишки, що генеруються, можуть імітувати різні дії, на кшталт відвідування веб-сайтів, кліки, перегляди відео та інші дії, щоб створити враження нормальної користувача поведінки.

Ферми мобільних проксі часто використовуються в арбітражі трафіку для отримання доступу до сайтів, які заборонені державами, а також щоб уникнути банів від деяких джерел трафіку за численні акаунти та для більш ефективного керування трафіком загалом.

Арбітраж трафіку (Traffic arbitrage) — це процес купівлі трафіку на одному ресурсі з подальшим перепродажем на іншому задля отримання

прибутку. Суть методу полягає в тому, щоб купити рекламний трафік за низькою ціною та перепродати його з вищою маржинальністю [1].

У сучасному інтернет-просторі з'являється дедалі більше можливостей для заробітку. Арбітраж трафіку є однією з них і відзначається високою ефективністю, особливо за правильного підходу.

Основні характеристики арбітражу трафіку.

1. Арбітраж трафіку дає можливість заробляти гроші без необхідності створення власного продукту, контенту або навіть вебсайту. Задача полягає у тому, щоб ефективно купувати рекламний трафік на одному майданчику і перепродавати його на іншому з вищою ціною.
2. Цей метод заробітку є дуже гнучким. Можливо обирати різні майданчики, типи реклам, аудиторії, а також працювати в різних географічних локаціях. Це дає змогу швидко адаптуватися до змін на ринку і оптимізувати свою стратегію.
3. Арбітраж трафіку дає змогу фокусуватися на аналітиці і оптимізації, замість того, щоб витратити час на створення контенту або керування продуктом. Потрібно лише добре розуміти, як аналізувати дані та як коригувати свої кампанії для досягнення максимального рівня ROI (повернення інвестицій).
4. Арбітраж трафіку не вимагає великих інвестицій на старті. Зазвичай потрібен лише доступ до рекламних майданчиків і невеликий бюджет для перших купівель трафіку.

Все це робить арбітраж трафіку цікавою і потенційно прибутковою можливістю для тих, хто хоче заробляти в інтернеті.

Типи арбітражу: білий, сірий, чорний.

Арбітраж трафіку має кілька відмінностей від інших заробіткових стратегій. Зокрема він не вимагає створення власного контенту, товарів або послуг. Головна задача — купити трафік за низькою ціною і продати його дорожче.

Білий, сірий та чорний арбітражі мають ключові відмінності.

Білий арбітраж. Все здійснюється чесно і прозоро. Купується трафік і направляється на легітимні сайти. Такий підхід є найбезпечнішим, але і може забезпечити нижчий прибуток.

Сірий арбітраж. Тут може бути використано різноманітні хитрощі, як-от прихований контент або реклама. Це ще може бути ефективно, але збільшує ризик санкцій з боку рекламних платформ.

Чорний арбітраж. Цей метод включає в себе нелегітимні дії, як-от фрод, спам або маніпуляції з пошуковими системами, і може призвести до блокування акаунтів.

Арбітраж трафіку для початківців. Перший крок – це вибір конкретної ніші або “вертикалі”, в якій працюватиме. Для цього необхідно провести глибокий аналіз ринку, вивчити основних конкурентів, а також популярні рекламні майданчики. Це допоможе зрозуміти, які можливості для заробітку існують в обраній сфері.

Найкращі інструменти для старту. Після того, як визначилися із нішею, наступний крок – це вибір інструментів для роботи. Серед найпопулярніших інструментів можна виділити Google Analytics для аналізу трафіку, Semrush для конкурентного аналізу, а також спеціалізовані платформи для арбітражу, як-от Voluum чи RedTrack. Ці платформи допоможуть ефективно керувати рекламними кампаніями, відслідковувати конверсії та аналізувати ефективність реклами.

Перші кроки у налаштуванні проксі-ферми. Після вибору інструментів прийшов час налаштувати першу рекламну кампанію. На цьому етапі важливо вирішити, з яких джерел краще купувати трафік. Чи це буде пошуковий трафік із Google, соціальний трафік із Facebook, чи, можливо, спеціалізовані партнерські мережі. Після визначення джерела трафіку потрібно буде налаштувати параметри таргетингу: географію, вікові рамки,

інтереси користувачів тощо. Це допоможе зосередити рекламу на цільовій аудиторії і збільшити шанси на високий рівень ROI (повернення інвестицій).

Отже, почати заробляти на арбітражі трафіку можна, маючи гарний аналіз ринку, правильний вибір інструментів та ефективно налаштування рекламних кампаній. При відстеженні змін на ринку та адаптації стратегії успіх не змусить себе довго чекати.

Огляд популярних вертикалей. В арбітражі трафіку є безліч напрямків, які можна вибрати для заробітку. Ось декілька популярних вертикалей.

1. E-commerce. Ця вертикаль досить популярна, зокрема через великий обсяг ринку і множинність підніш, які можна обрати.
2. Ігри. Сегмент мобільних і комп'ютерних ігор надзвичайно широкий. І тут можна не тільки приводити гравців на платформи, але й залучати їх в купування внутрішньоігрових активів.
3. Фінансові послуги. Кредитування, страхування, інвестиції – все це напрямки, де є високі ставки і, відповідно, великі можливості для заробітку.
4. Туризм. Хоча ця вертикаль постраждала під час пандемії, вона повільно відновлюється, і тут є багато можливостей для заробітку.

При обиранні вертикалі дуже важливо зрозуміти кілька ключових факторів:

1. Наявні знання та навички. Обирайте треба ту нішу, у якій є достатній досвід або хоча б базові знання.
2. Рівень конкуренції. Деякі ніши дуже конкурентні, і буде важко тут просунутися без значущих інвестицій в рекламу і оптимізацію.
3. Цільова аудиторія. Розуміння цільової аудиторії може бути ключовим фактором успіху. Потрібно знати, що потрібно, і як аудиторія досягає своїх потреб.
4. Бюджет. Деякі вертикалі можуть вимагати більших витрат на старт, тоді як інші можуть бути менш коштовними, але і менш прибутковими.

5. ROI (Повернення інвестицій). Необхідно оцінити потенційний ROI в різних вертикалях, базуючись на проведених дослідженнях і прогнозах.

Рекомендується звернути увагу на те, в якій ніші є більше розуміння найкраще, а також на рівень конкуренції в даній ніші. Так можливо зробити обґрунтованіший вибір і збільшити шанси на успіх.

Важливі арбітражні терміни для розуміння світу інтернет-маркетингу наведені у додатку Д.

Для побудови ферми застосовують не лише мобільні термінали, а й модеми. Крім того, створення власної проксі-ферми не є складним завданням, з ним може впоратися будь-хто, хто хоча б раз у житті користувався інтернетом. Актуальність використання проксі-фермне зменшиться з роками, бо арбітражний світ процвітає, а разом із ним збільшується потреба у нових та якісних проксі-фермах.

Цікавим є питання заробітку на проксі-фермах. У будь-якому бізнесі заробіток залежить від кількості продажів. А скільки можна отримувати з однієї IP-адреси? Наприклад, на європейському ринку оренда проксі коштує від \$30 на місяць. У США готові купувати від \$50 за штуку. А в Україні прибуток починається з \$20 за помісячну оренду мобільного проксі.

Збільшення об'єму продажу IP-адрес збільшує заробіток. Але все залежить від кількості телефонів, які є для створення IP-адрес. Наприклад, для середнього заробітку до \$1000 вистачить і невеликого приміщення (власної квартири, маленького офіса). А якщо більше, тоді потрібне місце для ферми масштабніше, наприклад, гаража або міні-складу може навіть вантажівка.

Крім того, бажано мати помічника, який забере частину роботи на себе, аби була можливість збільшити масштаби проекту. Крім технічних деталей, для високих заробітків потрібні маркетингові навички. Також важливо обрати відповідне географічне розташування ферми.

Створити проксі ферму можна або за допомогою модемів, або за допомогою мобільних телефонів.

Для налаштування проксі ферми з використанням USB-модемів у якості обладнання знадобляться: модеми, SIM-карти, USB-хаб, віртуальний виділений сервер або фізичний сервер.

Модеми. Рекомендується використовувати розлучені типи модемів. Одним із найпрактичніших варіантів є використання бренду Хуавей.

SIM-карти. Необхідно використовувати SIM-карти різних мобільних операторів, які надають безперебійний інтернет зі швидкістю 100 Мбіт/с.

USB-хаб. Повинен мати мінімальну кількість портів для початку побудови ферми - 10, протягом часу цю кількість пропорційно треба збільшувати.

Віртуальний виділений сервер (VDS) або фізичний сервер повинен мати не менше ніж 4 ГБ вільної ОЗП (для підтримки 10 портів).

Ситуацію, коли віртуальний сервер надається користувачу, як послуга, називається виділенням віртуального виділеного сервера.

Така послуга має велику низку плюсів у порівнянні з використанням сервера еликого виділеного фізичного сервера, а саме [2] :

- витрати на фінансування зменшуються,
- відмови не такі часті,
- налагодження та керування спрощуються.

Для налаштування проксі ферми з використанням мобільних телефонів у якості обладнання потрібні: смартфони, SIM-карти, USB-хаб, віртуальний виділений сервер або фізичний сервер.

Смартфони повинні підтримувати систему Android версії 5.0 і новіші. Важливо мати додаткові батареї для смартфонів, так як вони періодично виходять із ладу через сильне навантаження або обладнання для контролю режимів роботи батарей смартфонів.

Способи запуску ферми. Варіант з мобільними телефонами найпростіший. Для початку знадобиться 5-10 смартфонів із SIM-карти, а також хаб для підключення та доступу до Інтернету. Необхідно також зареєструватися на будь-якому з доступних сервісів, що надають можливість створення мобільних проксі.

На рис.1.1 надано приклади постачальників мобільних проксі (Bright Data, SmartProxy, ProxySale) [3, 4].

Критерії вибору сервісів.

1. ГЕО. Не всі послуги покривають широку географію. Звертайте увагу на обмеження.
2. Наявність пробного періоду. Бажано вибирати сервіс із пробником, щоб оцінити його можливості та протестувати чи підходить він за всіма параметрами.
3. Ліцензія або передплата. Програма може пропонувати тарифний план підписки або придбання ліцензії на використання.
4. Ціна. Є послуги безкоштовні, і ті, що пропонують тарифні плани. Який вибір залежить від кінцевої мети.

Також при виборі сервісів необхідно звертати увагу на такі параметри.

1. Швидкість підключення - варто використовувати мобільні проксі з 4G і LTE, тому що при них з'єднання встановлюється з високою швидкістю, а користувач здається реальнішим.
2. Кількість IP-адрес - чим вона більша, тим краще. Якщо сервіс надає великий пул IP, ви зможете обирати різних операторів зв'язку, ГЕО.
3. Процес ротації адрес - найчастіше послуги пропонують автоматичну зміну через певний проміжок часу. Зручніше, якщо є можливість проводити цю процедуру на запит.

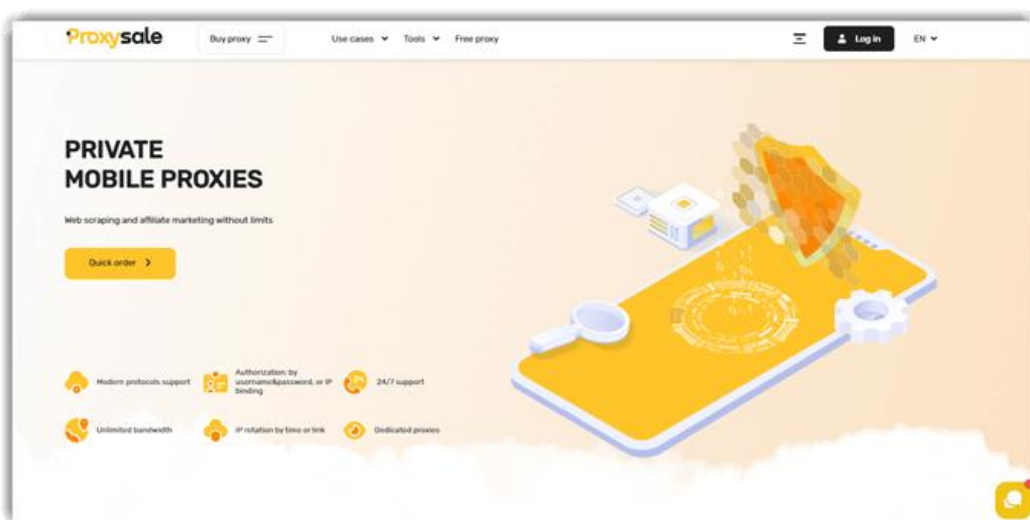
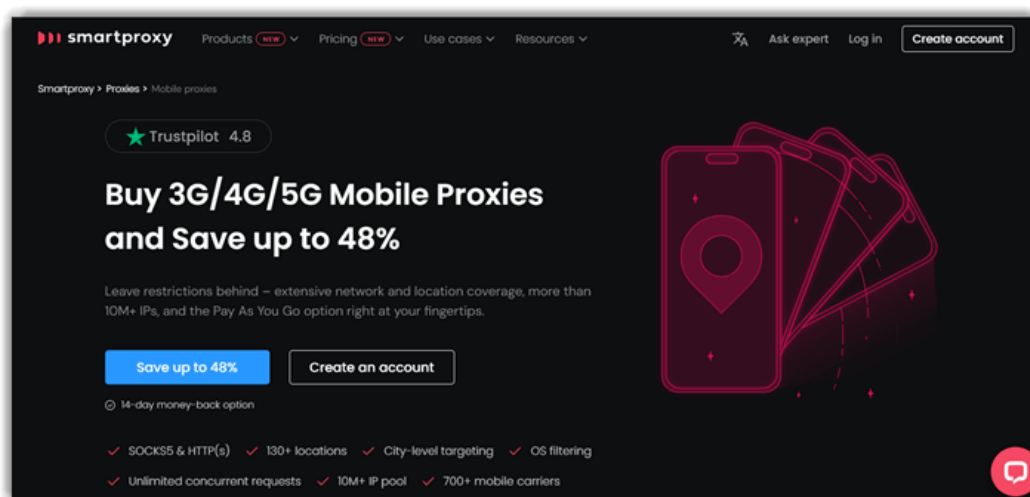
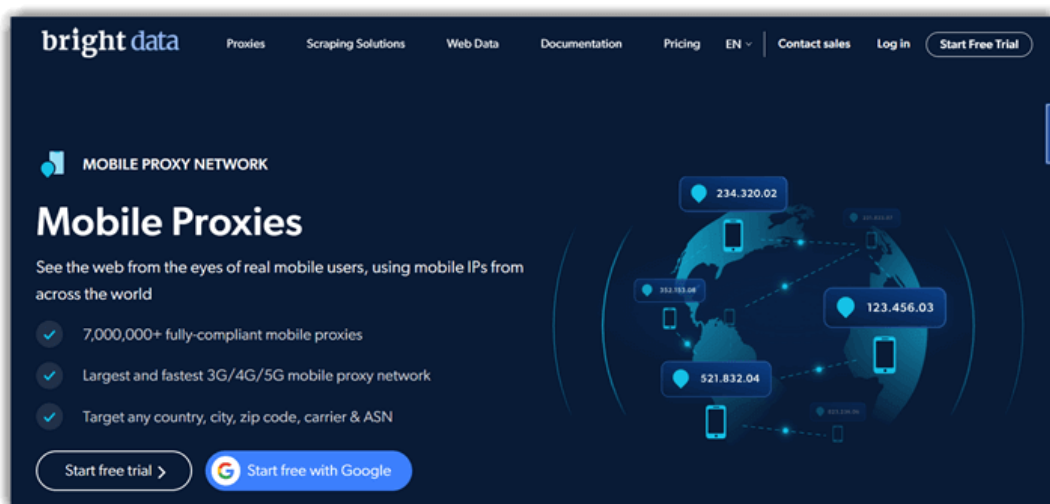


Рисунок 1.1 – Приклади постачальників мобільних проксі
(Bright Data, SmartProxy, ProxySale)

4. Приватність - загальні канали дешевші, але дії інших користувачів можуть негативно вплинути на роботу. Краще звертатися до сервісів, які пропонують приватні канали, вони більше підходять для арбітражу.
5. Вартість - надто дешеві пропозиції обмежені за обсягом трафіку, ГЕО та швидкості. Необхідно знайти оптимальний варіант. Можливо користуючись пробними періодами або можливістю сплатити за 1 день.
6. Швидкість - ефективний провайдер проксі має мінімально уповільнювати підключення до мережі. Працювати із малою швидкістю буде незручно.

Враховуючи ці фактори, обираються варіанти з необхідним ГЕО та оператором зв'язку. Це дозволить максимально імітувати поведінку реального користувача для пошукових систем, соціальних мереж.

Смартфони повинні бути вже з правами root, інакше розширити проксі неможливо. При цьому не доведеться вдаватися в дрібні деталі, оскільки всі ці сервіси надають чіткі інструкції, які допоможуть виконати всі необхідні налаштування.

Android – відкрита система, але також має низку обмежень, поставлених Google. Якщо є потреба мати розширені можливості керування пристроєм, то необхідно інстальовати на смартфон root [5].

Види root-прав.

1. Root Temporary – це права, які надаються на короткий час та втрачаються при перезавантаженні Android.
2. Shell root – постійні права, в яких, тим не менш, відсутня можливість вносити зміни в розділ system
3. Full root – це повний або найбільша доступ до системного розділу Android.

Можливості, які надає root такі.

1. Можна заморозити і відновити прошивку.
2. Є можливість отримати додатки без реклами.
3. Можна позбутися аудіофайлів і додатків, вбудованих виробником.
4. Можливо самостійно вбудувати медіадодатки, наприклад, аудіофайли, але неможливо буде видалити такі файли у статусі Гостя.
5. Зміна зовнішнього вигляду операційної системи (заміна дизайну кожної іконки на свій смак тощо), редагувати функціонал додатків.
6. Одного разу заряджений акумулятор буде працювати в півтора-два рази довше, але загальний час експлуатації акумулятора значно зменшиться.
7. Збільшення і зниження тактової частоти процесора (може обернутися меншим терміном його служби).
8. Розблокування графічного ключа.
9. Створення власної прошивки.

Root спочатку не вбудовується на Android тому, що:

- зміна системної інформації привести до фізичного зламу пристроя (якщо збільшити частоту роботи центрального процесора, то він дуже швидко вийде з ладу);
- відмова у додатках від реклами завдяки використанню root зменшує прибутки Google.

Негативні сторони root-прав.

1. Треба буде оновлювати операційну систему самостійно, бо офіційні права будуть втрачені.
2. Відсутність фірмових переваг та особливостей певної оболонки.
3. Миттєва втрата гарантії, оскільки Google не підтримується root.

4. Треба буде відмовитися від використання програм-антивірусів (root погано співпрацює з ними).

1.2 Аналіз відомих методів розв'язання задач використання проксі ферми

Арбітраж трафіку вимагає парсингу акаунтів, фармінгу, обходу блокувань та повної анонімності.

Арбітражники, які працюють з Facebook або Google Adwords, знають, як швидко банить система за порушення правил. На практиці доходить до того, що після кількох санкцій створити нові облікові записи, або ж залогінитися в проплачених стає неможливим.

Щоб уникнути цих проблем, важливо використовувати proxies.

Мобільні проксі замінюють реальну IP-адресу на адресу стільникового оператора. Принцип базується на тому, що коли на сервер надходить безліч запитів, заблокувати їх неможливо. Це не спам, а дії з пристроїв, прив'язаних до однієї IP-адреси. Арбітражники можуть легко «загубитися» серед інших користувачів, зробивши мобільні проксі самому. Постійні зміни даних створять ефект «присутності реального користувача», що дозволить обійти алгоритми платформ.

По суті, мобільні проксі - те саме, що і стандартні, з тією лише різницею, що з'єднання з мережею здійснюється через стільникового оператора. Тобто. стандартні проксі дозволяють користувачеві отримати анонімність, а динамічні мобільні проксі показують, що він вийшов в Інтернет через мобільний оператор. Це особливо актуально для соцмереж, які більшість користувачів відвідують через мобільні девайси. А якщо говорити про Instagram, то туди із десктопних пристроїв практично не заходять.

Основна особливість цього рішення та аргумент на користь того, щоб купити мобільний проксі: IP-адреси головних володарів стільникового

бізнесу використовуються як проксі вкрай рідко. У них дуже мало проксі-адрес – на кілька млн. абонентів їх припадає в кращому разі кілька тисяч. Тому блокувати подібні IP-адреси для соціальних мереж не вигідно – санкції блокування торкнулися б не лише тих, хто використовує динамічні мобільні проксі, а й користувачів звичайних проксі серверів. Більше того, насамперед саме їх вони й торкнуться – заблокують користувачів, які взагалі ні в чому не винні. Тому соцмережі ніяк не працюють з IP-адресами, що належать ключовим мобільним операторам. І навіть якщо санкції будуть застосовані в поодиноких випадках, довіра у IP-адрес оператора стільникового зв'язку відновлюється швидко. Достатньо купити дешевий проксі і правильно побудувати його роботу, і можна знизити кількість банів, пов'язаних з використанням підозрілих IP, практично до нуля.

Переваги мобільних проксі.

1. Приховують IP-адресу. Це гарантує безпеку у мережі.
2. Повна анонімність. Жодні дані не передаються, що захищає обліковий запис на 100%.
3. Немає обмежень щодо трафіку. Використовуйте у тій кількості, в якій необхідно.
4. Динамічна зміна IP-адреси. Ця функція дозволяє працювати одночасно з декількома обліковими записами. Алгоритм самостійно змінює дані згодом.
5. Низький ризик бана. Жодній платформі не вигідно накладати обмеження, за яких постраждають тисячі користувачів.

Для чого потрібні проксі ферми арбітражнику.

1. Реєстрація кількох облікових записів, коли система не запідозрить умисне збільшення кількості облікових записів.
2. Накрутка. Збільшити кількість лайків та переглядів тепер не проблема, обмеження можна обійти, зробивши мобільні проксі самому.

3. Парсинг – можливість безпечно зібрати дані, імітуючи при цьому діяльність зі смартфона.
4. Фарм акаунтів та запуск реклами з мультиаккаунта. Допомогає оминати обмеження. Читайте більше про проксі для фармінгу.

Переваги власних мобільних проксі ферм.

1. Є можливість контролю трафіку.
2. Економія. Є кілька способів зробити мобільні проксі, ціна коливається в залежності від обраного методу. Сам метод залежить від завдань та необхідної кількості айпу на добу. Підйом власних IP-адрес обійдеться вдвічі дешевше за орендовані.
3. Мобільні проксі можна зробити під конкретну геолокацію (ГЕО) проксі-ферми.
4. Як додатковий плюс – за допомогою власної проксі-ферми можна додатково заробляти, здаючи їх у найм.

Арбітражникам ферма може стати в нагоді не тільки для фарма акаунтів, але і для здачі в оренду за ціною, нижчою від ринкової.

1.3 Постановка задачі

Проксі-сервер діє як посередник між користувачем та інтернет-ресурсом. Процес працює так: користувач відправляє запит через проксі, який зі свого боку перенаправляє його на цільовий сайт. Після отримання відповіді від сайту, проксі пересилає її назад користувачеві. Основна функція проксі – зміна IP-адреси, наприклад, користувач перебуває в Києві, але через проксі може здаватися, ніби він у Польщі. IP-адресу можна порівняти з поштовою адресою, тільки в рамках інтернету.

Інтервал зміни IP варіюється від 2 хвилин до 12 годин, або можна керувати через API.

Проксі-сервери корисні для обходу антифрод-систем, доступу до сайтів, заблокованих у певних географічних регіонах, а також для запобігання блокувань під час запуску рекламних кампаній. Існують різні типи проксі, і без заглиблення в технічні деталі, головний критерій – це кількість користувачів на один проксі. Для джерел інтернет-трафіку рідко важливий протокол, але якщо буде виявлено підозрілу активність з вашого IP, відбудеться блокування. Наприклад, для реклами в Facebook, TikTok і Google підходять тільки резидентні (домашні провайдери) і мобільні проксі (стільникові оператори), в той час як серверні (хостинг) IP швидко призведуть до блокування, хоча і підходять для парсингу даних.

Найпростіша власна ферма мобільних проксі - це створення мобільних проксі на смартфонах, формування своєрідної мережі IP. Створюються на смартфонах на Android. З цього методу можна починати новачкам. Знадобиться Android 5.0 та сервіс (наприклад, такі як на рис.1.1). Сам процес простий - реєстрація в системі та виконання запропонованих кроків. Все це займає трохи більше 10 хвилин. Потрібно тільки натискати на «далі» і система самостійно налаштується.

IP-адреси змінюватимуться за допомогою «Авіарежиму» двома способами:

- 1) вручну (включити, вимкнути);
- 2) віддалено через root права.

Для цього потрібно таке обладнання (рис.1.2).

1. Смартфони (5-10 штук) на одну ферму. Ціна залежить від того де купувати. Це необов'язково мають бути нові телефони, підійдуть і старі робочі мобільники. Рекомендуємо віддавати телефон на рутування фахівцям, або купувати вже готовий до використання, спеціально прошитий гаджет.
2. USB хаб великого розміру.
3. SIM-карти.

4. Розумна розетка.

5. Доступ до сервісу для створення проксей.

Мобільні проксі вигідніші за інших за такими ознаками:

- підвищений рівень довіри від сервісів,
- доступ до всіх IP-адрес пула та можливість змінювати ГЕО та оператора,
- змінити IP-адресу за посиланням (API) або таймером,
- повністю безлімітний трафік,
- можливість використання одного проксі для багатьох облікових записів,
- підтримка протоколів HTTP,
- налаштовуваний таймер на зміну ір адреси,
- доступ до сотень тисяч IP-адрес.

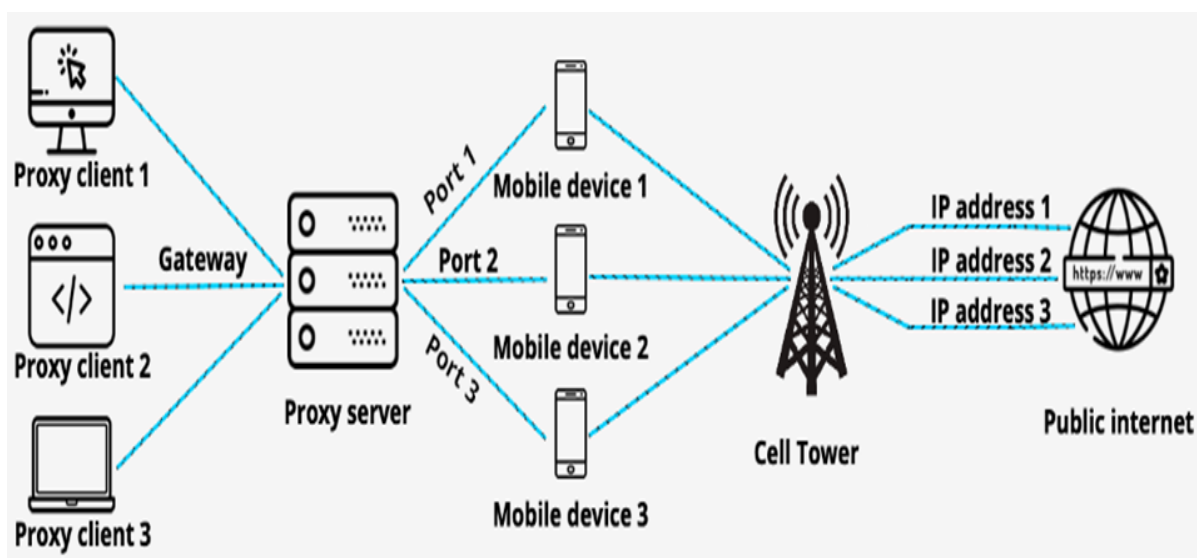


Рисунок 1.2 – Структура проксі ферми з динамічною зміною IP-адреси

Щоб користуватися 4G, смартфони та SIM карти мають підтримувати технологію LTE [6].

1.4 Висновок до розділу 1

Аналіз сфери використання проксі ферм підтвердив переваги власних мобільних проксі ферм, які дозволяють використовувати їх у розв'язанні багатьох задачах сучасних інформаційних технологій. Аналіз проблем, які пов'язані із задачами створення та використання проксі ферми, актуалізує питання розробки найпростішої власної мобільних проксі ферми на смартфонах, що дозволяє формування мережі з динамічною зміною IP-адреси.

2 РОЗРОБКА ПРОГРАМНОГО МОДУЛЯ ПРОКСІ-ФЕРМИ З ДИНАМІЧНОЮ ЗМІНОЮ ІР-АДРЕСИ

2.1 Обґрунтування засобів розробки програмного модуля

Підтримка смартфоном стандарту 4G є необхідною умовою їх використання у мобільній фермі.

На рисунку 2.1 надано структуру мережі, що працює за 4G-стандартом.

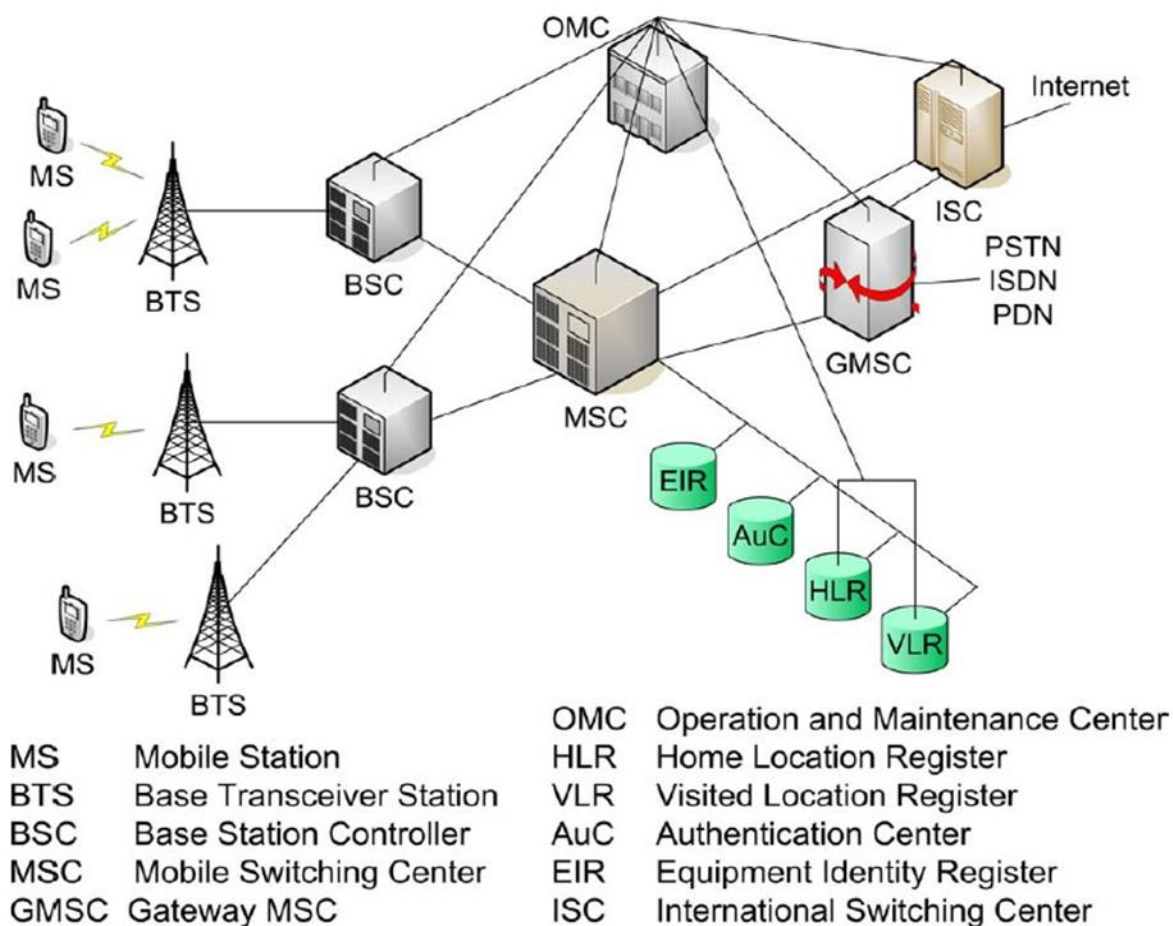


Рисунок 2.1 – 4G/GSM структура

SWG (Serving Gateway) – це LTE-мережевий шлюз обслуговування [7, 8]. Його функції – це отримання та перенаправлення даних у пакетах від станцій підсистеми.

Для усунення випадків відмови від обслуговування або спотворення передачі сигналів, а також необхідність підтримки роботоспроможності функціонування операторів зі своїми мережами як 4G , так і 3G, SGW сполучається напряду.

Пакетна передача комутує голосові та інформаційні передачі у LTE.

Для пришвидшення роботи, оптимізації циклу життя, експлуатації на кожному кроці API використовується Postman.

В широкому сенсі значення – Postman представляє платформу API.

Шлюзовання потоків даних множиною операторів мобільного зв'язку покладено на PGW (PDN Gateway).

Пов'язання різноманітних мереж GSM або Internet у задачах маршрутизації потоків LTE покладено на PGW.

З ціллю забезпечення безперешкодного переміщення користувачів мобільного зв'язку у середині мережі використовується керуючий центр MME (mobility management).

Абонентські дані зберігаються у призначеному для цього великому сховищі – абонентському сервері, - HSS (home subscriber server).

Також HSS виконує завдання по автентифікації, протоколів шифрування та створення усієї необхідної проміжної інформації.

Кількість абонентів, складність структури мережі зумовлює кількість наявних HNS.

Для контролю і управлінню питаннями якості мобільного зв'язку та вартості наданих послуг, згідно обраного користувачем пакету послуг, використовується компонента PCRF (policy and charging rules function).

Типи інформаційної передачі.

Різні типи каналів використовуються для передачі різних типів даних за різними інтерфейсами.

Така організація дозволяє фільтрувати дані за відповідним типом та робе передавання даних набагато продуктивнішим.

Таке каналне розбиття сегрегує різноманітні дані та уможливорює використання значно продуктивніших протоколів обміну.

У середовищі передачі виділяють такі основні типи рівнів.

1. Фізичний рівень виконує передачу даних та службової інформації, яка адресована абонентам. Ця інформація різниці в залежності від того, куди вона надається каналом (всередину чи назовні).
2. Транспортний рівень використовується для передачі даних у МАС за допомогою каналів попереднього рівня. Фізичний рівень за допомогою радіопередачі розташовує оброблені за допомогою MAC/RLC дані від логічного рівня. Основні показники роботи радіоінтерфеса за параметрами передачі даних визначаються транспортним рівнем. Блоки транспортування представляють собою основні компоненти, на які поділяються дані передавання. Для цього цикл передачі поділяється на інтервали передавання – TTI (transmission time interval). Під час такого інтервала повинен передаватися не менше ніж один блок транспортування. Існує можливість передачі і 4-х блоків транспортування за допомогою алгоритмів MIMO.
3. Логічний рівень використовується для сервісу на проміжному керуючому рівні – МАС (medium access control). Підрівень трафіку та підрівень керування є основними компонентами логічного рівня.

Розглянемо основні параметри LTE.

1. Використання при пакетній комутації радіо інтерфейсу
2. Убезпечення роботи до двохсот абонентів на кожному стільнику, який працює на 5 МГц.
3. Заощадження електричної потужності, яка використовується.
4. Підвищена мобільність абонентів.
5. Часовий та частотний каналний поділ
6. Частотна смуга передачі [1,4 ... 20] МГц.
7. Зменшена до 5 мс затримка.
8. Вихідна швидкість – 75 Мб за секунду.
9. Завантажувальна швидкість – 300 Мб за секунду.

Цікаво розглянути інформацію про advanced-структуру LTE (рис.2.2)

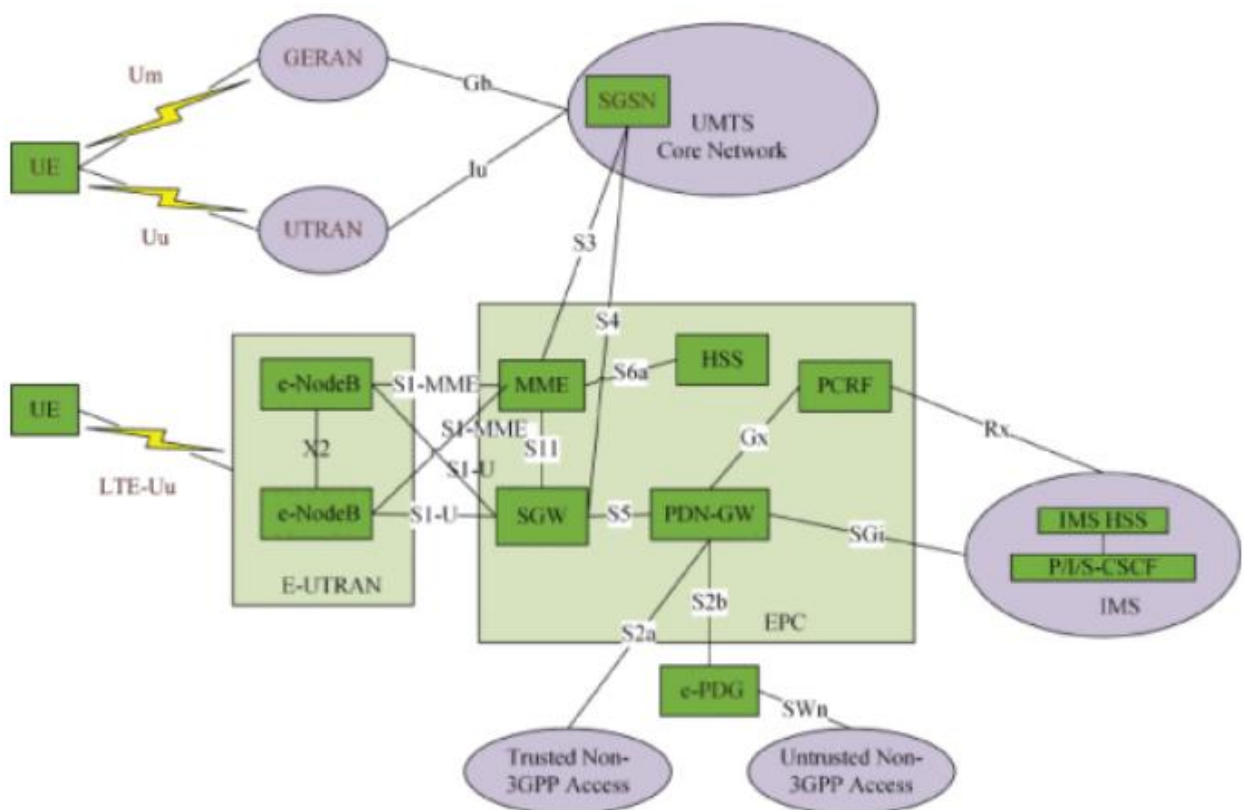


Рисунок 2.2 - Advanced-структура LTE

Базова компонента та компонента радіодоступу є основними компонентами мережі.

- E-UTRAN – це компонента радіодоступу.
- EPC – це базова компонента.

Можливість підключення різних мереж завдячує базовій компоненті.

Процес автентифікації сотового абонента використовує такі компоненти:

- MME – модуль керування мобільністю,
- HSS – сервер даних домашніх даних,
- PGW - шлюз пакетної передачі.

Автентифікацію стільникових пристроїв необхідно робити аби запобігти використанню зламаних пристроїв чи таких, що здобуті злочинним шляхом.

MME-компонента починає процес автентифікації сотового пристрою (рис.2.3).

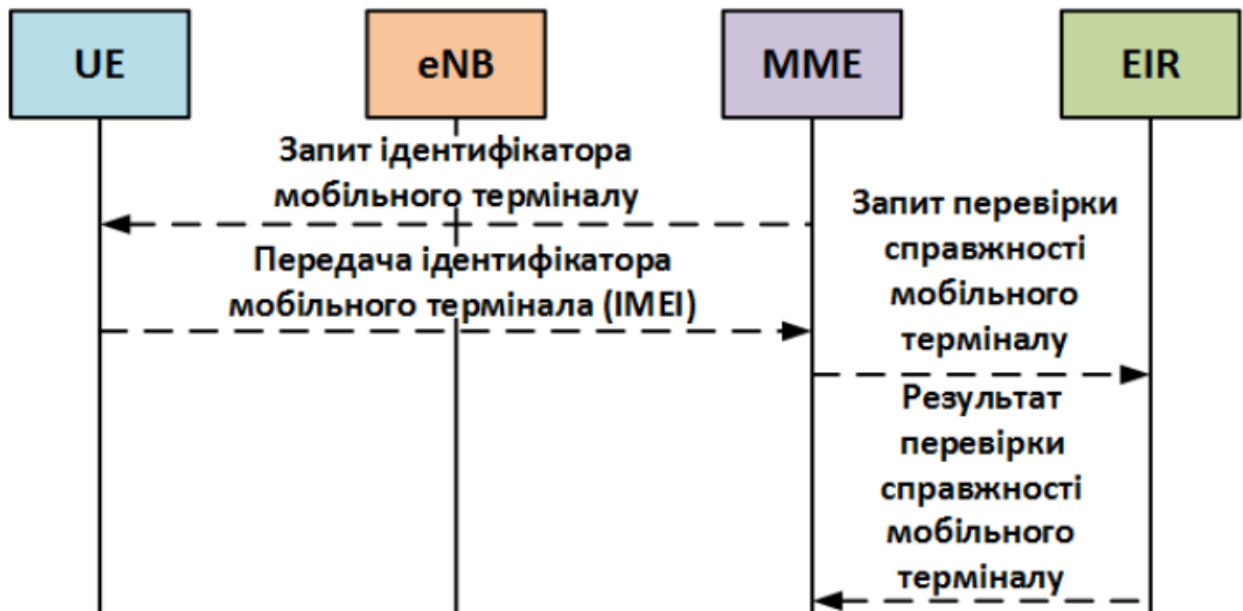


Рисунок 2.3 – Процедура автентифікації мобільного термінала

Основні особливості використання IP у стільникових пристроях.

Особливості використання IP пов'язані із особливостями інтернету у мобільних мережах.

Кожний мобільний оператор зв'язку має справу з множиною своїх користувачів. Тому неможливо технічно надати кожному з них свій IP. Крім того, оператору стільникового зв'язку це не має сенсу з економічних міркувань [10].

Тому було використано таке технічне рішення, при якому визначається набір IP, що одночасно надається множині користувачів.

Отже, можлива ситуація, при якій велика кількість користувачів одночасно користуються інтернетом та мають однакові IP.

Якщо ж зробити блокування деякого IP, який використовує недоброчесний абонент, то можливо заблокувати і велику кількість доброчесних користувачів інтернету.

На такий крок оператори стільникового зв'язку та соціальні мережі ніколи не підуть.

Посередник, який пов'язує абонента і користувача інтернету, представляє собою складний апаратно-програмний комплекс та отримав назву – мобільний проксі.

Оператори стільникового зв'язку виконують обмін інформацією між абонентами мережі використовуючи веб-додатки.

Обмін даними між абонентами виконується відповідно через стільникового оператора за допомогою IP, які обираються самим оператором мобільного зв'язку.

Мобільний інтернет, який використовує стільниковий зв'язок, надає низку переваг, як і сам мобільний зв'язок.

Мобільні проксі мають характерні особливості.

Існує безліч способів створення таких проксі.

Для побудови мобільних проксі кожна фірма або сервіс можуть мати свої підходи та рішення.

Самим надійним, простим та ефективним підходом є підхід, при якому трафік перенаправляється на сервера внутрішньої структури оператора з стільникових карток абонентів (рис.2.4).

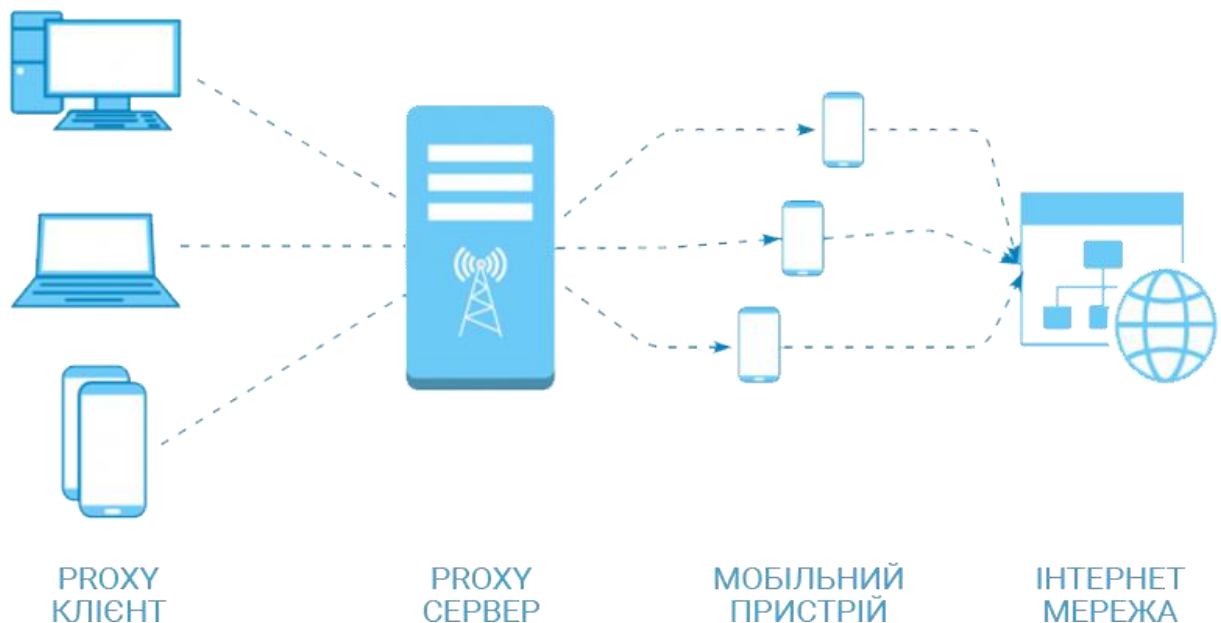


Рисунок 2.4 – Перенаправлення трафіку

Експлуатація конкретних стільникових пристроїв з конкретною картою у мережі стільникового зв'язку є великою перевагою такого варіанту використання мережі.

Відбувається ідентифікація та реєстрація пристрою абонента за його IMEI у вузлі стільникового оператора.

За цим кроком відбувається інформаційна передача різноманітних запитів у стільниковому інтернеті під управлінням сервера між стільниковими абонентами.

Сервери підтримують різноманітні протоколи роботи самих серверів – HTTP, HTTPS, SOCKS, SOCKS5. Створюють різноманітні набори стільникових IP. На деяких каналах обміну керують перепускною

спроможністю цього каналу. Добре справляються з управлінням системи уцілому.

2.2 Розробка діаграми класів

В деяких особливостях звичайні проксі відрізняються від мобільних.

Пересічний громадянин, фізособа не може мати IP, які належать традиційним серверам, які в свою чергу узяли їх у провайдерів або у хмарах.

Зменшення довіри до аканту користувача чи блокування – такі дії виконуються алгоритмами звичайного проксі.

Приклад роботи проксі надано на рисунку 2.5.

Важливо розуміти, коли використовувати проксі. Коли потрібно використати спрощену версію складного або важкого предмета.

У цьому випадку ми можемо представити його за допомогою об'єкта-скелета, який завантажує вихідний об'єкт на вимогу, що також називається відкладеною ініціалізацією. Це відоме як віртуальний проксі.

Коли вихідний об'єкт присутній в іншому адресному просторі, і ми хочемо представити його локально.

Ми можемо створити проксі-сервер, який виконує всі необхідні шаблонні речі, такі як створення та підтримка з'єднання, кодування, декодування тощо, у той час як клієнт отримує доступ до нього так, як він присутній у його локальному адресному просторі.

Це називається віддаленим проксі-сервером.

Коли ми хочемо додати рівень безпеки до початкового основного об'єкта, щоб забезпечити контрольований доступ на основі прав доступу клієнта. Це називається проксі-сервером захисту

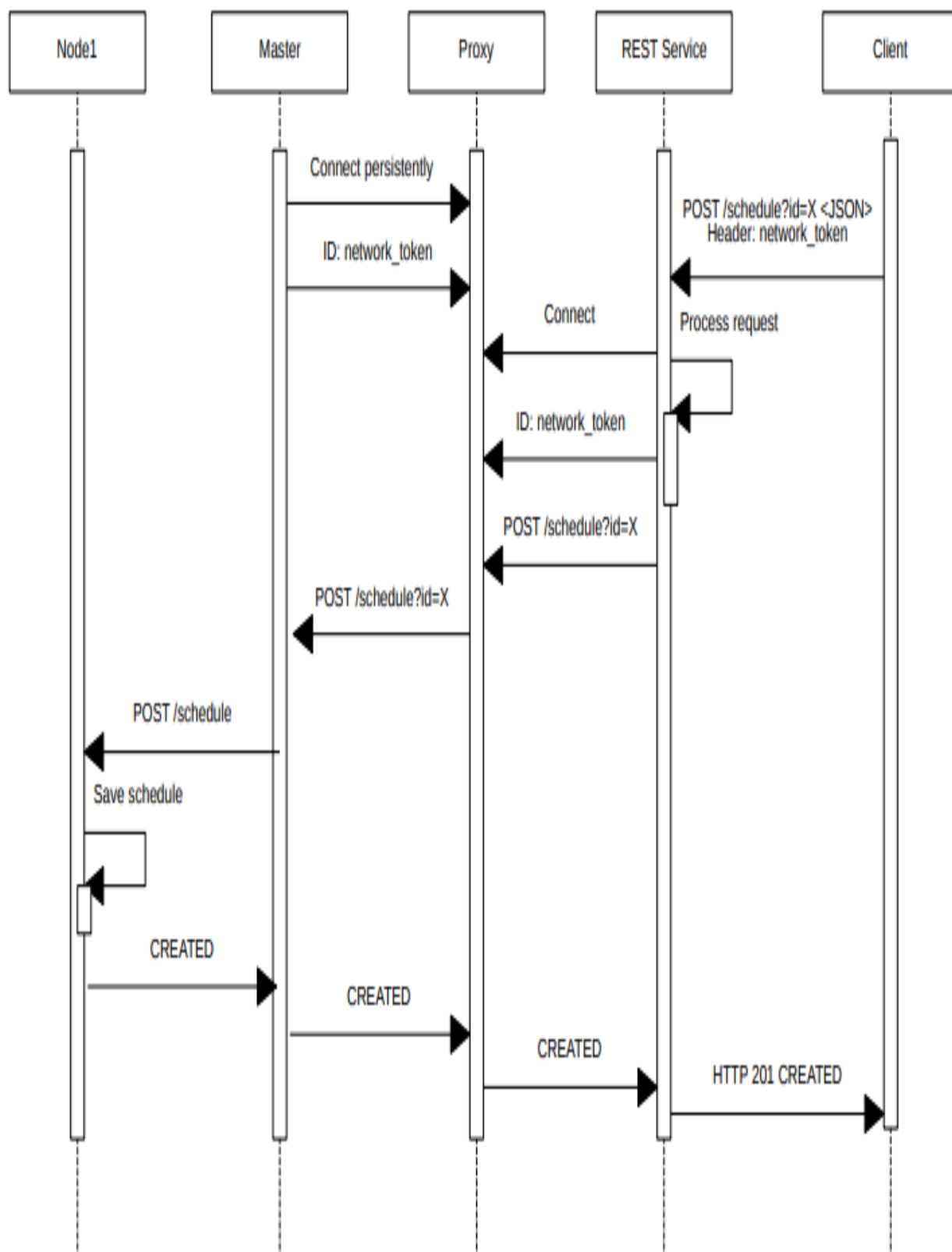


Рисунок 2.5 – Робота проксі

2.3 Розробка UML діаграми послідовності роботи проксі ферми

Є відмінності у використанні традиційних проксі та мобільних.

Якщо використовувати традиційний проксі-сервер, то як правило, при збільшеній активності використання IP, у 90 випадках із 100 цей IP стає підозрілим і великі ресурси за своїми алгоритмами розпізнавання відправляють IP у бан. Це пов'язано з тим, що традиційні сервери працюють з IP великих хмар або провайдерів, які в свою чергу окремим фізособам такі IP не надають.

Блокування IP відбувається повне чи за рахунок зменшенням трафіку до нього.

У стільникового оператора у його наборі є декілька сотень адрес, які він і роздає своїм мільйонам абонентам одночасно

Таким чином, блокування однієї адреси може призвести до блокування багатьох абонентів, які її використовують. Така ситуація є неприпустимою для усіх, наприклад, соцмереж.

Широковикористовуємий та відомий мобільний проксі – це HTTP-проксі. Оснований на протокольному стеку TCP. Використовує визначені порти 3128, 8080, та 80.

Функціонування відбувається завдяки відправленню запиту на завантаження у браузер визначеного ресурсу за його URL.

На абонентський браузер надходить інформація через сервер із шуканого ресурсу.

Особливості роботи HTTP-проксі:

- дані фільтруються,
- пришвидшення завантаження сайтів за раунок файлового кешування,
- можливість обійти обмеження доступу до деяких ресурсів.

Захищений проксі – це HTTPS. Схожий на HTTP, але використовує поєднання захищеного типу з забезпеченням SSL.

Широко використовується у випадках передачі засекреченої чи конфіденційної інформації – номери, паролі, логіни тощо.

Використовує визначені порти 3128, 8080, та 80.

Нажаль, існує ситуація, коли на нижньому рівні чи з використанням самого сервера є можливість заволодіти інформацією, коли у нас HTTP.

Так, уся активність абонента у мережі може бути перехоплена провайдером.

Провайдером для доведення дій користувача у мережі ця інформація зберігається.

Для стійкого шифрування трафіку відповідними алгоритмами і забезпечення даних фізичної особи треба використовувати HTTPS.

Трафік користувача при з'єднанні з ресурсом шифрується – так працює HTTPS. Неможливо використати сервер як фільтр даних бо немає змоги визначити тип переданої інформації. Причому, сервер не приймає участі у дешифруванні чи шифруванні. Браузер є тим засобом, який виконує цю роботу.

Отже, сервер тільки передає зашифровані дані без їх обробки. Його можливо використовувати з різними протоколами стеку TCP – NNTP, IMAP, SMTP, POP3, тощо.

Протокол Sock-Proxu на тепер є протоколом інформаційної передачі з найбільшим поширенням. Якщо програма або додаток безпосередньо не використовують проксі, то можна використати Socks.

Вперше описаний протокол Socks на тепер широко не використовується. Йому на заміну розроблені Socks 4 та Socks 5:

- Використання логіну та паролю для авторизації, коннект з використанням UDP та TCP – Socks 5;

- Коннект з використанням TCP – Socks 4.

HTTP у заголовках не підлягають модерації за допомогою Socks. Анонімність та незмінність даних, що передаються – це риси, які притаманні усім Socks.

Але інформація про IP користувача буде передана. Визначення самого факту використання проксі з боку браузера є неможливим. Браузер бачить тільки безпосередню роботу та прозоре сполучення. Реальний IP користувача веб-сайт не побачить. Замість нього – проксі IP.

Протоколи FTP, HTTPS, HTTP та інші підтримуються Socks [11, 12]

Принцип роботи мобільного проксі надано на рисунку 2.6.

На рисунку 2.7 надано приклад обробки запиту GET, яке було надіслано веб-клієнтом.

2.4 Висновок до розділу 2

Використання мобільних проксі ферм з динамічною зміною IP-адрес дозволяє значно збільшити можливості використання сучасних інформаційних технологій. У результаті дослідження з'ясовано специфіку, напрямки та засоби побудови програмного модулю розробки. Розроблені UML діаграми послідовності роботи проксі ферми.

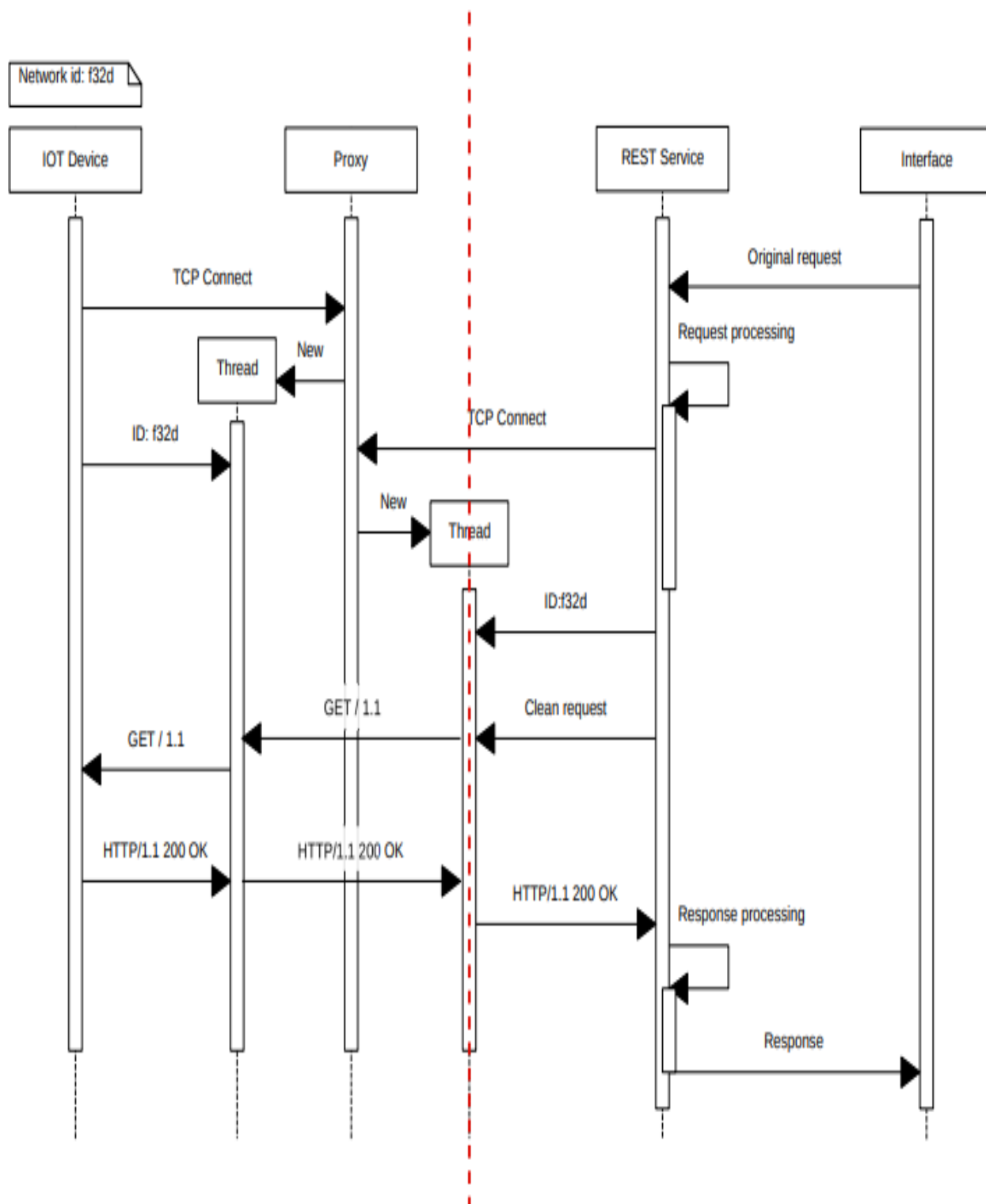


Рисунок 2.6 – Діаграма послідовності роботи мобільного проксі

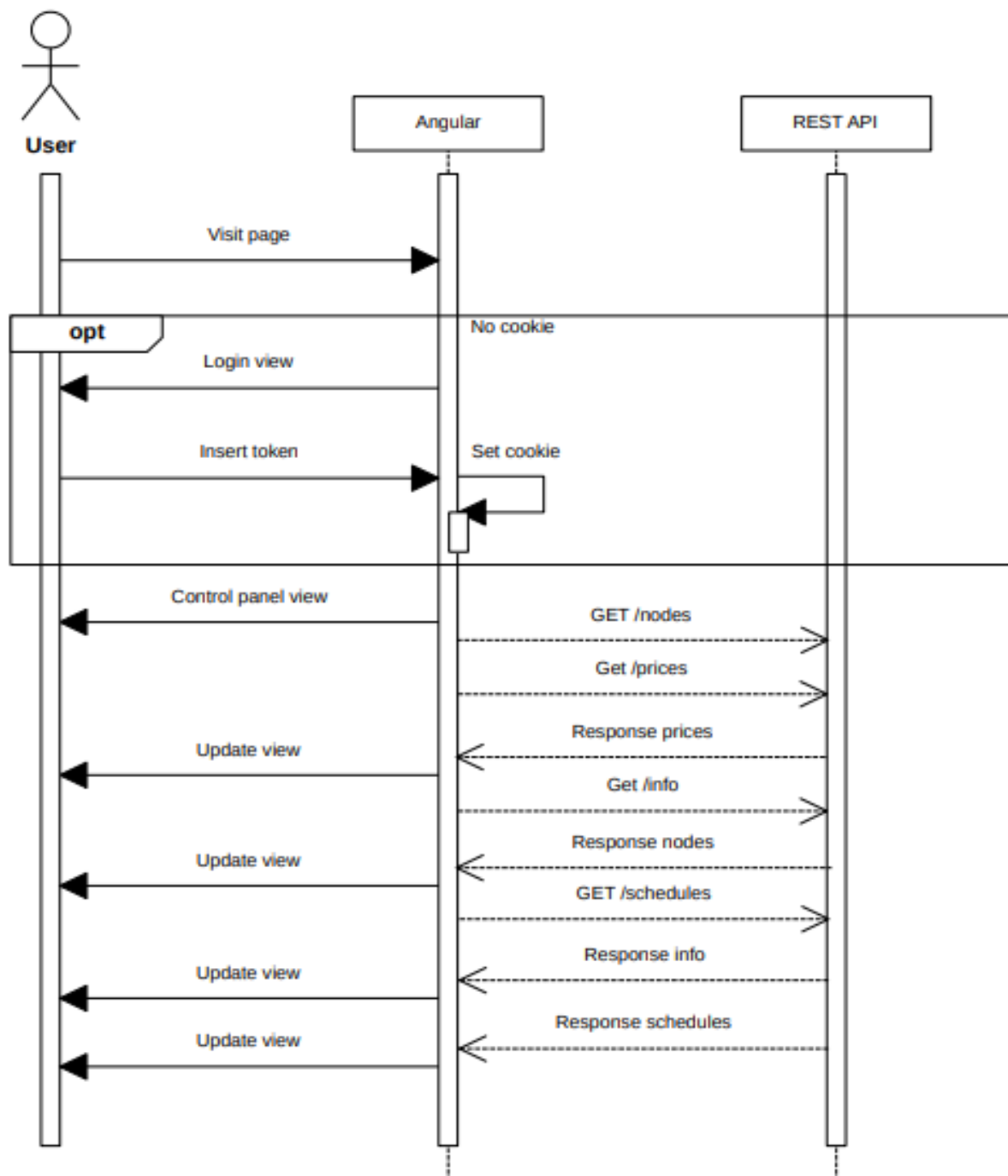


Рисунок 2.7 - Діаграма послідовності обробки веб-додатків

3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО МОДУЛЯ ПРОКСІ ФЕРМИ З ДИНАМІЧНОЮ ЗМІНОЮ IP-АДРЕСИ

3.1 Обґрунтування вибору програмного засобу реалізації

Вибір найкращої мови для проекту розробки програми є критичним бізнес-рішенням, яке потребує глибоких технічних міркувань і досвіду [9]. Мова програмування, якаобирається, може суттєво вплинути на успіх вашого проекту, витрати, швидкість розробки та довгострокову підтримку.

Відома статистика десятки найкращих мов розробки додатків станом на 2023 рік (рис.3.1) [9].



Рисунок 3.1 - Десятки найкращих мов розробки додатків на 2023 рік

Деякі з цих мов переважно використовуються для розробки комерційних веб-сайтів, тоді як інші використовуються для розробки нестандартних мобільних додатків. Проте є ті, які можна використовувати для різних доменів, наприклад .NET, який підходить для створення програмних продуктів на різних платформах.

Цікавим є питання про найкращу мову для розробки мобільних додатків. Популярні мови програмування зазвичай використовуються в різних ситуаціях, але секрет успішного та надійного рішення полягає в добре розробленій комбінації загальних технологій.

Важливість вибору мови програмування. Вибір між найкращими мовами розробки додатків схожий на закладення фундаменту будівлі. Вибір, який ви зробите, може поставити вас на шлях успіху або призвести до непотрібних проблем. Розберемося, чому це рішення є вирішальним.

Важливість вибору мови:

- ефективність, правильна мова може оптимізувати розробку, зменшивши час і зусилля;
- сумісність, деякі мови краще підходять для певних платформ;
- масштабованість, довгостроковий успіх часто залежить від масштабованості мови;
- експертиза розробника, обізнаність вашої команди з мовою є важливою;
- підтримка громади, широке використання сприяє міцній спільноті розробників.

Вибираючи мову програмування для проекту розробки програми необхідно враховувати такі фактори: вимоги до проекту, експертиза розробника, вибір платформи.

1. Вимоги до проекту:

- функціональність. які функції та можливості потрібні;
- продуктивність, швидкість чи ефективність вирішальні для програми
- платформа, програма призначена для певної операційної системи чи кількох платформ;
- інтеграція, чи потрібно програмі підключатися до інших систем або API.

2. Експертиза розробника

- командні навички, якими мовами володіють розробники (або які спеціалізовані команди програмування додатків можна найняти);
- крива навчання, чи готові розробники витратити час на вивчення нової мови, якщо це відкриває світ технологічних можливостей;
- доступність найму, чи є пул досвідчених розробників для вибраної мови.

3. Вибір платформи

Вибір між платформною розробкою, яка пропонує повторне використання коду на кількох платформах, або власною розробкою, яка забезпечує оптимізовану продуктивність. Це може стати основним фільтром, обмежуючи створення програми за допомогою рідної мови розробки програми. Однак певні комбінації серверних операцій, як-от розробка PHP, все ж можуть знадобитися.

Розглянемо поширені мови програмування [9]. Ось перелік деяких мов програмування, які часто використовуються в розробці програм:

- JavaScript, широко використовується для розробки веб-додатків і мобільних додатків;
- Java, домінує в розробці програм для Android;
- Swift, основна мова для розробки програм для iOS;
- Python, відомий своєю універсальністю та простотою використання;
- Kotlin, набирає популярності як альтернатива Java для Android;
- PHP, відомий розробкою веб-додатків і сценарієм на стороні сервера;
- Ruby, відзначається своїм елегантним синтаксисом і розробкою веб-додатків.

JavaScript. Інтерфейсна основа веб-розробки, часто вважають найкращою мовою розробки програм. Він вдихає динамізм у веб-програми для електронної комерції та фінтех, покращуючи взаємодію з користувачами та естетику. Його надійні функції безпеки протидіють шахрайству та зміцнюють код. У поєднанні з ReactJS він сприяє широкомасштабній розробці додатків для електронної комерції, банківської справи та фінансових технологій, забезпечуючи як безпеку, так і швидкість. Завдяки цьому ваша високопродуктивна програма займає перше місце в цифровій досконалості, а NodeJS вміло керує серверними операціями.

Python, універсальна мова розробки програм, працює на таких відомих платформах, як Reddit і Uber. Його кросплатформні можливості забезпечують безперебійну роботу в різних системах і пристроях. Підтримка Python багатопоточності покращує керування одночасними запитами та загальну продуктивність. Його чистий і зручний синтаксис зручний для початківців, що робить його чудовим вибором для новачків у програмуванні. Надійна підтримка спільноти надає велику кількість ресурсів для вирішення запитів у процесі розробки програми.

Java. Незважаючи на свою репутацію однієї з найкращих мов розробки мобільних додатків, Java поширює свій вплив за межі розробки мобільних банківських додатків і динамічної індустрії фінансових технологій. Його легкий і гнучкий характер робить його ідеальним вибором для веб-додатків у різних областях. З історією, що налічує понад чверть століття, Java є основоположною технологією в багатьох секторах, використовуючи свої надійні функції безпеки та ефективну обробку великих наборів даних. Архітектура Java «напиши один раз, запусти де завгодно» є ключовим фактором її тривалого успіху, що забезпечує безперебійну роботу на різноманітних платформах. У галузях, де існує різноманітне апаратне забезпечення та операційні системи, ця портативність є надзвичайно важливою.

RНР — це універсальна мова сценаріїв, яка широко використовується у веб-розробці, відома своєю здатністю створювати динамічні та інтерактивні веб-сторінки. Він може похвалитися широким набором фреймворків і бібліотек, що дозволяє розробникам створювати різноманітний набір веб-додатків. RНР дозволяє створювати безпечні та багатофункціональні веб-програми для сектору фінансових технологій. Його зручний характер і сумісність з різними базами даних роблять його найкращим вибором серед веб-розробників у всьому світі.

Swift, розроблений в екосистемі Apple, є мультипарадигмальною скомпільованою мовою програмування, яка славиться простотою вивчення та використання, а також надійною безпекою. Її вважають найкращою мовою розробки додатків, яка процвітає в сфері iOS, забезпечуючи виняткову продуктивність і процвітаючу спільноту розробників, забезпечуючи світле майбутнє для розробки додатків на платформах Apple. Його природа з відкритим вихідним кодом сприяє інноваціям і співпраці між розробниками, ще більше розширюючи його можливості.

Kotlin є однією з мов розробки мобільних додатків для Android. Він пропонує широкий набір технічних можливостей. Маючи значний резерв талантів, він надає широкі можливості для розширення знань. Його повна інтеграція з Java підвищує ефективність розробки коду додатків Android, забезпечуючи легку взаємодію між двома мовами.

Включення функції відкладеного завантаження оптимізує продуктивність шляхом вибіркового завантаження необхідних компонентів інтерфейсу користувача. Крім того, використання Kotlin класів даних спрощує кодування, зменшуючи багатослівність, зберігаючи функціональність.

Ruby - це об'єктно-орієнтована, динамічна та рефлексивна мова програмування з різноманітними додатками. Розробники можуть використовувати RubyMotion, оснащений унікальним компілятором, для

створення власних додатків як для iOS, так і для Android, забезпечуючи сумісність між платформами.

Крім того, структура Ruby on Rails, побудована на основі Ruby, є популярним вибором для створення програм, сумісних з Android та iOS, пропонуючи розробникам ефективний набір інструментів для розробки мобільних програм.

Кросплатформна розробка.

Гібридні програми, як випливає з їх назви, створені для безперебійної роботи в операційних системах iOS і Android, забезпечуючи роботу користувача без збоїв. Основні мови, які використовуються для створення цих програм, включають HTML, JavaScript, ReactJS, Python тощо. Оскільки ці додатки розроблені як кросплатформні, код має бути написаний на фреймворку з рушієм, здатним перекладати фрагменти коду в власні елементи, адаптовані до конкретної ОС (рис.3.2).

Advantages of Hybrid Apps	Disadvantages of Hybrid Apps
Operate on multiple platforms simultaneously	Features can be complex, making defect identification challenging
Require minimal maintenance	May need additional security features to meet OS-specific requirements
No specific programming language expertise required	Access to device-specific features may be restricted.
Easily integrable with web-based services	Familiarity with cross-platform frameworks is necessary.
Faster development and cost-efficiency	

Рисунок 3.2 - Переваги та недоліки гібридного програмування

Нативний розвиток (рис.3.3). Нативна програма спеціально створена для однієї платформи, як-от Android або iOS, із використанням відмінних функцій і можливостей цієї ОС.

Ці програми розроблено з використанням мов розробки програм, які залежать від платформи та використовують унікальне кодування операційної системи.

Advantages of Native Apps	Disadvantages of Native App
Optimized Performance	Higher Costs
Offline Capability	Cross-Platform Challenges
Easier Development	Skill Requirements
Enhanced User Experience	Longer Development Time
Simplified Code	

Рисунок 3.3 - Переваги та недоліки нативного програмування

Цей підхід дає змогу рідним додаткам розкрити весь потенціал пристрою, плавно поєднуючи високоякісні оптимізовані технології з новими інноваціями, такими як розробка мобільних додатків у хмарі, порівняння веб-додатків та інтеграція GPS. Вибір нативних мов розробки програм залежить від обраної платформи.Android (Java/Kotlin). Рідні програми для Android переважно використовують Java і Kotlin. Kotlin, відомий своєю стислістю та ясністю, останніми роками отримав все більше поширення.iOS (Swift/Objective-C). Власні програми для iOS часто створюються за

допомогою Swift, сучасної мови вибору. Однак Objective-C продовжує керувати застарілими програмами.

Розглянемо сильні та слабкі сторони наведених мов програмування (рис.3.4).

Плюси і мінуси JavaScript. Повсюдне поширення JavaScript у веб-розробці та розробці мобільних додатків у поєднанні з його багатою бібліотечною екосистемою робить його чудовим вибором для розробки гібридних додатків.

JavaScript	Ubiquitous for web and mobile app development.	Slower than native languages, potential compatibility issues.
Java	Rich toolset, robust security, suitable for finance.	Learning curve, verbosity, not the best choice for beginners.
Swift	Prioritizes performance, native iOS development.	Limited to iOS, smaller community.
Python	Versatile, readable, suitable for data and web dev.	Not ideal for high-performance applications, GIL limits threads.
Kotlin	Streamlined, gaining popularity in Android.	Primarily for Android, smaller developer base.
PHP	Versatile for web app development, extensive framework support, seamless database integration.	Limited scope, performance concerns, occasional security concerns.
Ruby	Elegant and readable syntax.	Performance limitations, smaller community, steep learning curve.

Рисунок 3.4 - Переваги та недоліки мов програмування

Однак він може демонструвати повільнішу продуктивність порівняно з рідними мовами, і потенційні проблеми сумісності через наявність кількох бібліотек і фреймворків потребують розгляду.

Переваги:

- широковідомий, використовується для розробки веб- і мобільних додатків;

- величезна екосистема, велика кількість бібліотек і фреймворків;
- кросплатформенність, ідеально підходить для розробки гібридних програм.

Недоліки:

- продуктивність, може бути повільнішим порівняно з рідними мовами;
- фрагментація, кілька бібліотек і фреймворків можуть призвести до проблем із сумісністю.

Плюси і мінуси Java. Структурована екосистема Java надає розробникам різноманітні інструменти та легку інтеграцію сторонніх розробників, що прискорює розробку програм. Завдяки надійним вбудованим функціям безпеки, таким як обмеження часу виконання та комплексний менеджер безпеки, Java забезпечує безпеку коду.

Його надійні, зручні API безпеки ефективно виявляють і ізолюють підозрілий код і віртуальні машини, що робить його кращим вибором для розробки торгових алгоритмів у кількісних фінансових і банківських програмах.

Переваги:

- домінант на Android, ідеально підходить для розробки програм для Android;
- стабільність, зріла мова з сильною екосистемою;
- продуктивність, рідні програми, розроблені на Java, працюють добре.

Недоліки:

- нелінійне навчання, може бути складним для початківців;
- багатослівність, вимагає більше коду порівняно з сучасними мовами.

Плюси і мінуси Swift. Swift, найкращий вибір для розробки програм для iOS, надає пріоритет продуктивності, чіткості коду та функціям, зручним

для розробників. Однак його зосередженість на рідних програмах iOS означає, що він не підходить для кросплатформних проектів.

Переваги:

- рідна iOS, основна мова для розробки програм для iOS;
- продуктивність, Swift відомий своєю швидкістю;
- безпека, надійна система типів зменшує кількість помилок під час виконання.

Недоліки:

- обмежено iOS, застосовується лише для розробки iOS;
- менша громада, менше ресурсів і сторонніх бібліотек.

Плюси і мінуси Python. Її універсальність і легкість читання Python у поєднанні з багатою бібліотечною екосистемою роблять його чудовим вибором для аналізу даних, наукових досліджень, веб-розробки та автоматизації.

Однак це може бути не найкращим варіантом для високопродуктивних програм.

Переваги:

- універсальність, використовується у веб-розробці, науці про дані тощо;
- читабельність, легкий для розуміння синтаксис;
- багата екосистема, велика кількість бібліотек і фреймворків.

Недоліки:

- продуктивність, повільніше, ніж такі мови, як C++;
- глобальне блокування інтерпретатора (GIL), обмежує багатопотокові можливості.

Плюси і мінуси Kotlin. Kotlin, розроблений як альтернатива Java, став найкращою мовою для розробки мобільних додатків, коли йдеться про розробку додатків для Android.

Він пропонує спрощений і простий досвід кодування. Його синтаксис точно відображає Java, що полегшує перехід для розробників Java, і він отримав підтримку від Google.

Переваги:

- зручний для android, набирає популярності як альтернатива Java;
- сумісність, повна інтеграція з Java;
- лаконічний, вимагає менше коду порівняно з Java.

Недоліки:

- обмежено Android, в основному використовується для розробки Android;
- менша громада, менша база розробників порівняно з Java.

Плюси і мінуси PHP. PHP пропонує різноманітний набір інструментів і фреймворків для оптимізації створення веб-додатків, особливо у бек-енд розробці. Хоча головна перевага PHP полягає в веб-розробці, універсальність PHP дозволяє знаходити застосування в інших сферах, включаючи фінансові технології та машинне навчання. Однак його здатність до адаптації дозволяє йому відігравати допоміжну роль у різних програмах, часто в поєднанні з іншими спеціалізованими мовами.

Переваги:

- веб-універсальність, одна з мов розробки додатків, яка підтримує широкий спектр веб-додатків, від простих веб-сайтів до складних систем керування вмістом і процвітаючих платформ електронної комерції;
- інтеграція бази даних, безпроблемна інтеграція бази даних розширює можливості PHP для обробки на стороні сервера та надійної взаємодії з базою даних;
- багата екосистема, PHP пропонує широкий спектр фреймворків і бібліотек, надаючи розробникам необхідні інструменти для ефективної веб-розробки.

Недоліки:

- обмежена сфера застосування, основний фокус РНР на веб-розробці може не підходити для не-веб-додатків або ширшої розробки додатків;
- міркування продуктивності, хоча РНР підходить для веб-розробки, РНР може не відповідати продуктивності мов, оптимізованих для конкретних випадків використання.

Плюси і мінуси Ruby. Ruby - універсальна мова програмування, відома своїм читабельним і елегантним синтаксисом. Хоча його серце лежить у веб-розробці, адаптивність Ruby поширює його вплив на різноманітні сфери, включаючи електронну комерцію, розваги та аналітику даних. Його гнучкість дозволяє ефективно співпрацювати зі спеціалізованими мовами в різних сценаріях розробки додатків.

Переваги:

- читабельний синтаксис, зручний для людини синтаксис Ruby полегшує ефективне та виразне написання коду;
- багата екосистема, безліч перлин і бібліотек спрощують розробку різноманітних веб-додатків;
- веб-розробка, Ruby on Rails, потужна структура, спрощує розробку веб-додатків.

Недоліки:

- обмежена сфера застосування, Ruby може бути не найкращим вибором для невеб-програм;
- міркування продуктивності, виразність Ruby може вплинути на його продуктивність порівняно з мовами, оптимізованими для конкретних випадків використання;
- крива навчання, менша спільнота розробників і крутіша крива навчання можуть вимагати додаткових зусиль і ресурсів для досягнення навичок роботи з Ruby.

Розглянемо застосування наведених мов програмування у реальних проектах.

Приклади проектів JavaScript:

- веб-розробка, ця мова розробки додатків працює на таких інтерактивних сайтах, як Google Maps, Facebook і Netflix. Це також ядро Node.js, яке використовується Mozilla, Netflix, PayPal, LinkedIn і Uber для різних програм;
- мобільні програми, використовується в таких програмах, як WhatsApp, Instagram і Airbnb через React Native і Flutter;
- приклади Java-проектів програми для Android, домінує в розробці додатків для Android за допомогою Google Maps, YouTube і Spotify;
- програмне забезпечення підприємства, ключ до систем CRM і ERP, таких як Salesforce і SAP.

Приклади проектів Swift:

- програми для iOS, основна мова розробки додатків TikTok, Amazon і Twitter;
- програми Mac, використовується в таких програмах, як Safari, Pages і Numbers.

Приклади проектів Python:

- машинне навчання та ШІ, використовує такі популярні фреймворки, як TensorFlow і PyTorch;
- наука про дані, - Pandas та NumPy ;
- веб-розробка, фреймворки Django та Flask живлять такі сайти, як Instagram і Pinterest.

Приклади проекту Kotlin:

- програми для Android, завойовує позиції завдяки таким програмам, як Google Pay, JetBrains і Slack;

- розробка на стороні сервера, використовується на таких платформах, як Netflix і Airbnb через Spring Boot і Ktor.

Приклади проектів Ruby:

- веб-розробка, Ruby відіграє ключову роль у створенні веб-сайтів, зокрема Airbnb і GitHub;
- електронна комерція, він підтримує такі платформи, як Etsy, відомий онлайн-ринок.

Приклади PHP проектів:

- управління контентом, PHP керує такими системами, як WordPress і Joomla;
- рішення для електронної комерції, розширює можливості WooCommerce та інших великих платформ онлайн-покупок.

Узагальнені фактори прийняття рішень. Коли завершено вибір мови програмування для мобільних додатків, важливо бути в курсі поточних галузевих тенденцій, таких як зростаюче значення штучного інтелекту (ШІ) у різних секторах. Ця інновація може забезпечити суттєву перевагу, якщо її інтегрувати у програму. Щоб прийняти рішення, враховуємо такі ключові фактори:

- вимоги до проекту, переконуємося, що вибрана мова відповідає потребам проекту та сумісності з платформою;
- експертиза розробника, оцінка наявності розробників, які володіють вибраною мовою;
- кросплатформенність проти нативної, визначаємо, чи кросплатформна чи власна розробка найкраще відповідає цілям проекту;
- сильні та слабкі сторони мови, зважити всі переваги та недоліки популярних мов, щоб знайти найкращий варіант для проекту.

- роль фреймворків, дізнатися, як фреймворки можуть прискорити та спростити процес розробки;
- реальні випадки використання та тенденції, тримати руку на пульсі реальних додатків і нових галузевих тенденцій, щоб залишатися конкурентоспроможними;
- спільнота розробників, звернути увагу на підтримку та ресурси, надані спільнотою розробників мови;
- сумісність із новими технологіями, переконатися, що мова підтримує передові технології, такі як III;
- довгострокова масштабованість, оцінити, чи підходить мова для сталого розвитку проекту.

З усього вищенаведеного робимо висновок про доцільність використання мови Python для нашої розробки.

3.2 Створення програмної реалізації

Реалізація проксі на Python [10, 11].

Мобільний проксі працює за тим же принципом, що і серверний, але змінилося підключення: тепер він маскує IP під ідентифікатор стільникового оператора, що дозволяє виконувати дії без загрози бану. Ідея полягає в тому, що мобільні з'єднання можуть надати IP для кількох пристроїв, тому кілька звернень до сервера виглядають природно.

Однак бажано вибрати мобільні проксі, які замінюють IP і маскують трафік, роблячи його більш гуманним. У цьому випадку вони будуть передавати інформацію на вихідний сервер так, ніби ви використовуєте смартфон і стільникову мережу (рис.3.5).

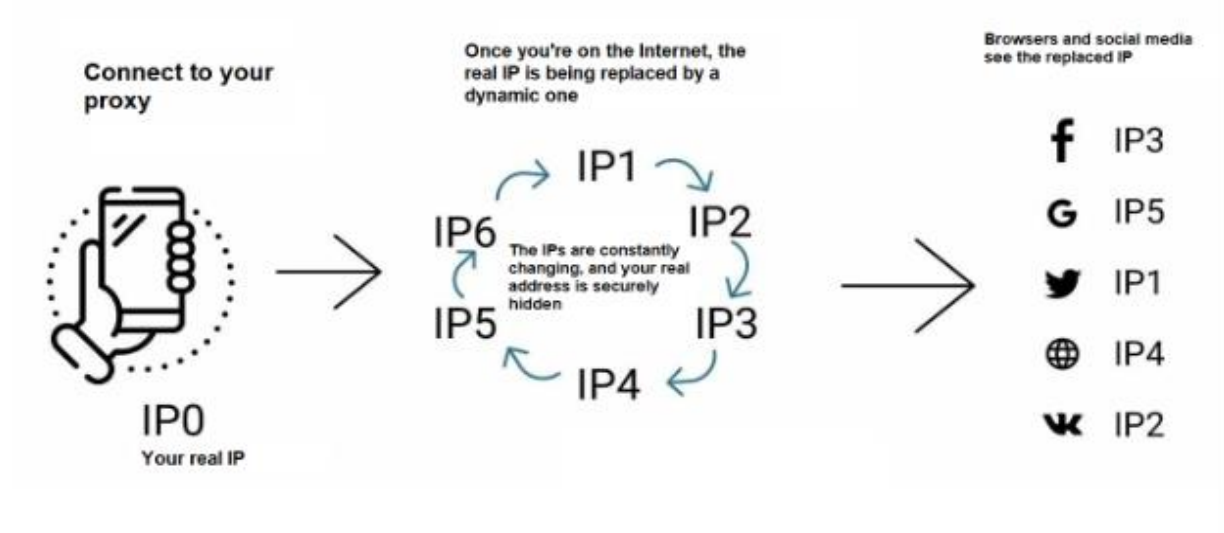


Рисунок 3.5 - Робота мобільного проксі

Відмінності мобільного проксі.

Стандартний проксі робить користувача максимально анонімним, а мобільний проксі замінює IP-адресу, гарантуючи, що системи безпеки сайту вважають користувача справжнім.

Мобільні проксі роблять це, розміщуючи динамічну IP-адресу, надану операторами стільникового зв'язку; однак вони мають обмежений діапазон. Але найголовніше, що їх неможливо заблокувати, оскільки афілійовані особи та дюжина звичайних користувачів можуть використовувати одну адресу.

Тому соціальні мережі більш поблажливо ставляться до IP-адрес, що належать стільниковим операторам. Блокують лише в крайньому випадку, не за рекламу, а за більш серйозні порушення. Таким чином, мобільні проксі зводять ризик блокування IP до мінімуму.

Приватні та загальні канали. Існує два типи каналів зв'язку:

- загальні - дешевше і видається великій кількості користувачів, кількість IP обмежена;
- private - доступний за окрему плату, але ви будете єдиним користувачем.

Переваги private варіанту:

- відсутність обмежень швидкості, за винятком тих, що вводяться оператором зв'язку;
- необмежену кількість підключень;
- замовити будь-який інтервал зміни адреси або викликати його через API.
- придбати доступ до приватного каналу без тайм-ауту роботи після зміни IP.

Досвідчені філії змушені використовувати приватні мобільні проксі-канали через часті проблеми, спричинені загальними з'єднаннями. Є «жадібні» користувачі, які перевантажують мережу, створюючи десятки з'єднань, що призводить до падіння швидкості. Працівники там зазвичай не мають великого досвіду, тому їхні дії підривають IP. У цьому випадку служби безпеки будуть ретельніше контролювати ваші облікові записи та потенційно можуть їх заблокувати. Зміна IP на таких каналах відбувається за таймером. Якщо це станеться в кількох акаунтах одночасно (наприклад, в Instagram), ситуація викличе підозри у алгоритмів безпеки. Якщо немає можливості вирішити, який канал вибрати в мобільних проксі, спробуйте обидва варіанти. Багато фахівців масового характеру віддають перевагу загальним зв'язкам приватним.

Проксі-ферма на Python дозволяє маршрутизувати запити HTTP/S через велику мережу IP-адрес за допомогою коду Python. Він підтримує такі функції, як ротація IP-адрес, збереження сеансу та геолокаційні націлення [11]. Можливості програмування на Python дозволяють легко реалізувати базову проксі-ферму, що дозволяє користувачам аналізувати, змінювати або перенаправляти мережевий трафік [10, 11].

Створення проксі-ферми містить 6 основних етапів.

1. Для початку треба ініціалізувати проекту Python. Перед початком роботи переконайтеся треба встановити на комп'ютері Python 3+. Потім

створити папки `python-http-proxy-server` та ініціалізації проекту Python з віртуальним середовищем усередині нього (рис.3.6):

```
mkdir python-http-proxy-server

cd python-http-proxy-server

python -m venv env
```

Рисунок 3.6 – Кроки ініціалізації проекту Python

Відкрититеку `python-http-proxy-server` у Python IDE та створіть порожній файл `proxy_server.py`.

2. Створити сервер веб-сокетів для прийому вхідних запитів (сокет - це низькорівнева програмна абстракція, яка забезпечує двонаправлений обмін даними між клієнтом і сервером, серверний сокет використовується для прослуховування вхідних підключень від клієнтів) (рис.3.7).

Це ініціалізує сервер вхідних сокетів та прив'язує його до локальної адреси `http://127.0.0.1:8888`. Потім це дозволяє серверу приймати з'єднання за допомогою способу прослуховування `()`.

Можливо змінити номер порту, який має прослуховувати веб-проксі. Можна також змінити сценарій для зчитування цієї інформації з командного рядка для забезпечення максимальної гнучкості.

Сокет береться із стандартної бібліотеки Python (рис.3.8).

Щоб переконатися, що проксі-сервер Python запущено належним чином, треба зареєструвати наступне повідомлення (рис.3.9).

3. Прийом запитів клієнтів. Коли клієнт підключається до проксі-сервера, йому необхідно створити новий сокет для цього конкретного клієнта (рис.3.10):


```
port = 8888
```

```
# bind the proxy server to a specific address and port
```

```
server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)  
# accept up to 10 simultaneous connections
```

```
server.bind(('127.0.0.1', port))
```

```
server.listen(10)
```

Рисунок 3.7 - Створення серверу веб-сокетів

```
import socket
```

Рисунок 3.8 - Сокет береться із стандартної бібліотеки Python

```
print(f"Proxy server listening on port {port}...")
```

Рисунок 3.9 – Перевірка правильності запуску проксі-сервера

```
# listen for incoming requests

while True:

    client_socket, addr = server.accept()

    print(f"Accepted connection from {addr[0]}:{addr[1]}")

    # create a thread to handle the client request

    client_handler =
    threading.Thread(target=handle_client_request, args=
    (client_socket,))

    client_handler.start()
```

Рисунок 3.10 – Створення сокету для нового клієнта

Для одночасної обробки кількох клієнтських запитів слід використовувати багатопоточність, треба імпортувати функцію багатопоточності зі стандартної бібліотеки Python (рис.3.11):

```
import threading
```

Рисунок 3.11 - Імпорт функції багатопоточності зі стандартної бібліотеки Python

Проксі обробляє вхідні запити за допомогою спеціальної функції `handle_client_request ()`.

4. Обробка вхідних запитів. Після створення клієнтського сокету необхідно використовувати його для:

- зчитування даних із вхідних запитів;
- вилучення хоста та порту цільового сервера з цих даних;

- використання його для надсилання клієнтського запиту на цільовий сервер;
- отримання відповіді та відправлення її вихідному клієнту.

Розглянемо перші два кроки.

Необхідно визначити функцію `handle_client_request ()`, вона та використовується для читання даних із вхідного запиту (рис.3.12).

```
def handle_client_request(client_socket):

    print("Received request:\n")

    # read the data sent by the client in the request

    request = b''

    client_socket.setblocking(False)

    while True:

        try:

            # receive data from web server

            data = client_socket.recv(1024)

            request = request + data

            # Receive data from the original destination server

            print(f"{data.decode('utf-8')}")

        except:

            break
```

Рисунок 3.12 – Функція `handle_client_request ()`,

`setblocking` (брехня) переводить клієнтський сокет у неблокуючий режим.

`recv()` використовується для читання вхідних даних та додавання їх у запит у байтовому форматі.

Оскільки невідомо розмір даних вхідного запиту, тоді потрібно буде зчитувати їх частинами за раз. В даному випадку вказано фрагмент розміром 1024 байти. У неблокувальному режимі, якщо `recv()` не знайде жодних даних, виникне виняток через помилку. Таким чином, інструкція `except` (виключення) позначає завершення операції.

Також треба звернути увагу на зареєстровані повідомлення, щоб відстежувати, що робить проксі-сервер Python.

Після отримання запиту необхідно визначити з нього хост (рис.3.13). і порт цільового сервера (рис.3.14).

```
host, port = extract_host_port_from_request(request)

In particular, this is what the extract_host_port_from_request()
function looks like:

def extract_host_port_from_request(request):

    # get the value after the "Host:" string

    host_string_start = request.find(b'Host: ') + len(b'Host: ')

    host_string_end = request.find(b'\r\n', host_string_start)

    host_string =
request[host_string_start:host_string_end].decode('utf-8')

    webserver_pos = host_string.find("/")

    if webserver_pos == -1:

        webserver_pos = len(host_string)

    # if there is a specific port

    port_pos = host_string.find(":")
```

Рисунок 3.13 – Визначення хосту цільового сервера

```

# no port specified

if port_pos == -1 or webserver_pos < port_pos:

    # default port

    port = 80

    host = host_string[:webserver_pos]

else:

    # extract the specific port from the host string

    port = int((host_string[(port_pos + 1):])[webserver_pos
- port_pos - 1])

    host = host_string[:port_pos]

return host, port

To better understand what it does, consider the example below.
This is what the encoded string of an incoming request usually
contains:

GET http://example.com/your-page HTTP/1.1

Host: example.com

User-Agent: curl/8.4.0

Accept: */*

Proxy-Connection: Keep-Alive

```

Рисунок 3.14– Визначення порту цільового сервера

`extract_host_port_from_request ()` повертає хост і порт веб-сервера з поля "Host:" ("Хост:"). Хостом є `example.com`, а порт — 80 (оскільки конкретний порт не вказаний).

5. Пересилання запиту клієнта та обробка відповіді. Знаючи цільовий хост та порт, тепер потрібно переслати запит клієнта на цільовий сервер (рис.3.15). У `handle_client_request ()` необхідно створити новий веб-сокет і використовувати його для надсилання вихідного запиту до потрібного місця призначення (рис.3.16).

```

# create a socket to connect to the original destination server

destination_socket = socket.socket(socket.AF_INET,
socket.SOCK_STREAM)

# connect to the destination server

destination_socket.connect((host, port))

# send the original request

destination_socket.sendall(request)

Then, get ready to receive the server response and propagate it
to the original client:

# read the data received from the server

# once chunk at a time and send it to the client

print("Received response:\n")

while True:

    # receive data from web server

```

Рисунок 3.15 – Запит клієнта на цільовий сервер

Потрібно працювати по одному фрагменту за раз, тому що не відомо розмір відповіді. Якщо дані порожні, їх не потрібно приймати і операцію можна перервати.

Після цього треба закрити два сокети, які визначили у функції (рис.3.17).

6. Заключний етап. Збирається все в один код. Повний код на Python надано у додатку В.

```

while True:

    # receive data from web server

    data = destination_socket.recv(1024)

    # Receive data from the original destination server

    print(f"{data.decode('utf-8')}")

    # no more data to send

    if len(data) > 0:

        # send back to the client

        client_socket.sendall(data)

    else:

        break

```

Рисунок 3.16 – Створення нового веб-сокету

```

# close the sockets

destination_socket.close()

client_socket.close()

```

Рисунок 3.17 – Закриття сокетів

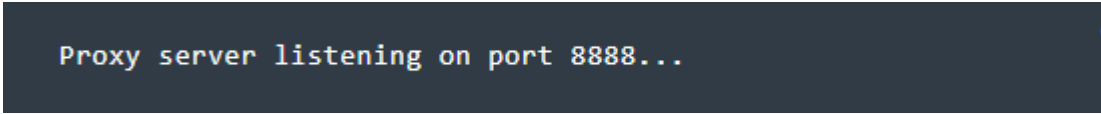
3.3 Тестування програмної реалізації

Для тестування розробленого програмного модуля було створено проксі-ферму з динамічною зміною IP-адреси у складі:

- домашній персональний комп'ютер (у якості сервера);

- роутер для підключення комп'ютера до мережі Інтернет;
- USB-концентратор на 10 портів для підключення смартфонів;
- смартфони Huawei з ANDROID (5 шт.) та дроти підключення до концентратора.

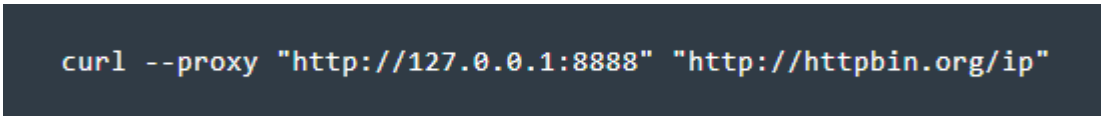
Завантажуємо додаток. У терміналі має з'явитися таке повідомлення (рис.3.18):



```
Proxy server listening on port 8888...
```

Рисунок 3.18 – Повідомлення термінала

Щоб переконатися, що сервер працює, виконали проксі-запит за допомогою cURL. Відкриваємо новий термінал та запускаємо (рис.3.19):

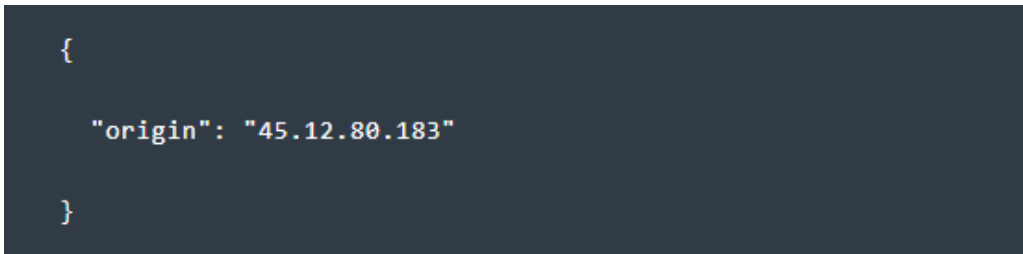


```
curl --proxy "http://127.0.0.1:8888" "http://httpbin.org/ip"
```

Рисунок 3.19 – Запуск у новому терміналі

Це дозволить надіслати запит GET до адресата <http://httpbin.org/ip> через проксі-сервер <http://127.0.0.1:8888>.

Отримаємо (рис.3.20):



```
{  
  
  "origin": "45.12.80.183"  
  
}
```

Рисунок 3.20 – Відповідь на запит

Це IP-адреса проксі-сервера. Оскільки кінцева точка /ip проекту HttpBin повертає IP-адресу, з якої надійшов запит. Якщо запускати сервер локально, origin буде відповідати нашій IP-адресі.

Тепер у журналі міститься запис програмою проксі на Python (рис.3.21)

```
Received request:

GET http://httpbin.org/ip HTTP/1.1

Host: httpbin.org

User-Agent: curl/8.4.0

Accept: */*

Proxy-Connection: Keep-Alive

Received response:

HTTP/1.1 200 OK

Date: Thu, 14 Dec 2023 14:02:08 GMT

Content-Type: application/json

Content-Length: 31

Connection: keep-alive

Server: gunicorn/19.9.0

Access-Control-Allow-Origin: *

Access-Control-Allow-Credentials: true

{
  "origin": "45.12.80.183"
}
```

Рисунок 3.21 - Запис програмою проксі на Python

Це свідчить про те, що проксі-сервер отримав запит у форматі, вказаному протоколом HTTP. Потім він переслав його на цільовий сервер, зареєстрував дані відповіді та відправив відповідь назад клієнту. Отже, проксі працює.

3.4 Висновок до розділу 3

Обрано об'єктно-орієнтовану мову програмування Python, як програмного засобу реалізації програмного модуля проксі ферми з динамічною зміною IP-адреси у середовищі розробки - Visual Studio. Створено програмний модуль реалізації проксі ферми. Проведено тестування програмної реалізації, яке підтвердило її роботоспроможність [12].

ВИСНОВКИ

У ході виконання бакалаврської дипломної роботи було досягнуто наступних результатів.

Проведено аналіз галузі застосування проксі ферм підтвердив переваги власних мобільних проксі ферм, які дозволяють використовувати їх у розв'язанні багатьох задачах сучасних інформаційних технологій. Аналіз відомих методів розв'язання задач створення та використання проксі ферми робе актуальним питання розробки найпростішої власної мобільних проксі ферми на смартфонах, що дозволяє формування мережі з динамічною зміною IP-адреси.

Виконана розробка програмного модуля проксі ферми з динамічною зміною IP-адреси, що пов'язано з використанням проксі-сервера, який використовує мобільні мережі та пристрої для маршрутизації інтернет-трафіку. Мобільний проксі-сервер використовує мобільні мережі та має мобільні IP-адреси. Використання мобільного проксі-сервера дозволяє перенаправляти інтернет-запити через мобільні пристрої та їхні мережі, даючи змогу користувачам змінювати свою IP-адресу та симулювати підключення через мобільні мережі. Швидкість обміну може варіюватися залежно від мережевих умов і якості з'єднання, але вони можуть бути швидкими для багатьох завдань.

Обґрунтовано засоби розробки програмного модуля проксі ферми. Деталізовані класи програмного забезпечення програмного модуля проксі ферми з динамічною зміною IP-адреси. Розроблені UML діаграми послідовності роботи проксі ферми.

Розроблено програмний модуль проксі-ферми з динамічною зміною IP-адреси пов'язана з використанням проксі-сервера, який використовує мобільні мережі та пристрої для маршрутизації інтернет-трафіку. Мобільний проксі-сервер використовує мобільні мережі та має мобільні IP-адреси.

Використання мобільного проксі-сервера дозволяє перенаправляти інтернет-запити через мобільні пристрої та їхні мережі, даючи змогу користувачам змінювати свою IP-адресу та симулювати підключення через мобільні мережі. Швидкість обміну може варіюватися залежно від мережевих умов і якості з'єднання, але вони можуть бути швидкими для багатьох завдань

Обґрунтовано вибір об'єктно-орієнтованої мови програмування Python як програмного засобу реалізації програмного модуля проксі ферми з динамічною зміною IP-адреси у середовищі розробки - Visual Studio. Створено програмний модуль реалізації проксі ферми. Проведено тестування програмної реалізації, яке підтвердило її роботоспроможність.

Результати власної розробки дозволяють збільшити швидкодію, продуктивність, надійність, зменшити витрат на експлуатацію та спростити керування проксі фермою з динамічною зміною IP-адреси за рахунок уникнення використання зовнішніх модулів та простої організації модуля. Одержані результати можливо використовувати сторонніми користувачами проксі ферми з динамічною зміною IP-адреси.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Арбітраж трафіку - що це таке. Комплексний огляд. URL: <https://leadpanda.media/blog/scho-tse-take-arbitrazh-trafik>
2. Virtual private server. URL: https://en.wikipedia.org/wiki/Virtual_private_server
3. 7 Найкращих постачальників мобільних проксі (2024). URL: <https://www.guru99.com/uk/mobile-proxies.html>
4. Топ проксі-сервісів для арбітражу. URL: <https://leadpanda.media/blog/top-proxi-servisiv-dlya-arbitrazu>
5. Root Android: що це таке і для чого він потрібен. URL: <https://technari.com.ua/ua/services/about-company/articles/android-root>
6. Стандарти мобільного зв'язку 4G-5G-6G – подібність, відмінності, перспективи. URL: <https://www.itbox.ua.ua/blog/Standarti-mobilnogo-zvyazku-4G5G6G--podibnist-vidminnosti-perspektivi>
7. PostMan. URL: <https://www.getpostman.com>
8. 4G LTE Advanced. URL: <https://www.electronics-notes.com/articles/connectivity/4g-lte-long-term-evolution/what-is-lte-advanced.php>
9. Choosing the Best Language for App Development: 7 Options to Consider. URL: <https://forbytes.com/blog/best-language-for-app-development>
10. 20 найкращих сайтів для вивчення Python. URL: https://uaspectr.com/2020/07/06/najkrashhi-sajty-dlya-vyvchennya-python/#google_vignette
11. Python Підручник. URL: https://w3schoolsua.github.io/python/index.html#google_vignette
12. Методичні вказівки до виконання бакалаврських кваліфікаційних робіт для студентів спеціальності 122 «Комп'ютерні науки» [Електронний ресурс] / уклад.: А. А. Яровий, О. В. Сілагін, І. В. Богач. Вінниця : ВНТУ, 2023. 54 с.

ДОДАТКИ

ДОДАТОК А

ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи: Розробка програмного модуля проксі-ферми з динамічною зміною IP-адреси

Тип роботи: бакалаврська дипломна робота
(БДР, МКР)

Підрозділ кафедра комп'ютерних наук, ФІІТА
(кафедра, факультет)

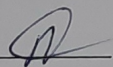
Показники звіту подібності Unicheck

Оригінальність 96,35% Схожість 3,65%

Аналіз звіту подібності (відмітити потрібне):

- ☒ Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
- ☐ Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її виконання автором. Роботу направити на розгляд експертної комісії кафедри.
- ☐ Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

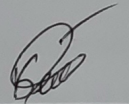
Особа, відповідальна за перевірку



Озеранський В.С.

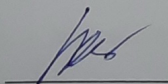
Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи.

Автор роботи



Богомазов А.А.

Керівник роботи



Денисюк В.О.

ДОДАТОК Б

ІНСТРУКЦІЯ КОРИСТУВАЧА

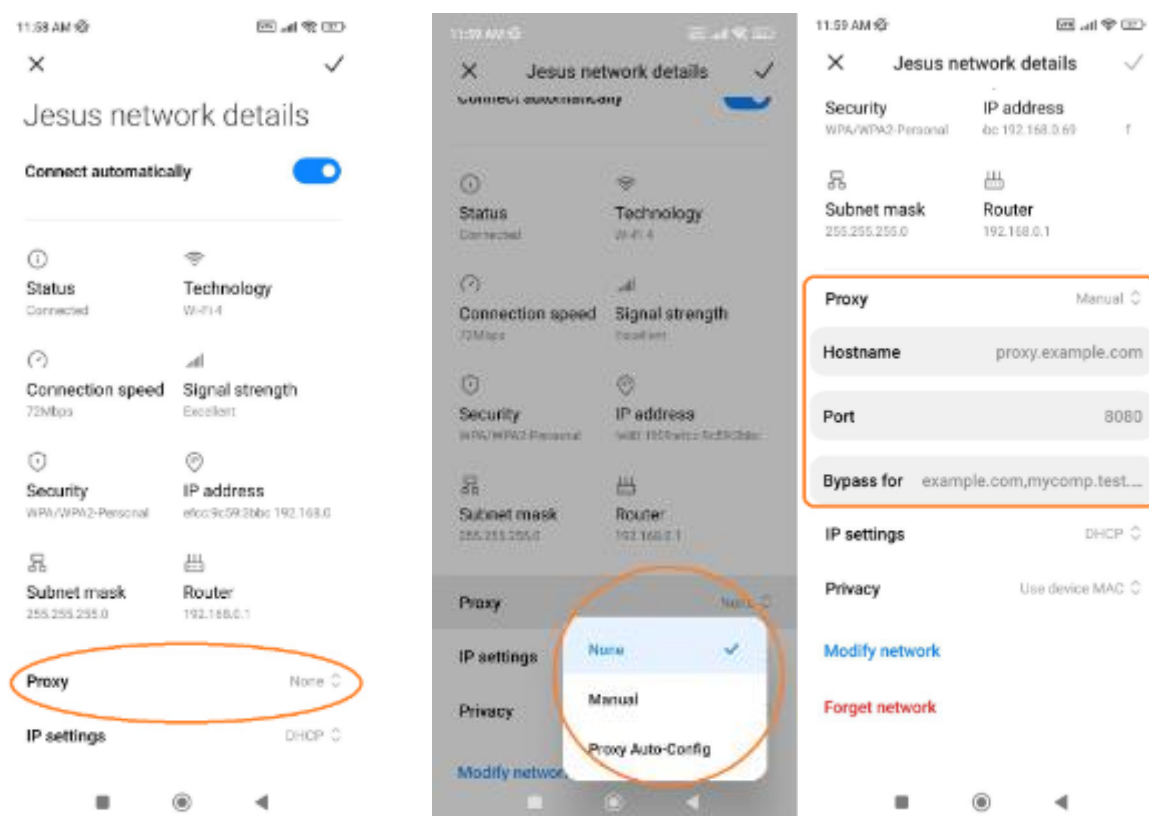
Підключити обладнання.

Завантажити Проксі .

Відкрийте список опцій, натиснувши опцію «Немає» , і виберіть «Вручну» .

Заповніть пропуски , використовуючи дані, надані вибраним сервером.

Тепер збережіть зміни та увійдіть під час першого відкриття браузера.



Дотримуйтеся того ж шаблону, щоб налаштувати мобільний проксі для iOS. Введіть координати підключення через налаштування.

Деякі параметри можуть допомогти автоматизувати процес налаштування.

Налаштування на прикладі сервісу ProxyDroid, який рекомендують.

Надати програмі права суперкористувача після першого запуску; без цього він не працюватиме належним чином. Немає ризику рутувати ваш корпоративний телефон, але постарайтеся не робити цього на особистому.

Після цього налаштуйте підключення. Почніть із введення своєї IP-адреси.

Вкажіть порт.

Вкажіть тип протоколу. Використання служб дозволить налаштувати не лише з'єднання HTTPS, а й використовувати Socks 4 або 5.

Наприклад, SOCKS5.

Введіть ім'я користувача та пароль.

Запустити програму.

Ви можете вказати, коли використовувати створене з'єднання, або вибрати глобальний режим.

Після налаштування мобільного проксі підключіть телефон до комп'ютера або перевірте працездатність програми через браузер смартфона.

Для деяких служб потрібні права суперкористувача. Намагайтеся уникати цього з міркувань безпеки. Кореневі права дають додаткам необмежений доступ до системних файлів, з можливістю їх зміни та заміни на інші.

Однак смартфони з необмеженим доступом небезпечні, тому фірмові банківські додатки та ліцензійні програми розважального контенту відмовляються з ними працювати.

Сам телефон може перегріватися під час роботи з root-доступом, що може призвести до збою акумулятора та інших систем. У цьому випадку на корпус може не поширюватися гарантія.

При цьому рутовані додатки працюють більш стабільно, але потрібно вибирати ті опції, які вже активно використовуються іншими партнерами. Вони мають меншу ймовірність збою та часто мають вбудовані функції, які

дозволяють блокувати стеження за додатками. Вони також імітують конфігурації звичайного користувача.

Якщо ви використовуєте службу для налаштування проксі, перевірте її на безпеку. Переглядайте інформацію та коментарі, щоб запобігти витоку ваших особистих даних. Такі витоки фіксуються спеціалізованими веб-сайтами та повідомляються користувачами в оглядах, у тому числі на форумах афілійованого маркетингу.

ДОДАТОК В ЛІСТИНГ ПРОГРАМИ

```
import socket

import threading

def handle_client_request(client_socket):

    print("Received request:\n")

    # read the data sent by the client in the request

    request = b''

    client_socket.setblocking(False)

    while True:

        try:

            # receive data from web server

            data = client_socket.recv(1024)

            request = request + data

            # Receive data from the original
destination server

            print(f"{data.decode('utf-8')}")

        except:

            break

    # extract the webserver's host and port from the
request

    host, port =
extract_host_port_from_request(request)

    # create a socket to connect to the original
destination server
```

```

    destination_socket = socket.socket(socket.AF_INET,
socket.SOCK_STREAM)

    # connect to the destination server

    destination_socket.connect((host, port))

    # send the original request

    destination_socket.sendall(request)

    # read the data received from the server

    # once chunk at a time and send it to the client

    print("Received response:\n")

    while True:

        # receive data from web server

        data = destination_socket.recv(1024)

        # Receive data from the original destination
server

        print(f"{data.decode('utf-8')}")

        # no more data to send

        if len(data) > 0:

            # send back to the client

            client_socket.sendall(data)

        else:

            break

    # close the sockets

    destination_socket.close()

    client_socket.close()

```

```

def extract_host_port_from_request(request):

    # get the value after the "Host:" string

    host_string_start = request.find(b'Host: ') +
len(b'Host: ')

    host_string_end = request.find(b'\r\n',
host_string_start)

    host_string =
request[host_string_start:host_string_end].decode('utf-
8')

    webserver_pos = host_string.find("/")

    if webserver_pos == -1:

        webserver_pos = len(host_string)

    # if there is a specific port

    port_pos = host_string.find(":")

    # no port specified

    if port_pos == -1 or webserver_pos < port_pos:

        # default port

        port = 80

        host = host_string[:webserver_pos]

    else:

        # extract the specific port from the host
string

        port = int((host_string[(port_pos +
1):][:webserver_pos - port_pos - 1])

        host = host_string[:port_pos]

```

```

        return host, port

def start_proxy_server():

    port = 8888

    # bind the proxy server to a specific address and
    port

    server = socket.socket(socket.AF_INET,
socket.SOCK_STREAM)

    server.bind(('127.0.0.1', port))

    # accept up to 10 simultaneous connections

    server.listen(10)

    print(f"Proxy server listening on port {port}...")

    # listen for incoming requests

    while True:

        client_socket, addr = server.accept()

        print(f"Accepted connection from
{addr[0]}:{addr[1]}")

        # create a thread to handle the client request

        client_handler =
threading.Thread(target=handle_client_request,
args=(client_socket,))

        client_handler.start()

if __name__ == "__main__":

    start_proxy_server()

Launch it with this command:

python proxy_server.py

```

ДОДАТОК Г

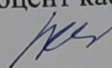
ГРАФІЧНА ЧАСТИНА

«РОЗРОБКА ПРОГРАМНОГО МОДУЛЯ ПРОКСІ ФЕРМИ З
ДИНАМІЧНОЮ ЗМІНОЮ ІР-АДРЕСИ»

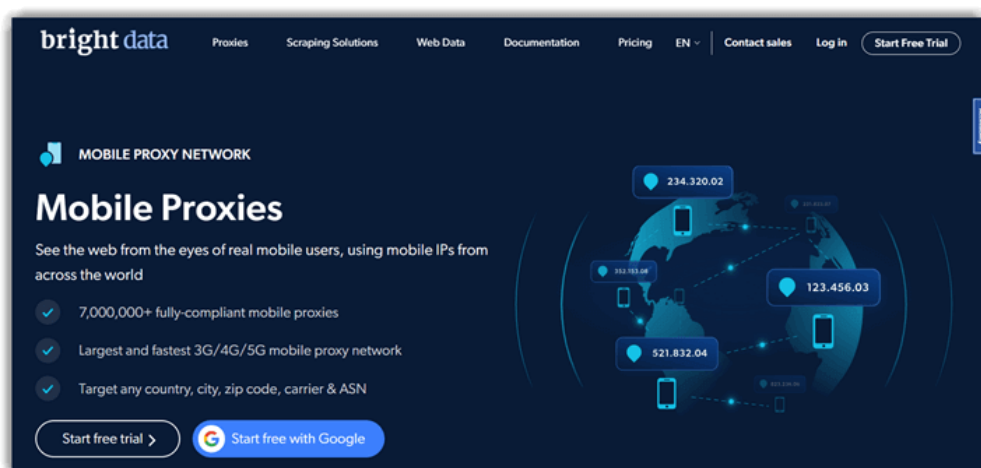
Виконав: студент 4 курсу, групи 22КН-мс
спеціальності 122 – Комп'ютерні науки
(шифр і назва напрямку підготовки, спеціальності)

 Богомазов А.А.
(прізвище та ініціали)

Керівник: к.т.н., доцент каф.КН

 Денисюк В.О.
(прізвище та ініціали)

« 07 » 06 2024 р.



bright data Proxies Scraping Solutions Web Data Documentation Pricing EN Contact sales Log in Start Free Trial

MOBILE PROXY NETWORK

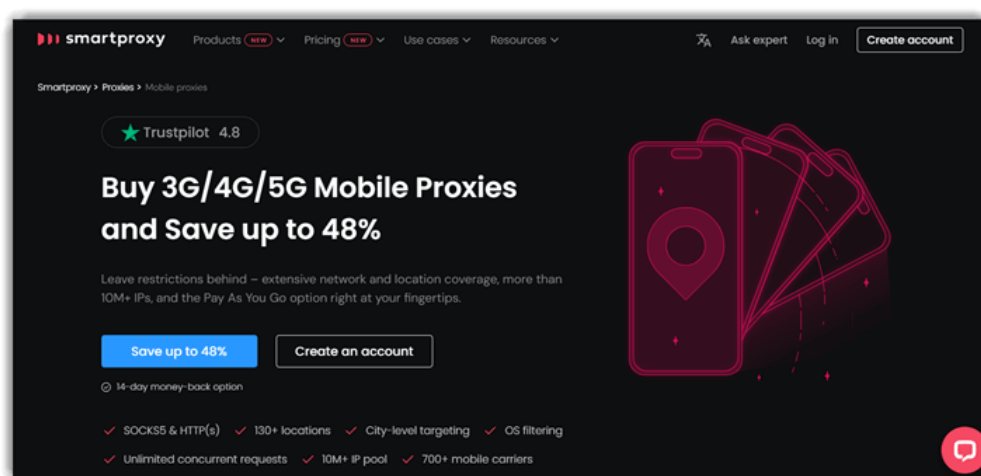
Mobile Proxies

See the web from the eyes of real mobile users, using mobile IPs from across the world

- ✓ 7,000,000+ fully-compliant mobile proxies
- ✓ Largest and fastest 3G/4G/5G mobile proxy network
- ✓ Target any country, city, zip code, carrier & ASN

Start free trial > Start free with Google

Diagram showing a globe with mobile proxy locations and IP addresses: 234.320.02, 123.456.03, 521.832.04, 155.153.08.



smartproxy Products Pricing Use cases Resources Ask expert Log in Create account

Smartproxy > Proxies > Mobile proxies

★ Trustpilot 4.8

Buy 3G/4G/5G Mobile Proxies and Save up to 48%

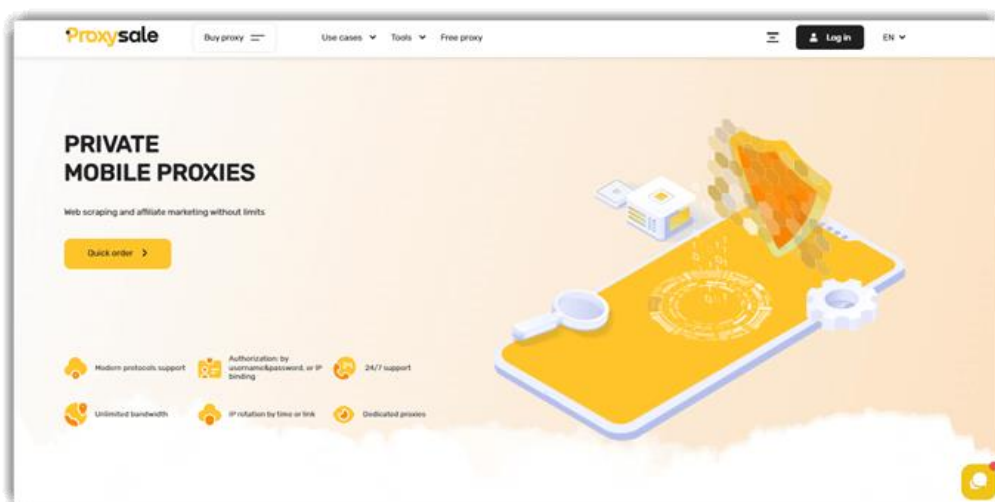
Leave restrictions behind – extensive network and location coverage, more than 10M+ IPs, and the Pay As You Go option right at your fingertips.

Save up to 48% Create an account

14-day money-back option

- ✓ SOCKS5 & HTTP(s)
- ✓ 130+ locations
- ✓ City-level targeting
- ✓ OS filtering
- ✓ Unlimited concurrent requests
- ✓ 10M+ IP pool
- ✓ 700+ mobile carriers

Diagram showing a smartphone with a location pin and a fan of cards.



proxysale Buy proxy Use cases Tools Free proxy Log in EN

PRIVATE MOBILE PROXIES

Web scraping and affiliate marketing without limits

Quick order >

- Modern protocols support
- Authorization by username/password, or IP binding
- 24/7 support
- Unlimited bandwidth
- IP rotation by time or link
- Dedicated proxies

Diagram showing a smartphone with a magnifying glass, a gear, and a shield.

Рисунок Г.1 – Приклади постачальників мобільних проксі (BrightData, SmartProxy, ProxySale)

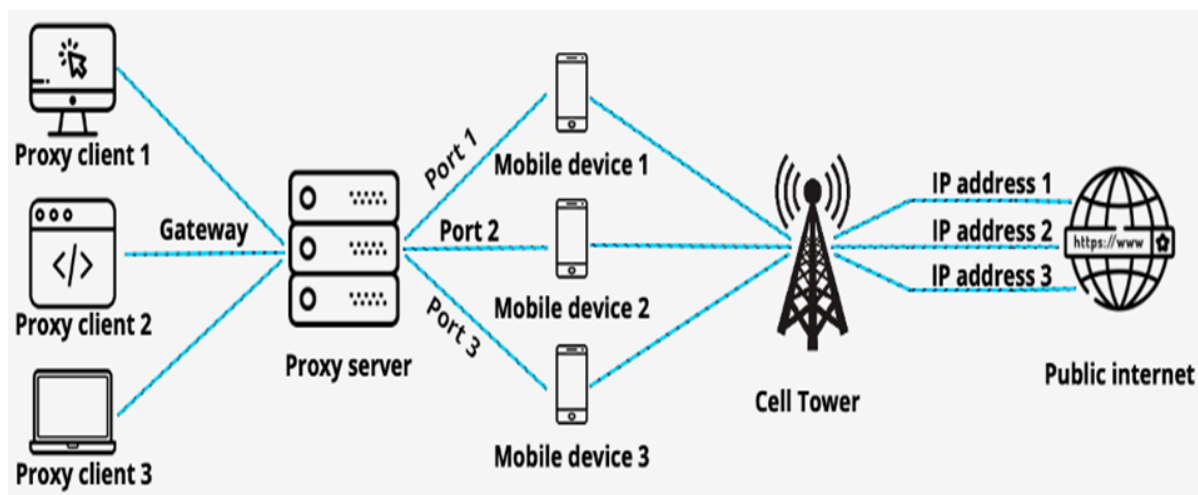


Рисунок Г.2 – Структура проксі ферми з динамічною зміною IP-адреси

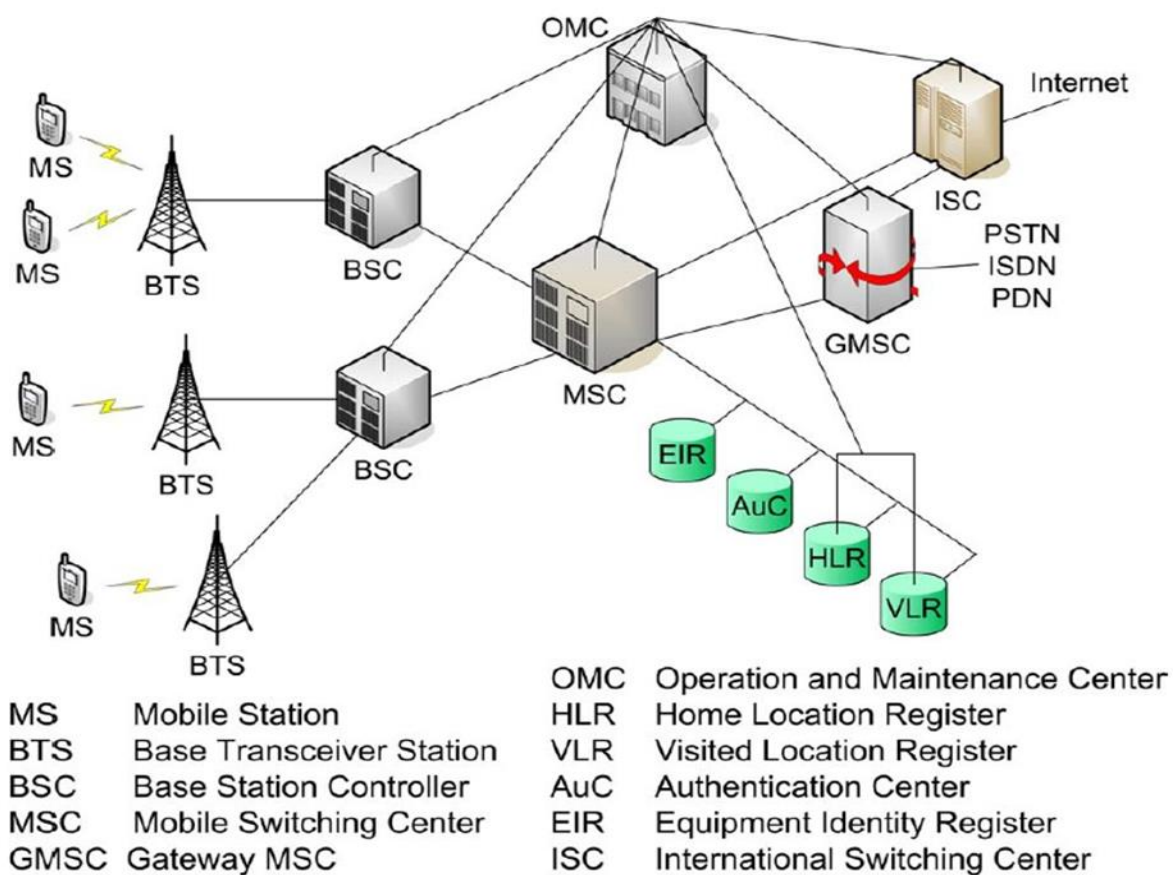


Рисунок Г.3 - Архітектура мережі GSM 4G

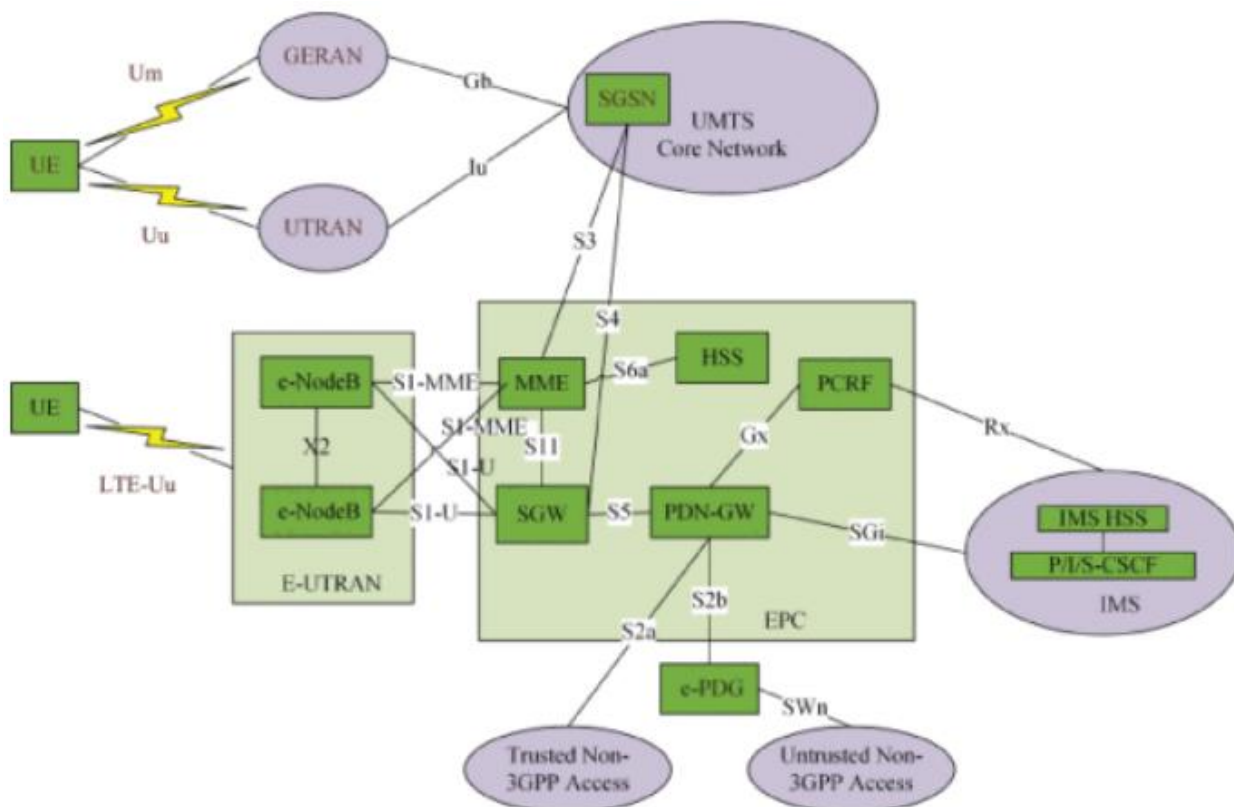


Рисунок Г.4 - Мережева архітектура LTE-Advanced

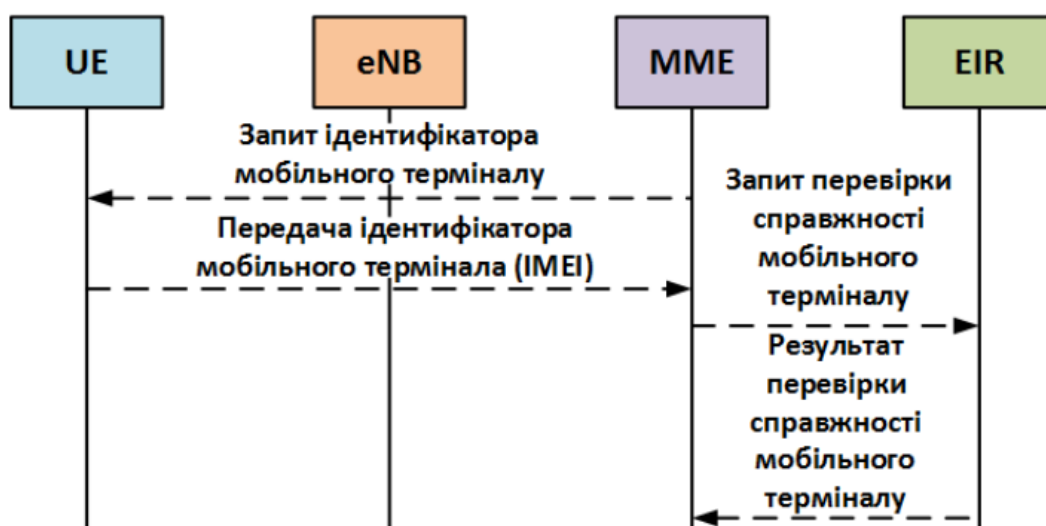


Рисунок Г.5 – Процедура автентифікації мобільного терміналу



Рисунок Г.6 – Структура мобільного-проксі

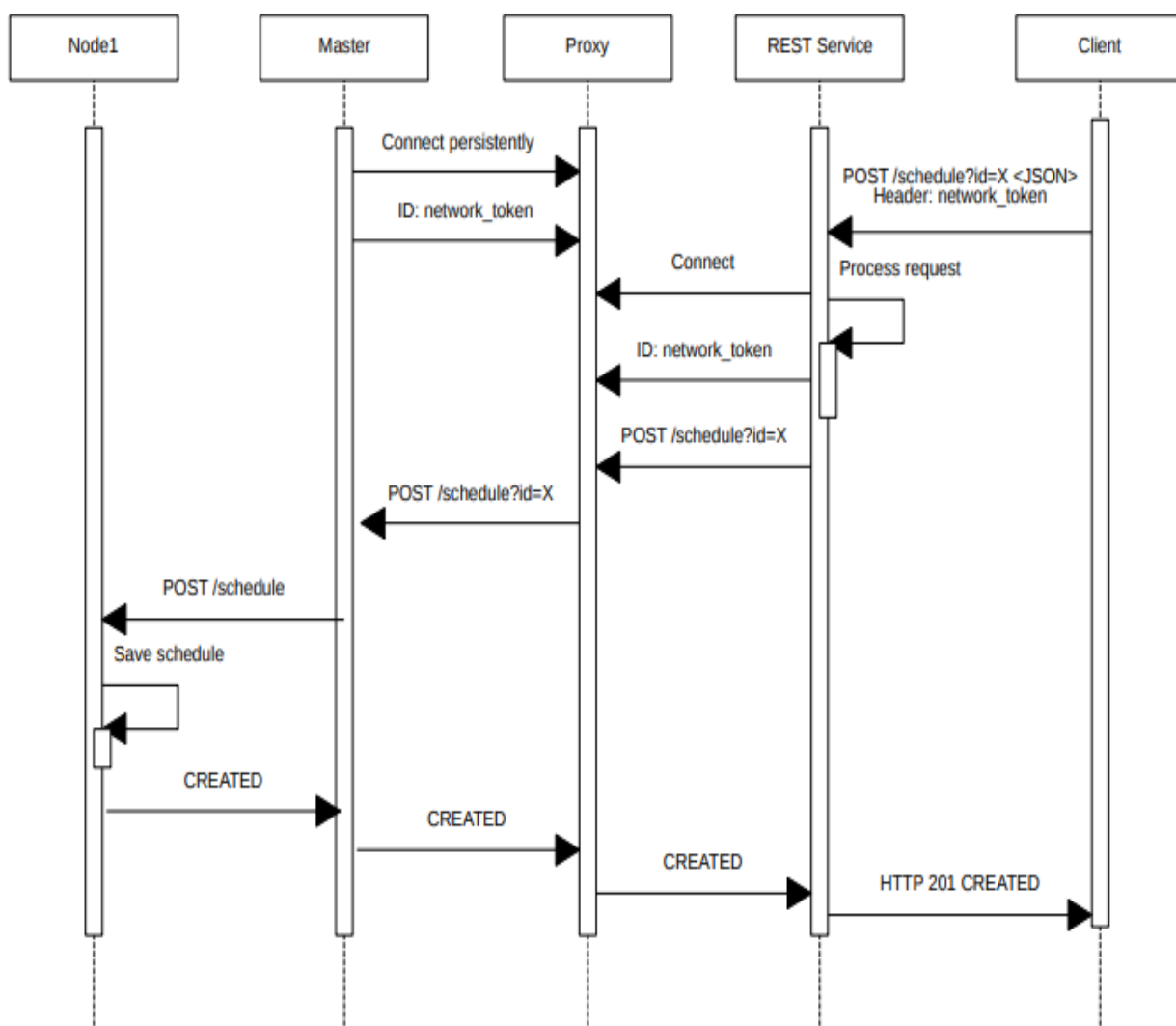


Рисунок Г.7 - UML діаграма послідовності додавання інтервалу операції проксі-сервером

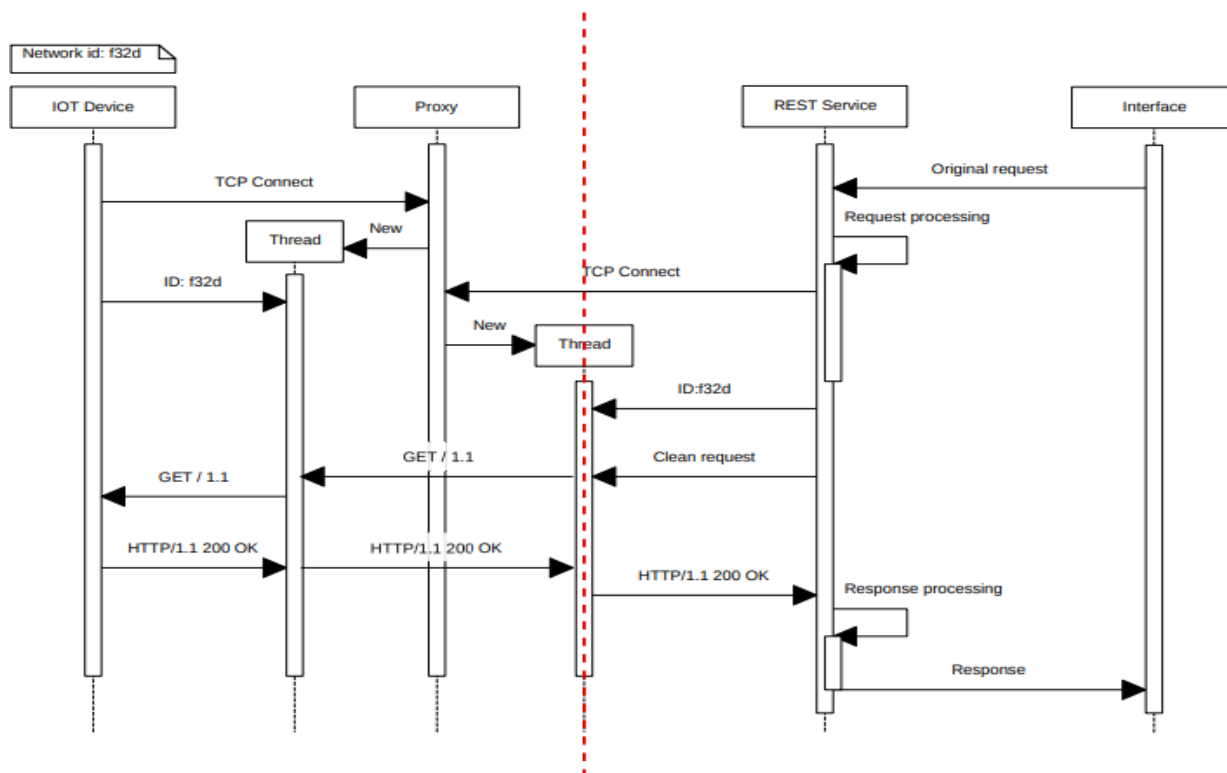


Рисунок Г.8 - UML діаграма послідовності прикладу роботи проксі-сервера

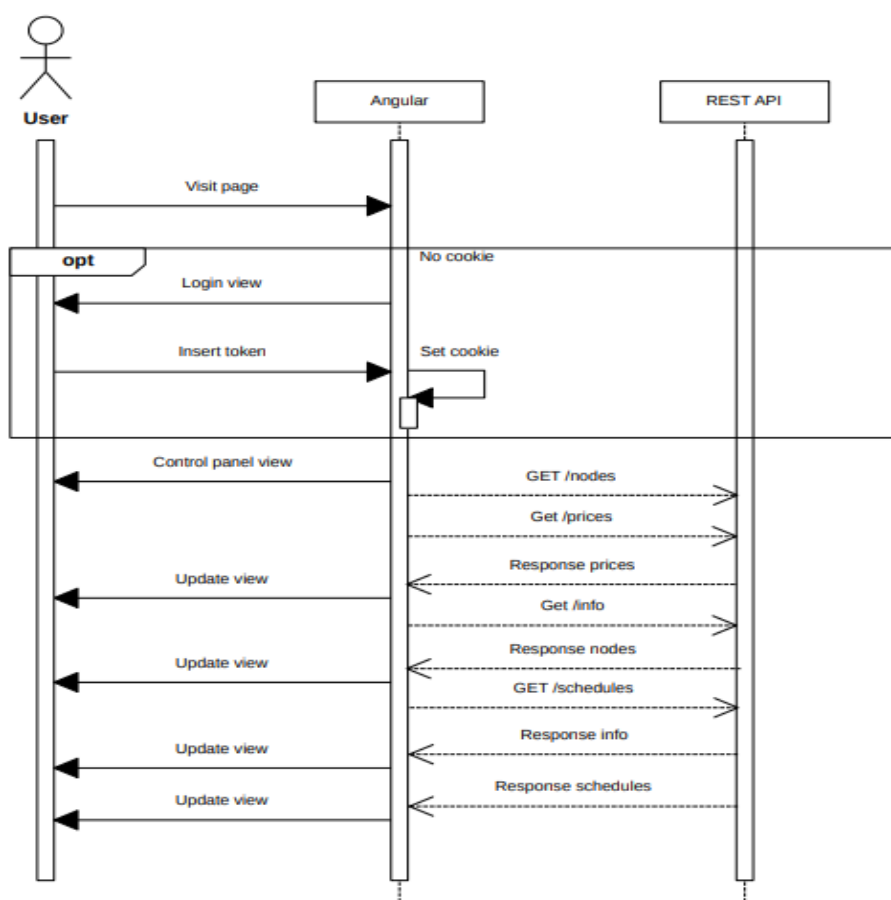


Рисунок Г.9 – UML діаграма послідовності обробки веб-додатків