

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

Бакалаврська дипломна робота на тему:

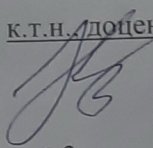
РОЗРОБКА ПРОГРАМНОГО МОДУЛЯ УПРАВЛІННЯ ПРОКСІ-ФЕРМОЮ

Виконав: студент (4) курсу, групи 1КН-206
спеціальності 122 – Комп'ютерні науки
(шифр і назва напрямку підготовки, спеціальності)



Мацько О.С.
(прізвище та ініціали)

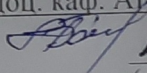
Керівник: к.т.н., доцент каф.КН



Колодний В.В.
(прізвище та ініціали)

«07» 06 2024 р.

Рецензент: к.т.н., доц. каф. АІТ



Барабаш М.В.
(прізвище та ініціали)

«07» 06 2024 р.

Допущено до захисту

Зав. кафедри Ірв'єв А.А.

«07» 06 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

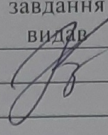
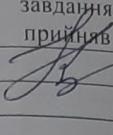
Завідувач кафедри КН
проф., д.т.н. Яровий А.А.
« 07 » 03 2024 р.

**ЗАВДАННЯ
НА БАКАЛАВРСЬКУ ДИПЛОМНУ РОБОТУ СТУДЕНТУ**

Мацько Олександр Сергійович

1. Тема роботи: Розробка програмного модуля управління проксі-фермою.
Керівник роботи: Колодний Володимир Володимирович, к.т.н., доцент
затверджені наказом вищого навчального закладу “_11_”_03_2024 року №80
2. Строк подання студентом роботи 27.05.2024
3. Вихідні дані до роботи :
мова програмування – об'єктно-орієнтована;
Кількість модемів – не менше 5шт
Наявність адміністративної панелі.
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):
 - обґрунтування доцільності розробки модуля проксі ферми;
 - аналіз переваг та недоліків існуючих проксі;
 - створення структури програмного модуля проксі ферми;
 - 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)

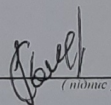
Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1-3	Колодний В.В., к.т.н., доцент		

7. Дата видачі завдання 07.03.2024 р.

КАЛЕНДАРНИЙ ПЛАН

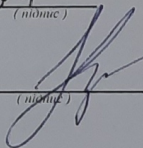
№з/п	Назва та зміст етапу	Термін виконання		Примітка
		початок	закінчення	
1	Аналіз предметної області задачі розробки програмного модуля проксі-ферми	06.05.24	10.05.24	
2	Розробка програмного модуля проксі-ферми	12.05.24	14.05.24	
3	Проектування програмного модуля проксі-ферми	15.05.24	18.05.24	
4	Реалізація програмного модуля проксі-ферми	19.05.24	23.05.24	
5	Розробка інструкції користувача	24.05.24	27.05.24	
6	Оформлення матеріалів до захисту БКР	28.05.24	06.06.24	

Студент


(підпис)

Мацько О.С.
(прізвище та ініціали)

Керівник роботи


(підпис)

Колодний В.В.
(прізвище та ініціали)

АНОТАЦІЯ

Бакалаврська дипломна робота складається з 80 сторінок формату А4, на яких є 24 рисунки, список використаних джерел містить 15 найменувань.

У роботі розглянуто достатньо актуальну задачу розробки програмного модуля проксі-ферми, а саме створення власної мобільної проксі-ферми, яка надає ряд переваг: контроль трафіку; заощадження коштів; адаптування для конкретного GEO; можливість заробітку за її оренду. Розробка пов'язана з використанням проксі-сервера, який використовує мобільні мережі та пристрої для маршрутизації інтернет-трафіку, використовує мобільні IP-адреси. Це дозволяє перенаправляти інтернет-запити через мобільні пристрої та їхні мережі, даючи змогу користувачам змінювати свою IP-адресу та симулювати підключення через мобільні мережі. Обгрунтовано засоби розробки програмного модуля проксі-ферми. Розроблені UML діаграми послідовності роботи проксі-ферми. Деталізовані класи програмного забезпечення програмного модуля проксі-ферми. У якості програмного засобу реалізації проксі-ферми обрано об'єктно-орієнтовану мову програмування Python. У якості середовища розробки обрано Visual Studio. Створено програмний модуль управління проксі-фермою, проведено тестування обраної апаратної реалізації, яке підтвердило працездатність та оптимальність візначених рішень. Результати розробки програмного модуля управління проксі-фермою дозволяють підвищити надійності, зменшити витрати та спростити керування проксі-фермою за рахунок уникнення використання зовнішнього програмного забезпечення та простої організації особистого програмного модуля. Результати розробки можливо порадити для використання усім користувачам-початківцям проксі-ферм.

Ключові слова: проксі-ферма, модем, 4G LTE, трафік

ABSTRACT

The bachelor thesis consists of 80 pages of A4 format, on which there are 24 figures, the list of used sources contains 15 names.

The work deals with the fairly relevant task of developing a proxy farm software module, namely creating your own mobile proxy farm, which provides a number of advantages: traffic control; saving money; adaptation for a specific GEO; the possibility of earning money for its rental. The development involves the use of a proxy server that uses mobile networks and devices to route Internet traffic, uses mobile IP addresses. It allows the redirection of Internet requests through mobile devices and their networks, allowing users to change their IP address and simulate connections over mobile networks. The means of developing the software module of the proxy farm are substantiated. Developed UML diagrams of the proxy farm work sequence. Detailed software classes of the proxy farm software module. The object-oriented programming language Python was chosen as the software tool for implementing the proxy farm. Visual Studio was chosen as the development environment. The proxy farm management software module was created, the selected hardware implementation was tested, which confirmed the functionality and optimality of the identified solutions. The results of the development of the proxy farm management software module make it possible to increase reliability, reduce costs and simplify proxy farm management by avoiding the use of external software and simply organizing a personal software module. The development results can be recommended for use by all novice users of proxy farms.

Keywords: proxy farm, modem, 4G LTE, traffic

ЗМІСТ

ВСТУП.....	3
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ВИКОРИСТАННЯ ПРОКСІ-ФЕРМИ.....	5
1.1 Аналіз галузі застосування проксі-ферм.....	5
1.2 Аналіз відомих методів розв’язання задач використання проксі-ферми	20
1.3 Постановка задачі.....	23
1.4 Висновок до розділу 1	26
2 РОЗРОБКА ПРОГРАМНОГО МОДУЛЯ ПРОКСІ-ФЕРМИ	27
2.1 Обґрунтування засобів розробки програмного модуля.....	27
2.2 Розробка діаграми класів.....	38
2.3 Розробка додаткових UML-діаграм	41
2.4 Висновок до розділу 2	44
3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО МОДУЛЯ ПРОКСІ-ФЕРМИ.....	45
3.1 Обґрунтування вибору програмного засобу реалізації.....	45
3.2 Створення програмної реалізації.....	52
3.3 Тестування програмної реалізації.....	58
3.4 Висновок до розділу 3	61
ВИСНОВКИ.....	62
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	64
ДОДАТОК А ПРОТОКОЛ ПЕРЕВІРКИ НА ПЛАГІАТ.....	66
ДОДАТОК Б ІНСТРУКЦІЯ КОРИСТУВАЧА	67
ДОДАТОК В ЛІСТИНГ ПРОГРАМИ.....	69
ДОДАТОК Г ГРАФІЧНА ЧАСТИНА.....	74

ВСТУП

Актуальність теми дослідження полягає у тому, що станом на тепер розширюється сфера використання проксі-ферм. В них оператор мобільного зв'язку призначає IP-адреси таким пристроям, як мобільні модеми, смартфони або планшети. Мобільні проксі створюють враження, ніби їх користувач підключений до Інтернету через мобільну мережу передачі даних, маскуючи їхню справжню адресу.

Проксі-ферми використовуються в багатьох задачах. Для більш швидкого та безпечнішого підключення до Інтернету мобільні проксі використовують систему швидкого зберігання кеш-пам'яті, покращуючи таким чином підключення до Інтернету. Запобігають відстеженню файлів cookie. Дозволяють обійти географічні обмеження і доступ до обмеженого вмісту. Порівнюють інформацію про ціни та ринок. Компанії та рекламодавці можуть отримати вигоду від використання проксі-ферм для перевірки оголошень і забезпечення отримання органічного трафіку. Висока анонімність проксі-ферм дає можливість створювати «кілька» облікових записів і входити до них.

Мета роботи - це підвищення надійності, зменшення витрат та спрощення керування проксі-фермою.

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 програмного модуля;
- обґрунтування вибору програмного засобу реалізації;
- створення програмної реалізації;
- тестування програмної реалізації.

Об'єкт дослідження - процес створення програмного модуля проксі-ферми.

Предмет дослідження - алгоритми та засоби програмного модуля проксі-ферми.

Практична цінність одержаних результатів полягає у використанні користувачами проксі-ферми.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ВИКОРИСТАННЯ ПРОКСІ-ФЕРМИ

1.1 Аналіз галузі застосування проксі-ферм

Фермерство та парсинг - це дві речі, які неможливі без надійних проксі. Спроба створити обліковий запис через невідповідний проксі-сервер призведе до негайного та неминучого блокування. При цьому новачкові нелегко зорієнтуватися в різноманітті проксі-серверів, так як непрофесіоналу складно зрозуміти, які з них можна вважати хорошими, а які поганими. У цій статті буде висвітлено основні аспекти проксі та визначено, які з них відповідають вашим цілям [1].

Акаунт-фармінг - це створення фіктивних акаунтів у соціальній мережі, які неможливо відрізнити від справжніх, якими володіють звичайні користувачі. Такі акаунти створюються спеціально для рекламних кампаній. Очевидно, щоб розпочати прибуткову рекламну кампанію, вам потрібно мати надійний, справжній обліковий запис. Чим більше у вас таких акаунтів, тим більша ймовірність того, що ваші рекламні кампанії будуть успішними. Але створити та розіграти облікові записи недостатньо: є задіяні технології та інструменти.

Спочатку необхідно отримати основні елементи проксі (рис.1.1).

1. Отримати браузер.
- 2 Отримайте унікальні адреси електронної пошти, яких немає в чорному списку.
- 3 Знайти чисті номери мобільних телефонів.
- 4 Отримати хороший персональний проксі-сервер з домашніми IP-адресами.

Браузер із засобом виявлення спростить процеси, тому ви не заблукате в кількох облікових записах. Такий браузер створює унікальний відбиток для кожного наявного облікового запису (відбиток або відбиток містить

елементи конфігурації браузера, які дозволяють ідентифікувати користувача), а також запам'ятовує інформацію про логін-пароль.

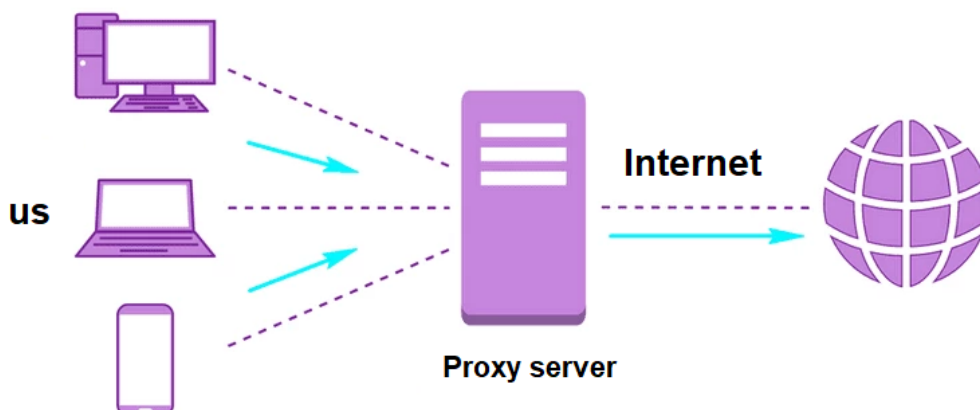


Рисунок 1.1 - Основні елементи проксі

У браузері можна призначити власний проксі-сервер для кожної відкритої вкладки, щоб ви могли одночасно відвідувати та редагувати кілька облікових записів однієї соціальної мережі. Звичайно, кожен обліковий запис має бути пов'язаний з унікальною адресою електронної пошти, номером мобільного телефону та IP-адресою, що відповідає вибраному GEO. Чисті електронні адреси та номери телефонів не є проблемою, але дійсна IP-адреса в бажаному GEO може бути складною.

Рішення цієї проблеми повністю залежить від правильного вибору проксі-сервера. Інформацію про проксі-фермерство, облік через проксі та проксі для Facebook можна знайти нижче.

Проксі-сервер - це проміжний сервер, через який примусово перенаправляється інтернет-трафік між комп'ютером користувача та цільовими серверами (де знаходяться відвідувані сайти або завантажені файли). Проксі-сервери найчастіше використовуються для обходу обмежень. Під час фармування афілійовані особи використовують проксі-сервери для маскування та заміни рідної IP-адреси.

Типи проксі-серверів. Проксі-сервери відрізняються один від одного багатьма технічними характеристиками і параметрами, але найважливішими аспектами є:

- тип даних протоколу передачі;
- анонімність;
- тип наданої IP-адреси.

Протокол передачі даних є найважливішим пунктом проксі-сервера. Зараз більшість проксі-серверів працюють за протоколами HTTP/HTTPS і SOCKS.

HTTP-проксі. Раніше це були найпоширеніші проксі-сервери. Оскільки протокол HTTP не забезпечує жодного захисту переданих даних від перехоплення та зчитування, HTTP-проксі вважаються небезпечними і тому не рекомендуються.

Проксі HTTPS. Проксі-сервери HTTPS є підтипом проксі-серверів HTTP. HTTPS-проксі використовують безпечний протокол передачі даних SSL/TLS, таким чином встановлюючи безпечне з'єднання між комп'ютером і проксі-сервером (рис.1.2)

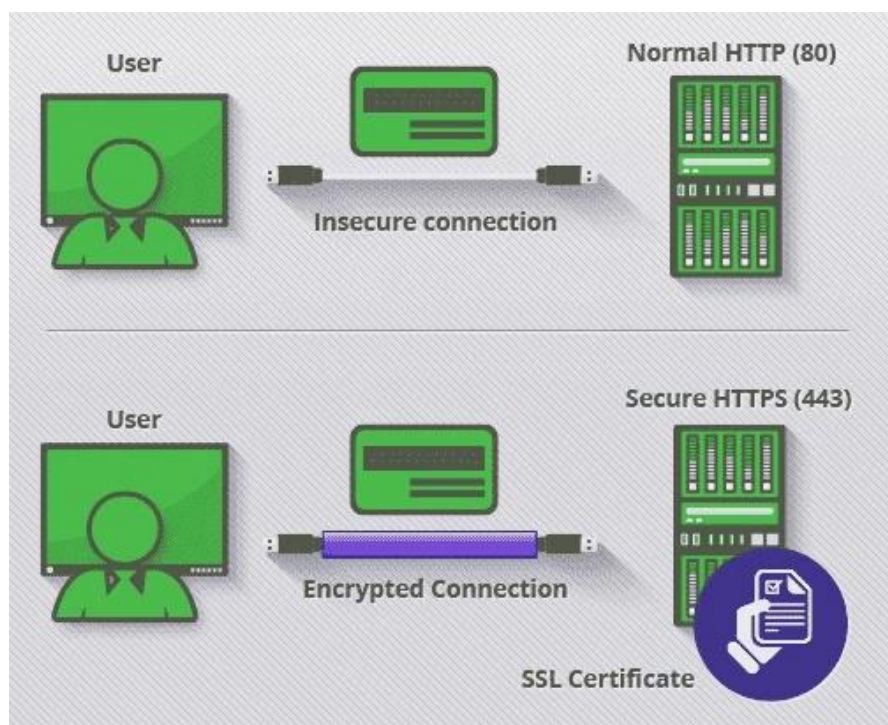


Рисунок 1.2 - Сертифікат SSL

Проксі HTTP/HTTPS розроблено виключно для трафіку браузера, тому вони можуть бути невідповідним варіантом для обробки трафіку, що надходить від інших програм або торрент-файлів.

Проксі SOCKS (рис.1.3). SOCKS є універсальним протоколом передачі даних. Проксі SOCKS обробляють і пересилають пакети даних майже всіх поширених мережових протоколів: HTTP, HTTPS, FTP, SMTP, POP3, IMAP тощо. На відміну від проксі-серверів HTTP/HTTPS, проксі SOCKS не спотворюють заголовки переданих пакетів. Це означає, що з точки зору цільового сервера всі дані надходять безпосередньо на нього з пристрою користувача.

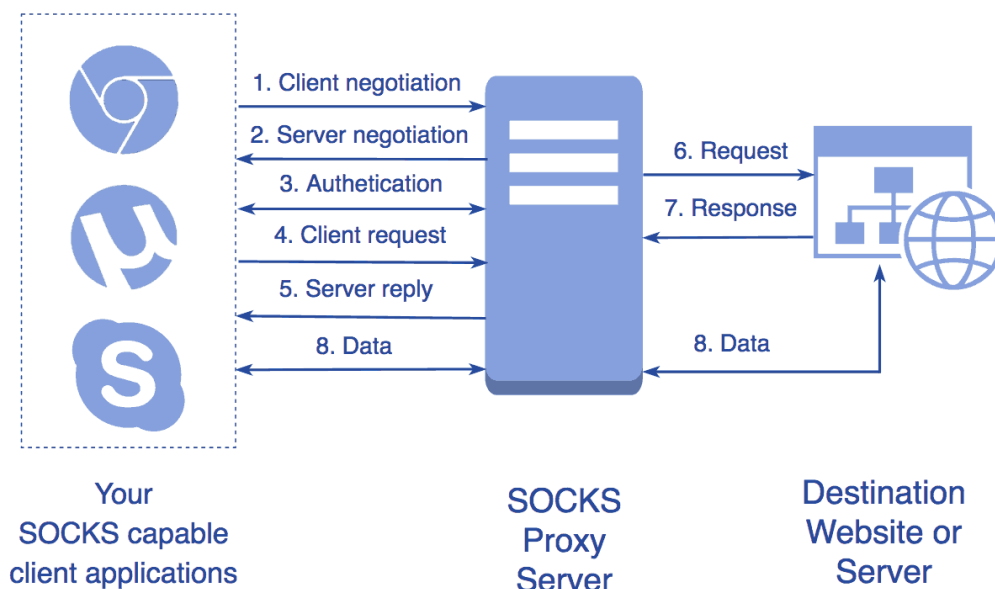


Рисунок 1.3 - Проксі SOCKS

Іншими словами, проксі-сервер невидимий для цільового сервера SOCKS, і саме це робить проксі-сервери SOCKS дуже анонімними за замовчуванням. Протоколи SOCKS4 і SOCKS5 є розширеннями протоколу SOCKS. На даний момент проксі-сервери SOCKS5 вважаються найбільш просунутими і універсальними, оскільки вони не тільки підтримують TCP, але і працюють з транспортним протоколом UDP. Коротше кажучи, вони

можуть обробляти трафік, що надходить майже від будь-якої програми, і ідеально підходять для завантаження торрент-файлів.

Неанонімні, анонімні та високоанонімні проксі-сервери.

За рівнем анонімності проксі-сервери можна розділити на прозорі, анонімні, високоанонімні та спотворюючі проксі.

Неанонімні або прозорі проксі-сервери передають реальну IP-адресу цільовому серверу.

Анонімні проксі-сервери замінюють (маскують) справжню IP-адресу користувача, але не приховують факту використання проксі.

Високоанонімні або елітні проксі-сервери замінюють справжню IP-адресу користувача та приховують факт використання проксі.

Спотворені проксі-сервери зазвичай не приховують, що проксі-сервер використовується, але вони передають хибну IP-адресу цільовому серверу.

Резидентні та серверні проксі IP-адреси.

Проксі-сервер - це звичайний сервер, розташований у центрі обробки даних. Він дуже швидкий, надійний, стабільний і цілком доступний. Однак цільовий Інтернет-ресурс зазвичай може відносно легко виявити, що відвідувач використовує проксі-сервер, навіть якщо він налаштований і працює як високоанонімний. Причина в тому, що IP-адреси проксі-серверів належать дата-центрам, а не інтернет-провайдерам.

Домашні проксі-сервери мають IP-адреси, які неможливо відрізнити від IP-адрес справжніх користувачів Інтернету. Домашні проксі-сервери дорожчі за серверні проксі та мають нижчу швидкість передачі даних.

Проксі-сервер зворотного підключення. Проксі-сервери Backconnect, також звані ротаційними проксі, змінюють свою IP-адресу через проміжок часу, указаний у налаштуваннях. Проксі-сервер зворотного з'єднання також можна запрограмувати так, щоб кожен запит сервера надсилався з нової (зміненої) IP-адреси.

Мобільні проксі-сервери (рис.1.4). Мобільні проксі є підтипом резидентних проксі. Вони використовують ті ж IP-адреси, що й сервери

мобільних операторів. Мобільні проксі чудово підходять для створення облікових записів зі смартфонів і планшетів.

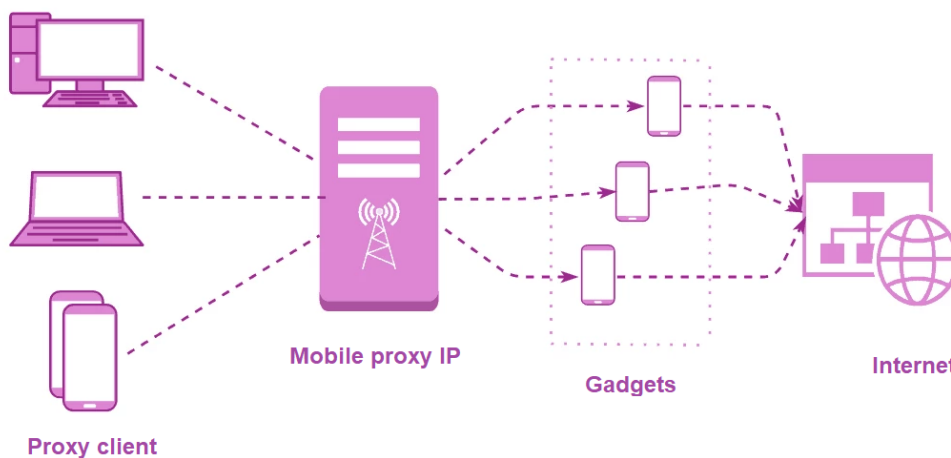


Рисунок 1.4 – Структура мобільної проксі ферми

Розглянемо приватні та публічні проксі.

Відкриті (загальнодоступні) проксі-сервери не вимагають пароля і доступні всім для безкоштовного використання.

Доступ до приватних проксі захищений паролями. Приватні проксі-сервери створюються компаніями для власних потреб і для оренди. Приватні проксі-сервери, в свою чергу, діляться на виділені та загальні.

Виділені та спільні проксі. Виділений проксі - сервер належить одному користувачеві. Спільний проксі використовується кількома користувачами одночасно.

Анонімайзери - це публічні безкоштовні проксі-сервери з веб-інтерфейсом. З точки зору користувача, анонімайзер - це звичайний сайт з адресним рядком, куди вводиться адреса цільового інтернет-ресурсу. Природно, анонімайзери не підходять для парсингу або фарму акаунтів.

Порти проксі-сервера.

У HTTP/HTTPS-проксі найчастіше використовують порти 80, 3128 і 8080.

Проксі-сервери SOCKS використовують 1080 або 1081, але насправді номер порту може бути будь-яким: його визначає власник проксі-сервера під

час налаштування. Таким чином, щоб почати використовувати проксі-сервер, вам необхідно знати хоча б його IP-адресу та номер порту. Тому в списках доступних проксі-серверів для більшості служб вони представлені як пара IP-адреси та номера порту, наприклад:

123.45.67.890:8080 або 149.19.224.38:3128.

Таким чином, наприклад, 123.45.67.890 - IP-адреса проксі-сервера; 8080 - використаний порт.

Проксі для афілійованих осіб. Лише дуже анонімні проксі-сервери HTTPS і SOCKS придатні для ферми. Крім того, IP-адреса проксі-сервера не повинна бути підозрілою з точки зору цільового сайту. Адміністрація багатьох популярних інтернет-ресурсів постійно збирає IP-адреси проксі-серверів і заносить їх у чорний список. Тому перед початком використання проксі-сервера рекомендується перевірити, чи є його IP-адреса в чорному списку цільового інтернет-ресурсу.

Перевірка IP. Якщо треба дізнатися, чи включена IP-адреса вибраного проксі-сервера до чорних списків, треба відвідати будь-який із сайтів перевірки. Природно, потрібно отримати доступ до перевірки через проксі-сервер, який тестується. (Також можна перевірити свою IP-адресу, відвідавши засіб перевірки без проксі.)

Після відвідування чекера рекомендується видалити всі файли cookie, пов'язані з ним, або використовувати режим приватного браузера з самого початку, оскільки, з точки зору адміністраторів соціальних мереж і пошукових систем, просте відвідування сайту чекера може викликати підозру.

Аналізатор конфіденційності надає детальну інформацію про вашу видиму IP-адресу та перевіряє відбитки пальців браузера на автентичність.

Pixelscan.net - найпростіший ресурс, який перевіряє достовірність IP-адреси та відбитків браузера[2]. Якщо перевірка не виявить жодних проблем, результат тесту буде відповідним. Для більш детальної перевірки IP-адреси треба перейти на сторінку IP Check ресурсу pixelscan.net.

Whoer.net - це проста служба перевірки IP-адрес і браузера [3]. Якщо IP-адреса нерезидентна, служба виявить, що використовується проксі-сервер, і вкаже, чи є він у чорному списку.

Fake.Vision - потужний ресурс для тестування IP-адрес і відбитків пальців браузера [4]. Щоб розпочати повний тест, треба прокрутити сторінку вниз і натиснути «Почати розширені тести», щоб отримати повну інформацію про поточну IP-адресу та браузер, який використовується. Проблемні пункти будуть виділені червоним кольором. Натиснути кнопку «ЧОРНИЙ СПИСОК», щоб переконатися, що поточна IP-адреса не включена до жодної бази даних підозрілих IP-адрес.

CheckerProxy.net - сервіс для перевірки працездатності та анонімності проксі-серверів [5]. Можна тестувати кілька проксі-серверів одночасно. Щоб перевірити, введіть IP-адреси та номери портів проксі-серверів, які тестуються, у відповідне поле, розділивши їх двокрапкою, і натиснути кнопку «КОНТРОЛЬНИЙ СПИСОК».

Getipintel.net - оцінює ймовірність того, що перевірена IP-адреса належить проксі-серверу або вузлу TOR [6]. Результатом перевірки буде число від 0 до 1. Чим ближче результат до одиниці, тим вища ймовірність того, що IP-адреса, яка перевіряється, є поганою та належить до проксі-сервера або вузла мережі TOR. Щоб почати тестування, треба ввести IP-адресу в поле «Введіть дійсну IP-адресу» (номер порту вказувати не потрібно) і натиснути кнопку «Пошук».

Зверніть увагу, що багато компаній, які надають проксі-сервери в оренду, мають на своїх сайтах власні чекери. Однак завжди краще скористатися послугами незалежного сервісу.

Проксідля Facebook і Instagram [7]. Програмне забезпечення Facebook відстежує та блокує користувачів, які заходять на сайт через проксі. Очевидно, що лише високоанонімні проксі-сервери SOCKS і HTTPS з домашніми IP-адресами підходять для створення облікових записів Facebook.

Розташування вибраного проксі має відповідати мовним налаштуванням браузера. Детальну інформацію про Instagram можна знайти тут.

Проксі для пошукових систем. Пошукові системи зазвичай не блокують трафік, що надходить із проксі-серверів. Однак, виявивши підозрілу активність, пов'язану з певною IP-адресою, Google пропонує перевірити капчу. Тому довго не можна буде автоматично аналізувати результати Google з однієї IP-адреси. Отже, для цієї мети вам потрібно вибрати проксі-сервери Backconnect (ротаційні), а також мобільні проксі-сервери.

Проксі для месенджерів. Оскільки месенджерами найчастіше користуються власники смартфонів і планшетів, найкращим варіантом є мобільні проксі-сервери.

Проксі-сервер для аналізу та атак грубої сили. Щоб регулярно збирати великі обсяги даних, необхідно постійно змінювати IP-адреси. Тому для цієї мети найкраще підходять проксі Backconnect (ротаційні) і, меншою мірою, мобільні проксі.

Можливо орендувати свій проксі на відомих майданчиках.

LuminatiProxyNetwork - один із найпопулярніших сервісів оренди проксі-серверів у світі. Десятки мільйонів віртуальних проксі-серверів цієї компанії знаходяться практично в кожній країні. В оренду доступні проксі-сервери різних типів: домашні, мобільні, серверні.

Proxy-Seller.com - це гідний сервіс, який володіє мільйонами проксі-серверів у десятках країн. Сервіс пропонує проксі-сервер, адаптований спеціально для реєстрації та роботи в соціальних мережах і парсингу сайту.

Microleaves – є чи не найбільшою у світі службою оренди проксі-серверів. Окрім внутрішніх проксі, також пропонуються проксі Backconnect (обертаючі).

Проксі-сервери Storm - надають проксі-сервери з IP-адресами сервера та локальними IP-адресами, а також проксі-сервери Backconnect (ротаційні).

Zyte Smart Proxy, раніше відомий як Crawlera, — це стара, надійна служба, яка змінила назву після ребрендингу. Сервіс пропонує розумні (ротаційні) проксі-сервери Backconnect, які автоматично змінюють IP-адресу, якщо попередній IP-адресу заблоковано цільовим веб-ресурсом. Мабуть, найкращий доступний вибір.

NetNut - пропонує величезну кількість серверних і домашніх проксі-серверів, також є ротаційні проксі-сервери з домашніми IP-адресами.

Партнерський маркетинг - це аналіз облікового запису, ведення господарства, обхід і повна анонімність.

Партнери, які працюють з Facebook або GoogleAdwords, знають, як швидко система блокує людей за порушення правил. Іноді афілійовані особи, які вже кілька разів були заблоковані платформами, виявляють, що стає важче - або зовсім неможливо - створювати нові облікові записи або входити в придбані.

Однак є спосіб уникнути таких проблем: проксі.

Мобільні проксі замінюють справжній IP адресою стільникового оператора. Принцип простий: кілька запитів, надісланих на сервер, не можуть бути заблоковані. Однак, треба мати на увазі, що це не спам; це дії з пристроїв з однаковим IP. Постійні зміни даних створюють ефект присутності реального користувача, тим самим обманюючи алгоритми платформи.

Переваги мобільного проксі:

- мобільні проксі приховують IP, це забезпечує безпеку в мережі;
- повна анонімність, дані не передаються, що захищає ваш обліковий запис;
- без обмежень руху, використовуйте стільки, скільки потрібно;
- динамічна зміна IP, ця функція дозволяє працювати з кількома обліковими записами одночасно, алгоритм самостійно змінює дані;

- низький ризик бану, платформи не встановлюють обмежень, які б завдали шкоди тисячам користувачів.

Мобільний проксі дозволяє вирішити широкий клас задач.

1. Створити кілька облікових записів, які система не запідозрить як навмисне збільшення облікових записів.
2. Читерство стає можливим, оскільки збільшення кількості лайків і переглядів більше не є проблемою. Можливо обійти обмеження, якщо знаєте, як зробити проксі самостійно.
3. Можливо вдатися до синтаксичного аналізу, який передбачає безпечний збір даних під час імітації активності смартфона.
4. Фармування облікових записів і запуск реклами з кількох облікових записів допоможуть вам обійти обмеження.

Створення власного проксі проти проксі, який можна купити має ряд переваг.

1. Контролювати трафік.
2. Заощаджувати кошти, хоча ціни варіюються в залежності від завдань, які потрібно виконати, і кількості IP-адрес, які потрібні, але це вдвічі дешевше.
3. Мобільні проксі-сервери можна адаптувати для конкретного GEO.
4. На власних проксі можна заробляти, здаючи їх в оренду.

Розглянемо чотири варіанти створення особистих мобільних проксі,

Перший варіант. Використання власного сервера. Необхідне обладнання:

- USB концентратори;
- модеми з прошивкою;
- сім карти, обирати треба операторів з безлімітним інтернетом;
- сервер, можливо використовувати свій ПК;
- подовжувачі для підключення USB.

Послідовність дій така. Підключити USB до ПК, а модеми до концентратора. Розмістити їх на відстані, щоб не «заважали» один одному.

Змінити веб-адреси в модемах вручну. Встановити програму для створення проксі. Інструкцію по створенню мобільного проксі можна знайти на офіційному сайті обраної програми.

Переваги: IP-адреси на основі 4G надійніші та швидші, ніж 3G.

Недоліки: оплата Інтернету за кожен модем не завжди надійна.

Другий варіант. Використання віртуальних машин (рис.1.5).



Рисунок 1.5 - Структура віртуальної машини

Віртуальна машина - це хмара, яка імітує ПК. Її можна встановити на свій комп'ютер, щоб не обмежуватися лише одним vds. Всі дії відбуваються в хмарі, а її запити сприймаються як ті, що надходять від реального ПК. Кожна машина матиме унікальний відбиток пристрою.

Послідовність дій така. Придбати віртуальну машину (vds); підключити її до Linux; вибрати мережу; надати root-права (надають додаткам додаткові можливості); налаштувати підключення за допомогою коду; підключити модем до ПК, додати його до vds і перезапустити; запустити IP.

Недоліки: віртуальні машини вимагають досвіду та знань, тому новачкам краще обрати простіший спосіб, щоб уникнути помилок.

Третій варіант. Використання модему (рис.1.6)



Рисунок 1.6 - Модеми для створення проксі-ферми

Підключення здійснюється через мобільну адресу за допомогою модему з SIM-картою. Щоб змінити IP, потрібно перепідключити з'єднання в інтерфейсі або відключити і включити його в порту. Це працює лише з 3G. У мережі 4G використовувати треба іншу прошивку. Для зміни IP потрібно:

- 1) відключитися від мережі;
- 2) підключитися до 3G, потім знову підключитися до 4G;
- 3) потім повторно підключіться до мережі.

Недоліки: постійне ручне перемикання. Якщо є лише кілька IP - нічого страшного, але афіліатам, які працюють з великими обсягами, такий спосіб абсолютно не підходить.

Переваги: простий, підходить для 10 IP в день.

Четвертий варіант. Використання Android Proxy. Це можуть зробити початківці: цей варіант не потребує особливих знань кодування та не займає багато часу, щоб розібратися. Вам потрібен лише Android 5.0 і програма, наприклад, iProxy.online. Сам процес простий: зареєструйтеся та дотримуйтесь інструкцій. Процес займає не більше 5-10 хвилин. Треба просто підтверджувати запити системи, яка сама все зробить.

IP-адреси змінюються в режимі польоту двома способами:

- вручну (включати, вимикати);
- віддалено через root-права.

Переваги: безкоштовне розгортання, численні функції (блокування, фільтрування URL-адрес тощо).

Розглянемо нескладний приклад створення проксі на модемах (рис1.7).



Рисунок 1.7 - Мобільна ферма на модемах

Із обладнання потрібно:

- сервер (ПК), наприклад, 4-ядерний міні-комп'ютер, підключаємо до нього 5 модемів;
- модеми 4G/LTE;
- USB-концентратор;
- Інтернет;
- маршрутизатор, найпростіший, якого вистачить на 5 модемів, якщо планується підключити більше модемів у майбутньому, то потрібен модем із гігабітними портами.

Покрокова інструкція така.

- Зареєструйтеся на веб-сайті проксі і придбати ліцензію.
- Встановіть проксі. Процес необхідно проводити з правами root.
- Встановіть утиліту розпакування та розпакуйте архів.
- Перейти до /home/proxy, установіть дозвіл на виконання та запустіть сценарій встановлення.
- Під час встановлення ввести своє ім'я користувача, пароль та електронну адресу. Електронна пошта потрібна для отримання інформації про стан пристроїв, серверів та інших параметрів.
- Налаштуйте службу SMTP (пошта). Можна користуватися безкоштовною електронною поштою.
- Після виконання цих кроків надійде повідомлення про успішне встановлення.
- 4G проксі готовий.

Фармування проксі. Ферма в даному випадку - це IP-мережа. Афілійовані особи можуть отримати прибуток від проксі-ферми кількома способами: створення облікових записів вигідне, але спробуйте також здавати їх в оренду за ціною, нижчою від ринкової [8] .

Критерії обирання сервісу створення проксі.

1. GEO. Не всі служби забезпечують широке покриття карти. Зверніть увагу на обмеження.
2. Наявність пробного періоду. Спробуйте вибрати сервіс, який пропонує пробний період, щоб оцінити функції та можливості та перевірити параметри.
3. Ліцензія або підписка. Програма може пропонувати план передплати або ліцензію на використання.
4. Ціна. Є безкоштовні послуги та ті, що пропонують тарифні плани. Остаточний вибір залежить від кінцевої мети.

1.2 Аналіз відомих методів розв'язання задач використання проксі-ферми

Великі ресурси використовують складні алгоритми для ідентифікації ботів, аналізуючи дії користувачів на трьох рівнях:

- мережа - перевірка IP-адреси, відбитка ОС, MTU та чорних списків;
- програмне забезпечення - аналіз Canvas, WebGL, файлів cookie, шрифтів, відбитків аудіо та інших даних;
- поведінка - дії перевіряються на збіг зі статистикою використання сайту реальними користувачами; багаторазові звернення до сервера викликають блокування.

Мобільні проксі можуть допомогти уникнути блокування на першому рівні перевірки. Якщо ви дізнаєтеся, як налаштувати мобільні проксі, то згодом зрозумієте, наскільки легко працювати з кількома профілями та запускати автоматичні дії (наприклад, аналізувати конкурентів). Дії користувача виглядатимуть природно, тому системи захисту від ботів на інших сайтах не спрацюють.

Мобільні проксі - це інструменти, які виступають посередниками між вами та вибраними джерелами трафіку, наприклад, соціальними мережами, через які ви надсилаєте трафік до пропозиції [9]. Ресурси не можуть визначити тип обладнання, IP-адресу, місцезнаходження та інші параметри. Сервер замінює їх неправдивою інформацією (рис.1.8).

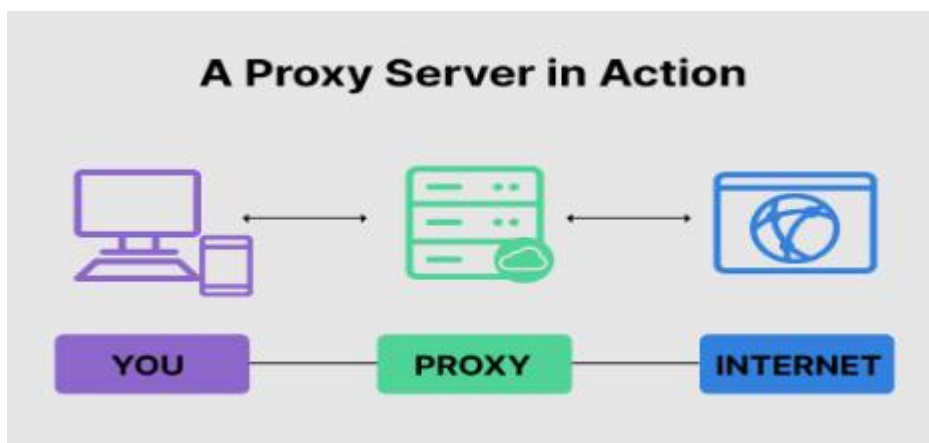


Рисунок 1.8 - Основний принцип проксі

Важливі особливості при розробці проксі-ферми.

Ретельно проаналізувати потреби параметрів проксі для задоволення параметрів арбітражу трафіку, оскільки він буде працюватимете з кількома GEO і виконуватимете різні дії.

Звернути увагу на такі параметри:

1. Чим більше покриття GEO вам запропонують, тим краще. Вам не доведеться змінювати його та платити знову, щоб імітувати доступ до мережі з певного місця.
2. Краще брати необмежену кількість варіантів, але дивитися на ціну. Безкоштовні підключення без обмежень розчарують вас низькою швидкістю.
3. Ціна. Найчастіше платять за трафік. Іноді вони стягують плату за запит. Будьте готові платити за додаткові послуги, наприклад, приватний канал.
4. Відгуки. Читати коментарі, пов'язані з обраною послугою, на сторонніх сайтах.
5. Зміна IP. Чим вища частота змін, тим вищий ризик того, що проксі має невеликий пул адрес.
6. Рівень обслуговування. Вибирайте тільки надійних і відомих постачальників з кришталевою репутацією і відмінними рекомендаціями.

Можуть запропонувати підключення за кількома протоколами, найпоширенішими варіантами є HTTPS і SOCKS5. Найкраще вибрати ті мобільні проксі, які працюють із Socks, оскільки це більш безпечний варіант, який допомагає зберегти анонімність у мережі, передаючи дані в зашифрованому вигляді.

Якщо користувач працює з Facebook або Instagram, то треба обрати проксі з SOCKS5 ще з однієї причини. Коли цей протокол працює, алгоритми соцмережі не зрозуміють, чи змінено свій IP. По суті, вони вважатимуть його

справжнім. Крім того, версія 5 Socks дозволяє програмам використовувати анонімайзер у випадках, коли розробник не включив таку опцію.

Найкраще обрати мобільні проксі. За ціною вигідніше підключати мобільні проксі: так можна отримати тисячі IP безкоштовно. В інтернеті навіть можна знайти безкоштовні версії таких проксі, але їх швидкість досягає 5 Мб/с, чого недостатньо для повноцінного серфінгу.

Перш ніж працювати з мобільними проксі, необхідно вибрати найкращий сервіс, який підходить для ваших цілей. Для афілійованих осіб важливі наступні аспекти: регулярна зміна адрес, можливість робити це за запитом, широкий вибір GEO та операторів зв'язку.

Мобільний проксі пропонує 5 варіантів підключення:

- загальний;
- загальний без очікування (без таймауту при зміні IP);
- приватний;
- приватний без очікування;
- багатопортовий.

Останнє краще підходить партнерам, оскільки кожне нове підключення надходить з іншої IP-адреси. Користуватися такою опцією через додатки зручно, але не вийде з браузером, оскільки для повного завантаження сторінок потрібно не менше 25 підключень.

У приватних проксі ви можете налаштувати частоту зміни IP-адреси або запитати її. Якщо вибрати простий варіант, пауза повторного підключення триває 5 секунд. Натакому каналі працюватиме лише один користувач.

У приватних каналах є більше варіантів налаштувань, але в публічних каналах ви можете вибрати 1-2 протоколи та швидкість обертання. Тариф можна змінити в будь-який час. Служба підтримки працює практично цілодобово (вночі бувають затримки відповіді).

Наприклад, найпростіший варіант організації проксі-ферми (рис.1.9):

- купити канал від \$1,5 на день;
- випробувальний термін: 2 години;
- швидкість від 5 Мб/с;
- є можливість змінити IP за запитом (посиланням);
- відбувається зміна IP в інтервалі від 2 до 5 хвилин;
- забезпечити до 5 користувачів на канал (публічний).

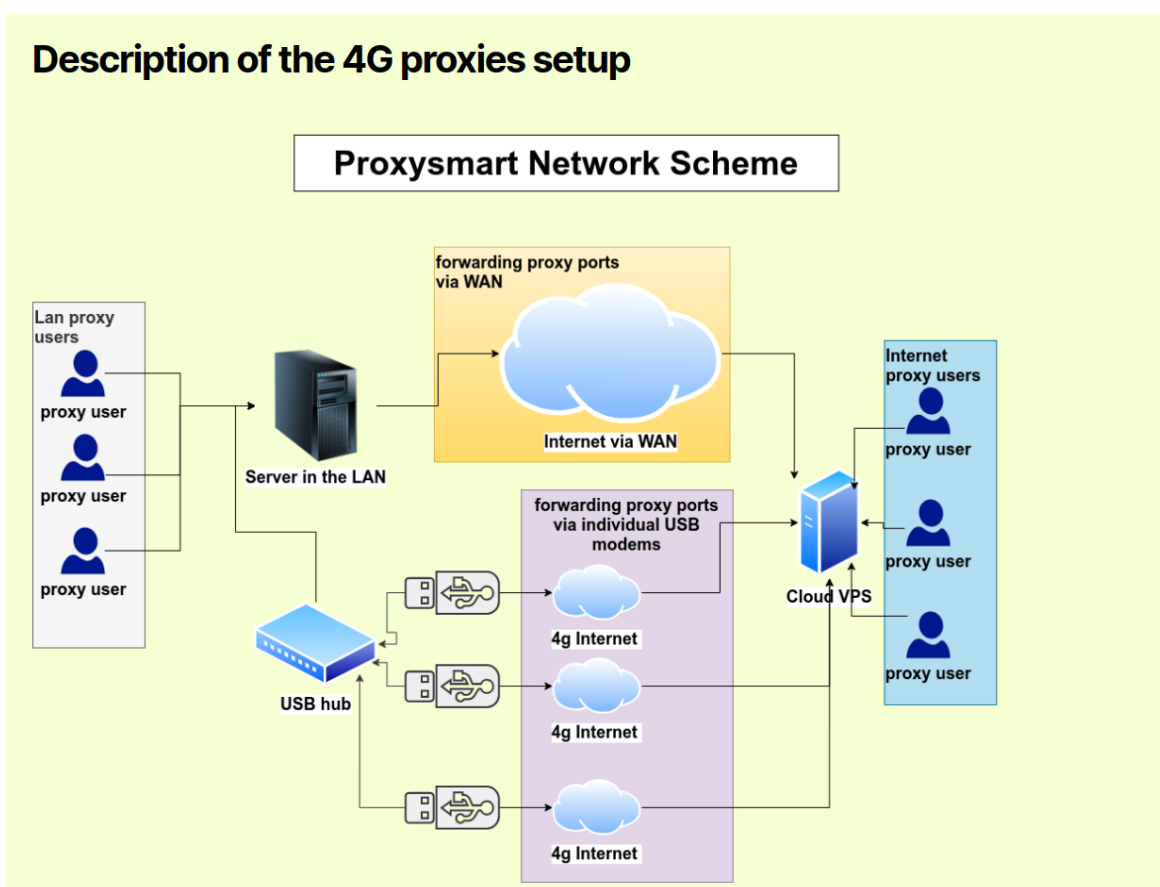


Рисунок 1.9 - Варіант організації проксі-ферми

1.3 Постановка задачі

Проксі-сервери є важливими інструментами для різноманітних дій в Інтернеті, таких як сканування веб-сторінок, аналіз даних і анонімність. Два популярних типи проксі - це мобільні проксі та проксі-сервери для мешканців. Хоча обидва служать для маскуванню вашої IP-адреси та забезпечення анонімності, між ними є значні відмінності. У цій статті ми

докладно порівняємо мобільні проксі-сервери та проксі-сервери для мешканців, досліджуючи їхні функції, переваги та випадки використання.

Особливості мобільних проксі. Визначення та функціональність - ознайомлення з мобільними проксі-серверами та тим, як вони направляють інтернет-трафік через мобільні пристрої та мобільні мережі. Мобільні IP-адреси — пояснення використання реальних мобільних IP-адрес, пов'язаних із постачальниками мобільних мереж, що додає додатковий рівень автентичності та зменшує шанси бути виявленим як проксі-користувач.

Переваги та випадки використання мобільних проксі.

1. Обхід блокування IP-проксі — мобільні проксі-сервери ефективно обходять механізми блокування IP-адрес, які зазвичай використовуються веб-сайтами та службами, дозволяючи користувачам отримувати доступ до вмісту, який інакше був би обмежений.
2. Гнучкість геолокації — Мобільні проксі-сервери пропонують можливість підключатися з різних геолокацій, що робить їх ідеальними для завдань, які вимагають доступу до конкретного місця, наприклад для локалізованих маркетингових кампаній або досліджень у певному регіоні.

Особливості домашніх проксі. Визначення та функціональність - ознайомлення з домашніми проксі-серверами, які направляють інтернет-трафік через домашні IP-адреси, призначені реальним домогосподарствам. Місцеві IP-адреси - пояснюється, що домашні проксі-сервери використовують IP-адреси, пов'язані з постачальниками послуг Інтернету (ISP), забезпечуючи рівень легітимності та надійності, щоб уникнути виявлення.

Переваги та випадки використання житлових проксі.

1. Анонімність і конфіденційність — домашні проксі пропонують вищий рівень анонімності порівняно з іншими типами проксі, оскільки вони використовують IP-адреси, пов'язані з реальними домашніми користувачами.

2. Збір даних і веб-скрапінг — домашні проксі-сервери дуже ефективні для веб-скрапінгу та збору даних, оскільки вони імітують реальну поведінку користувачів і з меншою ймовірністю активують заходи проти ботів.

Порівняння характеристик.

1. Автентичність - мобільні проксі-сервери використовують реальні мобільні IP-адреси, тоді як домашні проксі використовують IP-адреси, пов'язані з місцевими провайдерами Інтернету.
2. Ротація IP-адрес — мобільні проксі-сервери часто забезпечують динамічну ротацію IP-адрес, дозволяючи користувачам перемикатися між різними мобільними IP-адресами, тоді як домашні проксі-сервери пропонують статичні IP-адреси.
3. Геолокація — Мобільні проксі-сервери забезпечують більшу гнучкість щодо геолокації, дозволяючи користувачам підключатися з різних регіонів або країн, тоді як домашні проксі-сервери обмежені розташуванням місцевих IP-адрес.
4. Швидкість з'єднання. Мобільні проксі-сервери зазвичай забезпечують більш високу швидкість з'єднання через свою залежність від мобільних мереж, тоді як домашні проксі-сервери можуть мати різну швидкість залежно від якості домашнього інтернет-з'єднання.

Вибір правильного проксі-сервера для ваших потреб.

1. Цільові веб-сайти чи служби - необхідно розглянути конкретні веб-сайти чи служби, до яких вам потрібно отримати доступ, і визначте, чи мають вони переваги чи обмеження щодо типів проксі.
2. Вимоги до геолокації - якщо завдання включають дані або вміст, пов'язані з геолокацією, мобільні проксі-сервери можуть бути більш придатними через їхню гнучкість підключення з різних регіонів.

Отже. Мобільні проксі-сервери та проксі-сервери для мешканців є цінними інструментами для досягнення анонімності та доступу до обмеженого вмісту в Інтернеті. У той час як мобільні проксі-сервери чудово обходять механізми блокування IP-адрес і забезпечують гнучкість геолокації,

житлові проксі-сервери пропонують вищий рівень анонімності та добре підходять для завдань збору даних і веб-збирання. Розуміючи унікальні функції, переваги та варіанти використання кожного типу проксі, ви можете прийняти обґрунтоване рішення щодо того, який із них найкраще відповідає вашим конкретним потребам. Незалежно від того, чи обираєте ви мобільні проксі-сервери чи домашні проксі-сервери, використання цих інструментів у вашій онлайн-діяльності може значно покращити вашу конфіденційність, безпеку та доступ до Інтернету.

1.4 Висновок до розділу 1

Створення власної мобільної проксі-ферми є достатньо актуальною задачею, розв'язання якої надає ряд переваг, а саме: контроль трафіку; заощадження коштів; адаптування для конкретного GEO; можливість заробітку на їх в оренді. У якості завдання обираємо створенні програмного забезпечення для розгортання мобільної проксі-ферми.

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

2.1 Обґрунтування засобів розробки програмного модуля

У якості смартфонів для побудови проксі-ферми з динамічною зміною ІР-адреси треба використовувати смартфони, які підтримують 4G.

Основні елементи мережі стандарту 4G/LTE (рис.2.1).

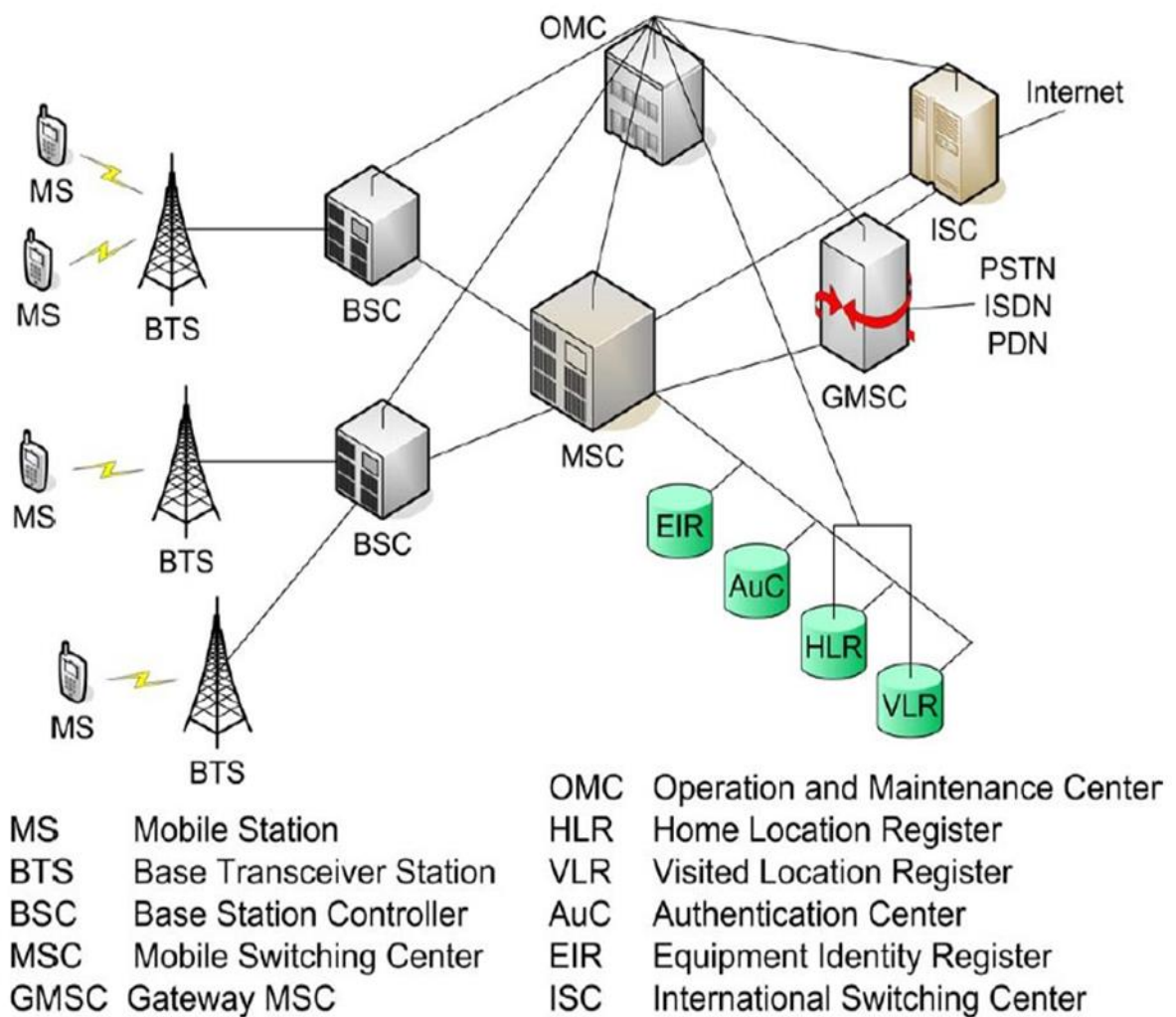


Рисунок 2.1 - Архітектура мережі GSM 4G

Serving SAE Gateway або просто ServingGateway (SGW) - обслуговуючий шлюз мережі LTE [7, 8]. Призначений для обробки і маршрутизації пакетних даних, що надходять з/в підсистему базових станцій. SGW має пряме сполучення з мережами другого і третього покоління того ж

оператора, що спрощує передачу з'єднання в них з причин погіршення зони покриття, перевантажень і т.п. У SGW немає функції кому та ції каналів для голосових з'єднань, тому що в LTE вся інформація, включаючи голос комутується і передається за допомогою пакетів.

Postman - це API-платформа для створення та використання API. Postman спрощує кожен крок життєвого циклу API та оптимізує співпрацю, щоб можливо було створювати швидші API.

Public Data Network SAE Gateway або просто PDN Gateway (PGW) - шлюз до мереж передачі даних інших операторів для мережі LTE. Основне завдання PGW полягає в маршрутизації трафіку мережі LTE до інших мереж передачі даних, таких як Інтернет, а також мереж GSM, UMTS. Mobility Management Entity (MME) - вузол управління мобільністю мережі стільникового зв'язку стандарту LTE. Призначений для обробки сигналізації, переважно пов'язаної з управлінням мобільністю абонентів в мережі. Home Subscriber Server (HSS) - сервер абонентських даних мережі стільникового зв'язку стандарту LTE. Являє собою велику базу даних і призначений для зберігання даних про абонентів. Крім того, HSS генерує дані, необхідні для здійснення процедур шифрування, аутентифікації тощо.

Мережа LTE може включати один або кілька HSS. Кількість HSS залежить від географічної структури мережі і числа абонентів.

Policy and Charging Rules Function (PCRF) – елемент мережі стільникового зв'язку стандарту LTE, який відповідає за управління нарахуванням плати за надані послуги зв'язку, а також за якість з'єднань відповідно до заданих конкретному абоненту характеристиками.

Організація інформаційних каналів.

Для того, щоб дані могли бути транспортовані через інтерфейс LTE, використовуються різні «канали». Вони використовуються для того, щоб виділяти різні типи даних і дозволити їм транспортуватися через мережу доступу більше ефективно. Використання декількох каналів забезпечує

інтерфейс більш високого рівня в рамках протоколу LTE і включають більш чітку і визначену сегрегацію даних.

Є три категорії, в якіможуть бути згруповані різні канали передачі даних.

1. Логічні канали - надає послуги середнього рівня управління доступом MAC (Medium Access Control) в межах структури протоколу LTE. Логічні канали на кшталт переданої інформації діляться на логічні канали управління і логічні канали трафіку. Логічні канали керування використовуються для передачі різних сигнальних та інформаційних повідомлень. На логічних каналах трафіку передають призначені для користувача дані.
2. Транспортні канали - транспортні канали фізичного рівня пропонують передачу інформації в MAC і вище. Інформацію логічних каналів після бробки на RLC / MAC рівнях розміщують в транспортних каналах для подальшої передачі по радіоінтерфейсу в фізичних каналах. Транспортний канал визначає як і з якими характеристиками відбувається передача інформації по радіоінтерфейсу. Інформаційні повідомлення на транспортному рівні розбиваються на транспортні блоки. У кожному часовому інтервалі передачі (Transmission TimeInterval, TTI) радіопослуга передають хоча б один транспортний блок. При використанні технології MIMO можлива передача до чотирьох блоків в одному TTI.
3. Фізичні канали - це канали передачі, які переносять призначені для користувача дані і керуючі повідомлення. Вони змінюються між висхідним і спадним потоками, оскільки кожен з них має різні вимоги і діє по-своєму.

Переваги Lte такі.

1. Швидкість завантаження з мережі до 300 Мбіт / с.
2. Швидкість віддачі до 75 Мбіт / с
3. Низька затримка передачі даних - може бути знижена до 5 мс

4. Смугапропускання частот від 1,4 до 20 МГц
5. Частотне (FDD) і тимчасове (TDD) поділканалівзв'язку
6. Поліпшенняпідтримкимобільності
7. Економіяенергії: для низхідноїлініїзв'язку - OFDMA, для висхідної - SC-FDMA
8. У кожнійсоті 5 МГц підтримується не менше 200 активнихклієнтів
9. Радіоінтерфейскомутаціїпакетів

Мережева архітектура LTE-Advanced (рис.2.2)

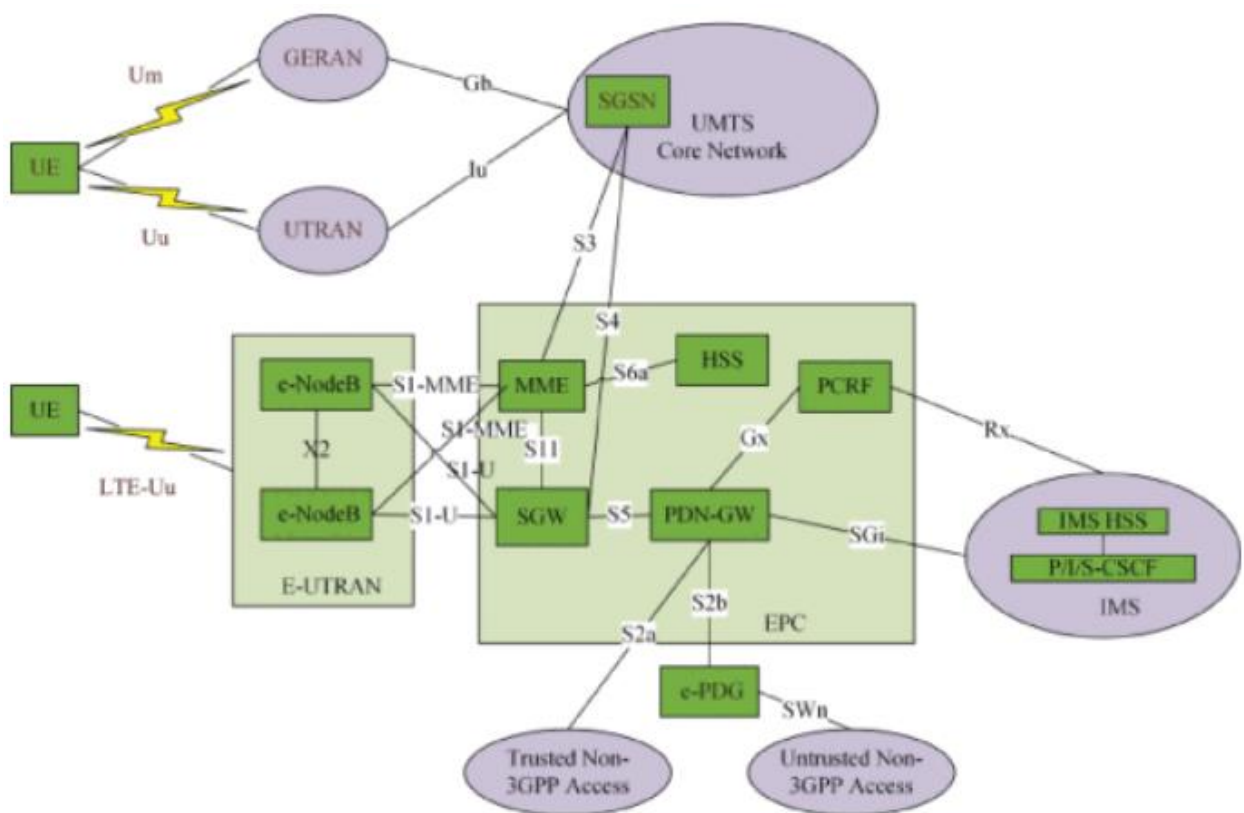


Рисунок 2.2 - Мережева архітектура LTE-Advanced

Архітектура LTE-Advanced складається з мережі радіодоступу та базової мережі. Мережа радіодоступу називається E-UTRAN яка виникла з вихідної наземної мережі радіодоступу (UTRAN). UMTS – це абревіатура від Universal Mobile Telecommunication System. E-UTRAN складається з декількох Вдосконалених NodeB (E-NodeB), які мають функціональні

можливості NodeB і беруть на себе більшість функцій контролера радіомережі в UTRAN. Користувацьке Обладнання(UE) і E-NodeB зв'язані через радіоінтерфейс(Uu).

E-NodeB з'єднуються між собою через інтерфейс X2 і в основному виконує наступні функції: керування радіоресурсами, стиснення IP-заголовка і шифрування користувацьких даних, підключення до об'єкту управління мобільністю (MME) через інтерфейс S1-MME(для досягнення мобільності управління та проходження сигналізації None Access Stratum) і з'єднання з обслуговуючим шлюзом (SGW) через інтерфейс S1-U.

Базова мережа LTE-A називається мережею EPC. Вона забезпечує підключення до декількох мереж доступу різного роду, які містяться в мережах 3GPP (E-UTRAN, GERAN, UTRAN) і мережі доступу не-3GPP (WiMAX, CDMA2000 і т.д.). EPC складається з MME, домашнього серверу абонентів (HSS), SGW, Шлюз Мережі Пакетних Даних (PDN-GW) і функція правил політики та тарифікації (PCRF). EPC забезпечує розділення поверхні управління і поверхні користувача. MME виконує функції поверхні управління, а SGW виконує функції поверхні користувача.

MME виконує функції сигналізації NAS, управління Безпеки Рівня Доступу(AS), обробка мобільності в стані очікування і т.д. В той час як HSS надає послуги для базової мережі LTE-A і мережі IMS в якості центральної бази даних. MME підключається до HSS через інтерфейс S6a для передачі даних автентифікації. Основні функції SGW-маршрутизації та пересилання. PDN-GW в основному призначений для розподілу IP-адресів UE, фільтрації пакетів і моніторингу.

PCRF виконує тарифікацію на основі потоків і керування мережею у відношенні знаходження потоку службових даних для гарантії Якості Обслуговування(QoS). Trusted Non-3GPP Access та Untrusted Non-3GPP Access являють собою мережі IP доступу, специфікації яких виходять за рамки 3GPP. Функціональність вдосконаленого шлюзу пакетних Даних (e-PDG) включає в себе виділення і транспортування віддаленого IP-адресу,

прив'язку локальної мобільності в ненадійних мережах доступу без підтримки 3GPP тощо.

В LTE-A, мережа IMS використовується для обробки послуг з комутацією пакетів, таких як VoLTE. Крім того, мережі GSM і UMTS підтримують процедури відновлення з комутацією каналів (CSFB). Від може спрацювати, коли IMS не використовується. IMS – це потужна структура, розгорнута мережею LTE-A для надавання різних типів мультимедійних послуг, таких як VoLTE, SMS, потокове відео тощо.

В даний час поряд з мобільними мережами другого, третього і четвертого поколінь широкого поширення набули високошвидкісні локальні мережі стандарту Wi-Fi. При цьому з'явилися спільні базові станції, призначені як для роботи в мережах Wi-Fi, так і в мережах стільникового зв'язку. Все це дозволяє обслуговувати абонентів із застосуванням різних варіантів доступу.

У зв'язку з цим в архітектурі SAE передбачені механізми вибору найбільш зручної мережі передачі даних для надання абоненту необхідних послуг. При цьому всі сучасні мобільні термінали вибирають, при наявності декількох мереж, передачу трафіку на дозволений користувачем мережі Wi-Fi, що зазвичай супроводжується перемиканням з однієї технології на іншу і відповідно передачею даних про користувача між мережами.

Підключення з'єднання між мережею LTE і іншою мережею стандарту 3GPP при встановленні голосового виклику відбувається за допомогою взаємодії логічного елемента MME з сервером MSC по інтерфейсу Sv. У разі викликів з мережі LTE в мережу комутації каналів (CS-домен); і за допомогою взаємодії логічного елемента MME з вузлом SGSN по інтерфейсу S3 в разі голосового виклику з мережі LTE в мережу комутації пакетів (PS-домен).

Взаємодія мережі LTE з мережами не-3GPP поділяється на взаємодію з мережами з гарантованою безпекою – «надійними» і взаємодію з мережами, безпека яких не гарантована – «ненадійними». «Надійними» є мобільні

мережі інших стандартів, «ненадійними» – загальнодоступні IP-мережі Інтернету. Взаємодія мережі LTE з «надійними» мережами стандартів не-3GPP здійснюється через шлюз P-GW, взаємодія з «ненадійними» мережами – через шлюз ePDG.

Ускладнення характеру трафіку, зокрема зниження мовного навантаження в порівнянні з обсягом мультимедійних повідомлень при організації різних видів мобільного доступу призводить до того, що безпека передачі інформації, може бути забезпечена тільки при використанні ефективних методів підвищення пропускної здатності, бо саме при бездротовому доступі можуть виникати різкі перекоси навантаження через стохастичне переміщення абонентів по зонах мобільної мережі, при цьому всі протоколи та схеми взаємодії є відкритими, а передача даних здійснюється на базі протоколу IP.

Розглянемо докладніше процедуру ідентифікації мобільного терміналу. Ідентифікація мобільного терміналу виконується модулем управління мобільністю MME, та/або домашнім сервером абонентських даних HSS, та/або пакетним шлюзом P-GW з допомогою реєстра ідентифікації (Equipment Identity Register - EIR) і призначена для автентифікації використовуваного обладнання мобільного терміналу. Перевірка справжності мобільного терміналу дозволяє виключити використання в мережі вкрадених або дефектних мобільних терміналів. Ідентифікація мобільного терміналу ініціюється модулем MME шляхом передачі міжнародного ідентифікатора мобільного терміналу IMEI реєстра ідентифікації EIR для подальшої перевірки автентичності терміналу, а потім аналізу відповіді реєстра EIR для виконання відповідної дії. Процедура автентифікації мобільного терміналу UE показана на рисунку 2.3.

Для використання в проксі-фермі треба використовувати мобільні проксі, які працюють на найактуальніших протоколах: HTTPS(s) і SOCKS 5. При використанні мобільного проксі на протоколі SOCKS 5 перебування користувача в мережі буде повністю анонімним. Реальна IP адреса

користувача буде надійно захищена.

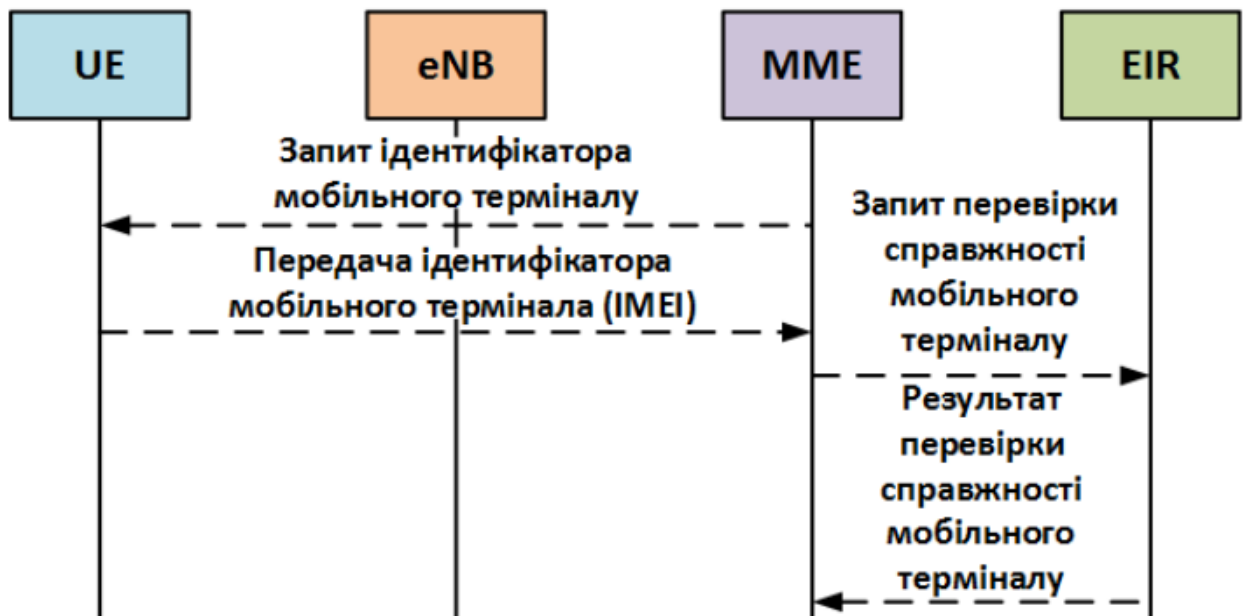


Рисунок 2.3 – Процедура автентифікації мобільного термінала

Розглянемо особливості мобільних IP-адрес.

У зв'язку зі специфікою роботи мобільного інтернету, мобільна IP-адреса є унікальним за своїми параметрами технічним рішенням. Так як у мобільного оператора величезна кількість абонентів які користуються послугами інтернету, видати кожному абоненту статичний IP-адресу технічно неможливо, але що найголовніше - не рентабельно для самого мобільного оператора [10].

Набагато зручніше створити єдиний пул IP-адрес, і надавати його декільком абонентам одночасно. З цього випливає що на одному мобільному IP можуть одночасно працювати сотні або навіть тисячі абонентів мобільного інтернету. Через це великим інтернет-ресурсам та соціальних мережам потрібні коригування в їх бан-системах і алгоритмах обмеження доступу. Якщо великий ресурс, наприклад Фейсбук, заблокує один з мобільних IP-адрес, то під блокуванням можуть виявитися тисячі користувачів.

Розглянемо особливості мобільних проксі.

Мобільні проксі це програмно-технічний комплекс, який виконує роль посередника між інтернетом і користувачем. Мобільні проксі забезпечують обмін даними, шляхом передачі веб-запитів користувача через мобільний інтернет від GSM-оператора.

В такому форматі роботи, всі запити користувача в мережі інтернет, будуть проходити через мобільного-оператора з використанням мобільної IP-адреси. Це дозволить користуватися інтернетом отримуючи всі переваги роботи в мобільній мережі GSM-оператора.

Особливості роботи мобільних проксі.

Підходів до реалізації мобільного-проксі досить багато. Багато сервісів і компанії використовують різні підходи для створення мобільних проксі. Найпродуктивнішим і надійним є пере направлення трафіку з реальних сім-карт на внутрішні сервера.

В такому форматі роботи безумовним плюсом є використання реальних мобільних пристроїв при роботі з сім-картою і мережею мобільного оператора в цілому. Мобільний пристрій абсолютно валідно проходить аутентифікацію в мережі мобільного оператора, з занесенням IMEI в реєстр вузла мобільного зв'язку. Після цього проксі-сервер починає керувати передачею даних через цімобільні пристрої, дозволяючи пропускати будь-які запити через мобільний інтернет в мережі GSM-оператора.

Мобільні проксі-сервера дозволяють створити якісну систему контролю доступу, можуть обмежувати пропускну здатність на окремому каналі, створювати динамічні пули мобільних IP-адрес і реалізовувати роботу мобільних проксі на різних протоколах (SOCKS5, HTTPS).

Відмінність мобільних проксі від звичайних.

Звичайні проксі, як правило, використовують IP адреси хостинг-провайдерів або хмарних сервісів, які за визначенням не можуть належати фізичній особі. А значить, подібну активність просто не може проявляти реальний користувач. Такий IP в 90% випадків стає причиною блокування при найменшій підозрі на "підозрілу активність" з точки зору бан-систем і

алгоритмів великих інтернет-ресурсів.

Алгоритми виявляють використовуваний адресу і застосовує конкретні дії: блокує або знижує рівень трасту до аккаунту.

Мобільний оператор виділяє користувачам адреси з власного пулу, маючи в розпорядженні обмежену кількість адрес IPv4. Виходить, кілька мільйонів користувачів ділять між собою всього пару сотень IP, що розподіляються між усіма пристроями за допомогою NAT.

Отже, якщо алгоритм блокує IP адресу мобільного оператора, то це дія відбивається на звичайних користувачах, які працюють під тим же IP. А втрата цільової аудиторії вкрай небажана для соцмереж або великих порталів.

HTTP мобільні проксі - це найпоширеніший вид проксі. Основне призначення - організація роботи браузерів і інших програм, що використовують TCP протокол. Стандартні порти 80, 8080, 3128.

Принцип роботи: програма або браузер посилає запит проксі-сервера на відкриття певного URL ресурсу. Проксі-сервер отримує дані з запитуваного ресурсу і віддає ці дані вашого браузеру.

HTTP проксі дозволяють:

- кешувати завантажені файли для збільшення швидкості відкриття веб-сайтів;
- отримати доступ до певних ресурсів доступ до яких у вас обмежений;
- фільтрувати дані.

HTTPS мобільні проксі - фактично це HTTP-проксі, буква "S" в даному випадку означає "secure" (захищений) з підтримкою захищеного SSL з'єднання. Ці проксі застосовуються, коли потрібно передати секретну інформацію (наприклад, логіни / паролі, номери пластикових карт). Стандартні порти 80, 8080, 3128.

При використанні звичайного HTTP проксі всю інформацію, передану через нього, можна перехопити за допомогою самого проксі або на більш

низькому рівні.

Наприклад, всі Інтернет-провайдери перехоплюють і логірують абсолютно всю вашу активність в мережі Інтернет. Ці логи зберігаються провайдером і є обґрунтованим доказом ваших дій в мережі Інтернет. Тому для безпеки особистих даних застосовується протокол HTTPS, при цьому весь переданий трафік шифрується стійким алгоритмом.

Принцип роботи: проксі-сервер з'єднується з ресурсом і ваш трафік шифрується. При такому методі відсутня можливість дізнатися, яка саме інформація передається через проксі-сервер (це обмежує застосування проксі як фільтра). Також в процесі шифрування і дешифрування проксі участі не бере. Цим займається клієнтська програма (браузер) і цільової сервер.

Таким чином, HTTPS проксі займається пасивної передачею зашифрованої інформації і не робить ніякої обробки інформації, що передається. Такий метод роботи дозволяє використовувати HTTPS проксі для передачі практично будь-якого TCP-протоколу. Тобто HTTPS проксі може використовуватися і як POP3, SMTP, IMAP, NNTP проксі.

На сьогоднішній день Socksпроксі - найпрогресивніший протокол передачі інформації. Протокол Socks розроблявся для програм, які не підтримують використання проксі безпосередньо.

Даний протокол пережив безліч змін і на даний момент використовуються дві версії протоколу:

- Socks 4 підтримує тільки TCP з'єднання;
- Socks 5 підтримує TCP, UDP, авторизацію за логіном і паролем.

Socks не займається модерацією HTTP-заголовків. Socks-сервер буде передавати інформацію через себе в чистому вигляді. Тому все Socks сервери є анонімними.

Socksпроксі передає інформацію про ваш IP адресу. Веб-сайт не зможе визначити використання проксі. З'єднання з веб-сайтом буде абсолютно прозорим, також як якби ви працювали з ним безпосередньо. При цьому веб-сайт буде бачити IP адреса проксі, а не ваш реальний IP адреса.

Socks підтримує всі протоколи, включаючи HTTP, HTTPS, FTP [11, 12].
HTTP Proxy.

Це найпопулярніший вид Proxy. Застосовується майже повсюдно.
Дозволяє вирішити кілька актуальних завдань:

- кешування (стиснення) даних, що прискорює обмін інформацією;
- фільтрація контенту за допомогою блокування сайтів певної спрямованості;
- відключення реклами та рекламних повідомлень.

HTTPS Proxy.

Проксі захищає персональні дані користувача за допомогою шифрування вихідного трафіку. Сервер такого виду особливо часто застосовується, якщо потрібно зберегти конфіденційну інформацію, наприклад, про реквізити банківських карток або електронних гаманців. Важливо розуміти, що HTTPS-сервер не є повноцінною захисною системою з точки зору професійного хакера, хоча вирішує деякі питання безпеки на досить успішному рівні.

Socks (SOCKS) Proxy.

Найсучасніший і прогресивний проксі. Перша версія мережевого протоколу розроблена досить давно в 1992 році. Сьогодні застосовується четверта та п'ята. Socks-проксі вважається найкращим засобом забезпечення конфіденційності користувача під час відвідування інтернету. Веб-ресурс не тільки не отримує IP-адреси відвідувача, але і не бачить, що контакт відбувається з завданням Proxy.

2.2 Розробка діаграми класів

Відмінність мобільних проксі від звичайних полягає в наступному.

Звичайні проксі, як правило, використовують IP-адреси хостинг-провайдерів або хмарних сервісів, які за визначенням не можуть належати фізичній особі. А значить, подібну активність просто не може проявляти реальний користувач. Такий IP в 90% випадків стає причиною блокування

при найменшій підозрі на "підозрілу активність" з точки зору бан-систем і алгоритмів великих інтернет-ресурсів.

UML діаграма послідовності додавання інтервалу операції проксі-сервером (рис.2.4)

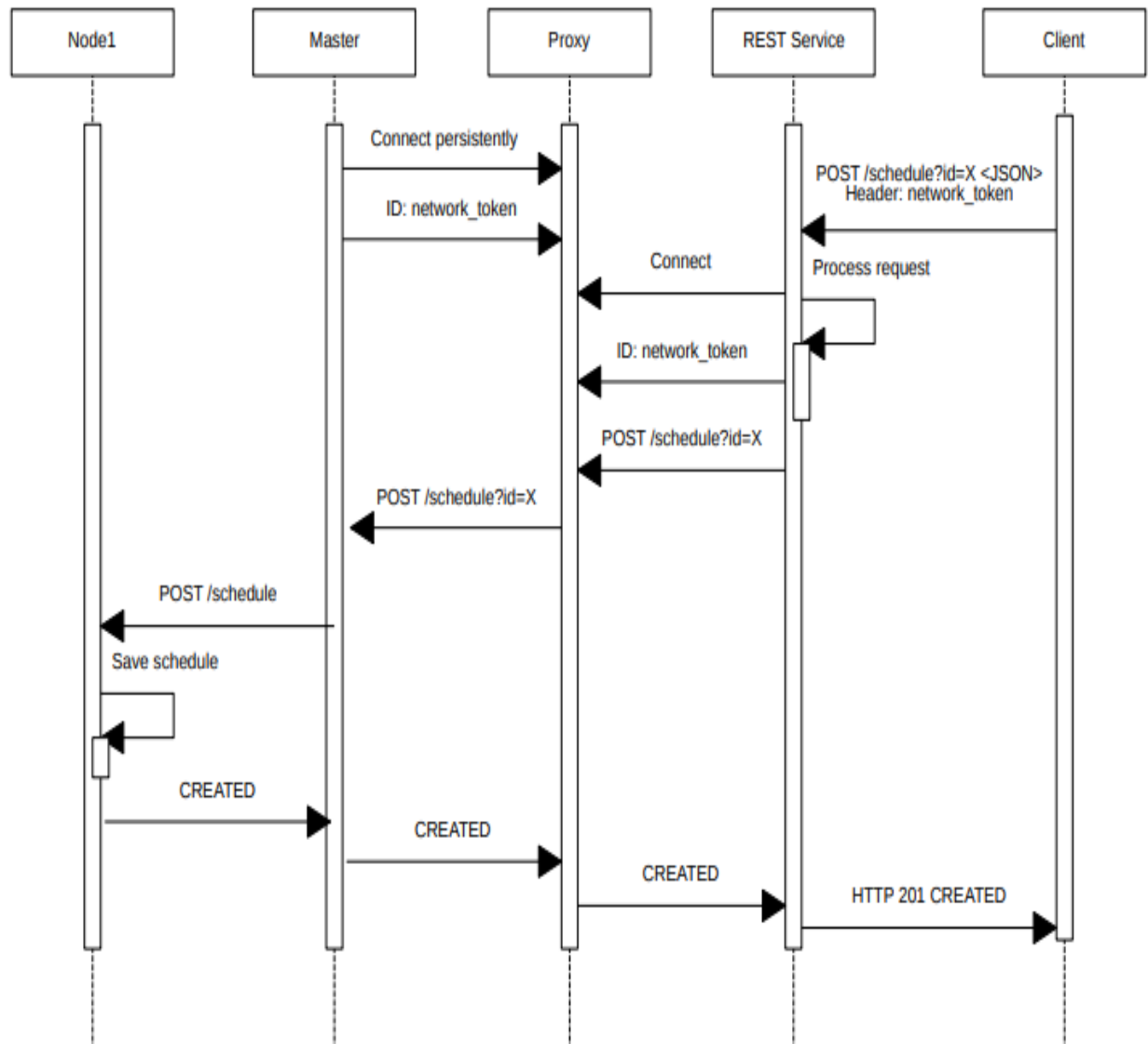


Рисунок 2.4 - UML діаграма послідовності додавання інтервалу операції проксі-сервером

Алгоритми виявляють використовуваний адресу і застосовує конкретні дії: блокує або знижує рівень трасту до аккаунту.

Мобільний оператор виділяє користувачам адреси з власного пулу, маючи в розпорядженні обмежену кількість адрес IPv4. Виходить, кілька мільйонів користувачів ділять між собою всього пару сотень IP, що

розподіляються між усіма пристроями за допомогою NAT.

Отже, якщо алгоритм блокує IP адресу мобільного оператора, то це дія відбивається на звичайних користувачах, які працюють під тим же IP. А втрата цільової аудиторії вкрай небажана для соцмереж або великих порталів.

HTTP мобільні проксі - це найпоширеніший вид проксі. Основне призначення - організація роботи браузерів і інших програм, що використовують TCP протокол. Стандартні порти 80, 8080, 3128.

Принцип роботи: програма або браузер посилає запит проксі-сервера на відкриття певного URL ресурсу. Проксі-сервер отримує дані з запитуваного ресурсу і віддає ці дані вашого браузеру.

HTTP проксі дозволяють:

- кешувати завантажені файли для збільшення швидкості відкриття веб-сайтів;
- отримати доступ до певних ресурсів доступ до яких у вас обмежений;
- фільтрувати дані.

HTTPS мобільні проксі - фактично це HTTP-проксі, буква "S" в даному випадку означає "secure" (захищений) з підтримкою захищеного SSL з'єднання. Ці проксі застосовуються, коли потрібно передати секретну інформацію (наприклад, логіни / паролі, номери пластикових карт). Стандартні порти 80, 8080, 3128.

При використанні звичайного HTTP проксі всю інформацію, передану через нього, можна перехопити за допомогою самого проксі або на більш низькому рівні.

Таким чином, HTTPS проксі займається пасивної передачею зашифрованої інформації і не робить ніякої обробки інформації, що передається. Такий метод роботи дозволяє використовувати HTTPS проксі для передачі практично будь-якого TCP-протоколу. Тобто HTTPS проксі може використовуватися і як POP3, SMTP, IMAP, NNTP проксі.

2.3 Розробка додатковихUML-діаграм

Принцип роботи: проксі-сервер з'єднується з ресурсом і трафік шифрується. При такому методі відсутня можливість дізнатися, яка саме інформація передається через проксі-сервер (це обмежує застосування проксі як фільтра). Також в процесі шифрування і дешифрування проксі участі не бере. Цим займається клієнтська програма (браузер) і цільової сервер (рис.2.5).

Наприклад, всі Інтернет-провайдери перехоплюють і логірують абсолютно всю вашу активність в мережі Інтернет. Ці логи зберігаються провайдером і є обґрунтованим доказом ваших дій в мережі Інтернет. Тому для безпеки особистих даних застосовується протокол HTTPS, при цьому весь переданий трафік шифрується стійким алгоритмом.

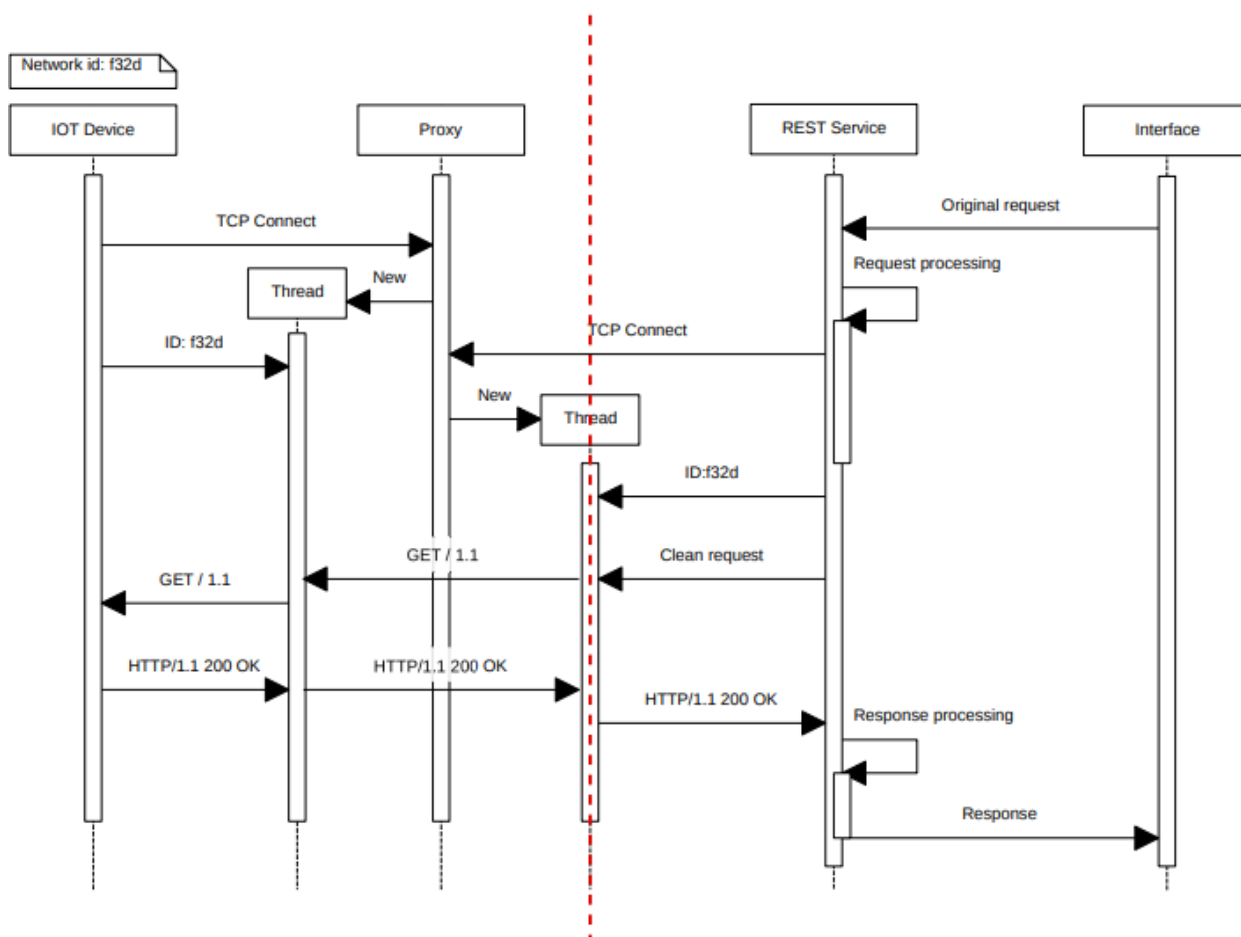


Рисунок 2.5 - UML діаграма послідовності прикладу роботи проксі-сервера

На сьогоднішній день Sockспроксі - найпрогресивніший протокол передачі інформації. Протокол Socks розроблявся для програм, які не підтримують використання проксі безпосередньо.

Даний протокол пережив безліч змін і на даний момент використовуються дві версії протоколу:

- Socks 4 підтримує тільки TCP з'єднання;
- Socks 5 підтримує TCP, UDP, авторизацію за логіном і паролем.

Socks не займається модерацією HTTP-заголовків. Socks-сервер буде передавати інформацію через себе в чистому вигляді. Тому все Socks сервери є анонімними.

Sockспроксі передає інформацію про ваш IP адресу. Веб-сайт не зможе визначити використання проксі. З'єднання з веб-сайтом буде абсолютно прозорим, також як якби ви працювали з ним безпосередньо. При цьому веб-сайт буде бачити IP адреса проксі, а не ваш реальний IP адреса.

Socks підтримує всі протоколи, включаючи HTTP, HTTPS, FTP [11, 12].
HTTP Proxy.

Це найпопулярніший вид Proxy. Застосовується майже повсюдно. Дозволяє вирішити кілька актуальних завдань:

- кешування (стиснення) даних, що прискорює обмін інформацією;
- фільтрація контенту за допомогою блокування сайтів певної спрямованості;
- відключення реклами та рекламних повідомлень.

HTTPS Proxy.

Проксі захищає персональні дані користувача за допомогою шифрування вихідного трафіку. Сервер такого виду особливо часто застосовується, якщо потрібно зберегти конфіденційну інформацію, наприклад, про реквізити банківських карток або електронних гаманців. Важливо розуміти, що HTTPS-сервер не є повноцінною захисною системою з точки зору професійного хакера, хоча вирішує деякі питання безпеки на

досить успішному рівні.

Socks (SOCKS) Proxy.

Найсучасніший і прогресивний проксі. Перша версія мережевого протоколу розроблена досить давно в 1992 році. Сьогодні застосовується четверта та п'ята. Socks-проксі вважається найкращим засобом забезпечення конфіденційності користувача під час відвідування інтернету. Веб-ресурс не тільки не отримує ІР-адреси відвідувача, але і не бачить, що контакт відбувається з завданням Proxy.

На рисунку 2.6 надано приклад обробки запиту GET, яке було надіслано веб-клієнтом.

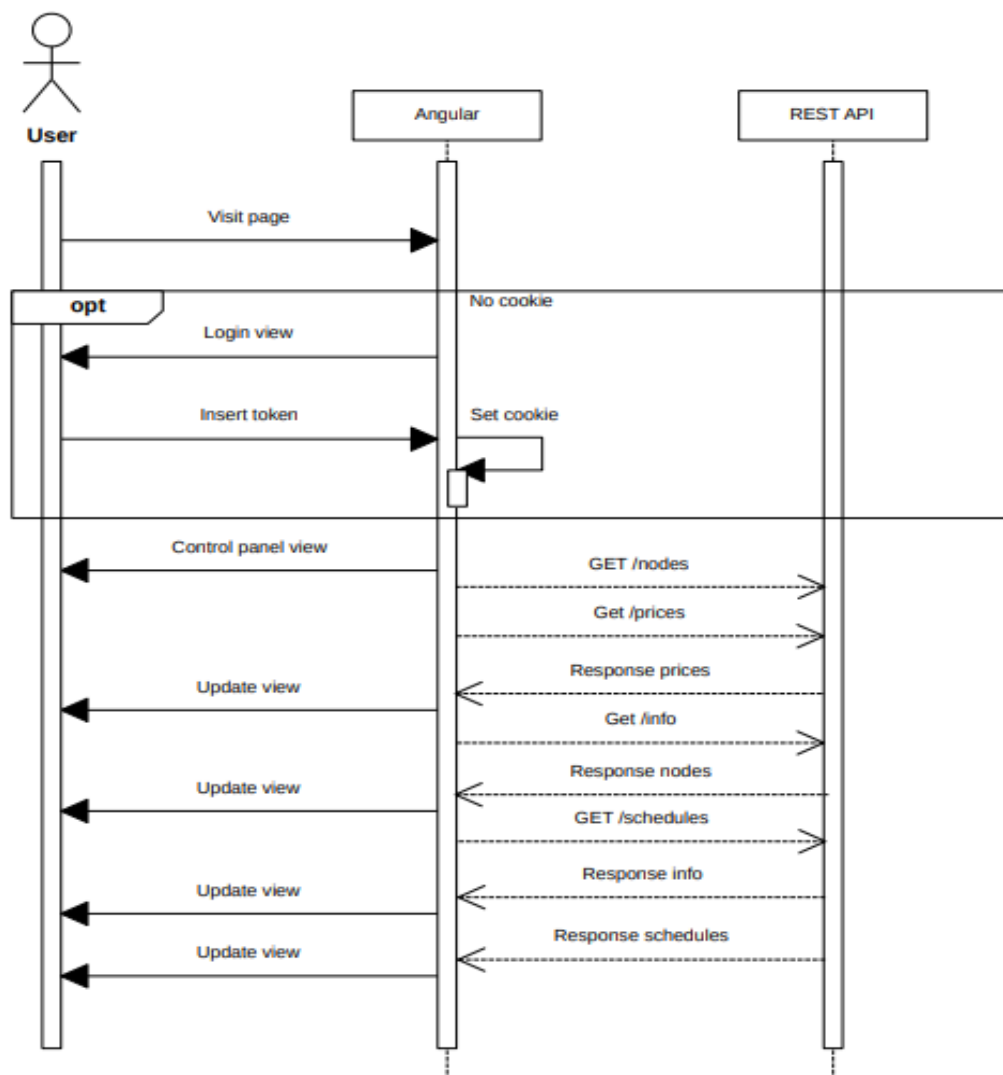


Рисунок 2.6 - Діаграма послідовності обробки веб-додатків

2.4 Висновок до розділу 2

Розробка програмного модуля проксі-ферми 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 засоби розробки програмного модуля проксі-ферми. Розроблені UML діаграми послідовності роботи проксі-ферми. Деталізовані класи програмного забезпечення програмного модуля проксі-ферми.

3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО МОДУЛЯ ПРОКСІ-ФЕРМИ

3.1 Обґрунтування вибору програмного засобу реалізації

Розглянемо та порівняємо мови програмування, які можливо було б використати для розв'язання задачі створення проксі-ферми.

1. Java. Посідає чи не найперше місце в списку найкращих мов програмування для мобільних додатків, Java є найбільш використовуваною мовою розробки додатків. У 1994 році Джеймс Гослінг розробив мову програмування Java в компанії Sun Microsystems, яка зараз належить Oracle. Це стало поширеним сьогодні. Відповідно до опитування VersionEye, яке відстежує бібліотеки програмного забезпечення з відкритим кодом, Java має найбільше проектів з відкритим кодом серед розробників. Java є найбільш бажаною мовою розробки мобільних додатків, оскільки вона працює на всіх платформах, включаючи всім відомий Android. Багато компаній, що займаються розробкою мобільних додатків, віддають перевагу Java, оскільки вона пропонує розробникам додатків гнучкість швидкої розробки мобільних додатків без помилок. Оскільки Java є найвідомішою мовою розробки додатків для Android, вона також має підтримку великої та активної спільноти користувачів, яка допоможе вам подолати будь-яку проблему. Окрім програм для Android, Java широко використовується для розробки ігор, вбудованого простору, веб-сайтів, серверних програм тощо. Крім того, Java можна запускати як у вікні браузера, так і на віртуальній машині, яка не вимагає браузера. Ось чому це дуже гнучка мова програмування з точки зору повторного використання коду.

Ключові особливості: його можна використовувати для розробки крос-платформних програм, мова об'єктно-орієнтована, API підтримуються, кодування стало простіше завдяки надійним інтегрованим середовищам розробки.

2. Swift — це мова програмування, представлена Apple у 2014 році, і ця мова спеціально розроблена для роботи з платформами iOS і OS X. Якщо й існує мова програмування, яка може змінити майбутнє, то це Swift. Swift розроблено для усунення ймовірності вразливості безпеки. Тепер розробники мобільних додатків увімкнули Swift. Багато компаній спеціалізуються на розробці найсучасніших мобільних додатків, використовуючи цю мову. Відповідно до TNW, Swift вважається мовою, яка розвивається найшвидше. Google також розглядає можливість зробити Swift своєю першокласною мовою замість Java. Цікаво, якщо Google перейде на Swift, попит на програми Swift стрімко зросте, і не буде іншої конкуруючої мови. Одним із найвідоміших прикладів програм Swift є програма SlideShare для iOS, повністю створена на Swift. Наразі Swift доступний лише для розробки під iOS, але оскільки він працює на Linux і є відкритим вихідним кодом, ним може користуватися кожен.

Ключові особливості: Swift - це мова з відкритим кодом, спрощена версія Objective-C, простий у кодуванні, обслуговування надзвичайно просте, це майбутнє розробки iOS, він вимагає менше кодування порівняно з іншими мовами.

3. Javascript. Сьогодні JavaScript став однією з провідних мов розробки мобільних додатків. Це найкраще для кросплатформної розробки додатків і веб-перегляду. React Native — це фреймворк на основі JavaScript, який широко використовується для розробки кросплатформних мобільних додатків. Вільям Тінг якось сказав: «JavaScript залишатиметься актуальним, доки люди користуються Інтернетом». Він безперебійно працює поза браузером в інших середовищах і може бути скомпільований з багатьох мов програмування. Згідно з опитуванням Stack Overflow у 2018 році, JavaScript був найпопулярнішою мовою програмування протягом шести років серед розробників. І більше 50% з них скористалися нею в 2023 році. Мова програмування є легкою та є широкою перевагою через її сумісність з усіма браузерами та гнучкість синтаксису.

4. C++. Це найбільш підходяща та надійна мова програмування для створення мобільних додатків для Android і Windows, і, головним чином для низькорівневого програмування, вона все ще є переважною мовою на платформах для розробників мобільних додатків. Як надійна мова програмування C++ дозволяє розробляти мобільні програми практично для будь-яких цілей на кожній платформі. Можливо, це не модна мова, але вона домінувала в епоху програмування ще до приходу революції смартфонів. Попит на C++ був завжди. Йдеться не лише про розробку мобільних додатків, але й про потужну мову, яка використовується у всьому: від фінансів до виробництва, банківської справи та багатьох інших сфер.

Ключові особливості: C++ простий і ефективний, він об'єктно-орієнтований, велика бібліотека, портативний, надзвичайно швидкий.

5. Python. Був створений Гвідо ван Россумом і випущений у 1991 році. Це одна з найкращих мов програмування на ринку, яка допомагає розробникам Python створювати потужні веб- та мобільні програми. Мова досить проста у вивченні та застосуванні, що робить її кращим вибором серед розробників. Найкраще в Python те, що він вважається ідеальним для розробки програм як для Android, так і для iOS. Вона чудова з точки зору масштабованості та читабельності, що робить її широко використовуваною мовою для кросплатформних мобільних додатків. Python також відкриває двері для багатопоточності та багатопроцесорності, що покращує продуктивність програми та реакцію на команди. В іншому опитуванні Python стає найбільш популярною мовою програмування в спільноті розробників.

Основні особливості Python: наявність сторонніх модулів дозволяє йому взаємодіяти з більшістю інших мов і платформ, великі бібліотеки підтримки, відкритий код і розвиток спільноти, швидкий час виконання, зручні вбудовані структури даних, тип даних згадувати не потрібно; він приймає тип даних на основі присвоєного значення, портативний та інтерактивний в операційних системах

6. C# - ще одна об'єктно-орієнтована мова, яка широко використовується для розробки мобільних додатків. Це мова програмування загального призначення, розроблена Microsoft, яка в основному використовується для платформи Windows Mobile. Ще одна особливість C#, яка робить його придатним для мобільних додатків, — кросплатформна розробка. Xamarin — це платформа, яка спрощує розробникам C# створення програм для Android та iOS. Крім розробки програм для мобільних пристроїв, C# широко використовується в бізнес-інструментах і інструментах продуктивності, підприємствах, іграх тощо. Ознайомившись із C#, ви зможете швидко розробляти програми для мобільних пристроїв, настільні програми, ігри та веб-програми. Таким чином, враховуючи функції C#, це буде один із найкращих варіантів для розробників мобільних додатків.

Ключові особливості: простий у використанні, широко використовується при розробці великих інструментів, це типово безпечна мова, масштабований, доступ до .NET framework

7. Kotlin - це статистично типізована кросплатформна мова програмування, яка була винайдена як альтернатива Java для розробки програм Android. У 2017 році він також був представлений як додаткова «офіційна» мова Java. Kotlin можна використовувати з JAVA для створення більш ефективних і високопродуктивних програм. Єдина істотна відмінність полягає в тому, що Kotlin видаляє функції Java, такі як винятки нульового показника. Це також усуває необхідність закінчувати кожен рядок крапкою з комою. Kotlin також можна використовувати як «точку входу» для розробки додатків для Android. Kotlin має дуже чистий синтаксис і простий код. Він взаємопов'язаний і універсальний, що створює компактний і чистіший код, ніж JAVA. Таким чином, це одна з найпопулярніших мов для розробки програм Android.

Причини обрати Python. Програмування на Python сприяє світовому ринку праці, тому що переваги Python очевидні [11, 12]. За даними ZDNet Python є однією з трьох найкращих мов програмування у світі, і вона

готується стати найпопулярнішою [13] . Фактично, згідно з індексом PYPL , Python є найпопулярнішою мовою програмування в усьому світі [14] (рис.3.1).

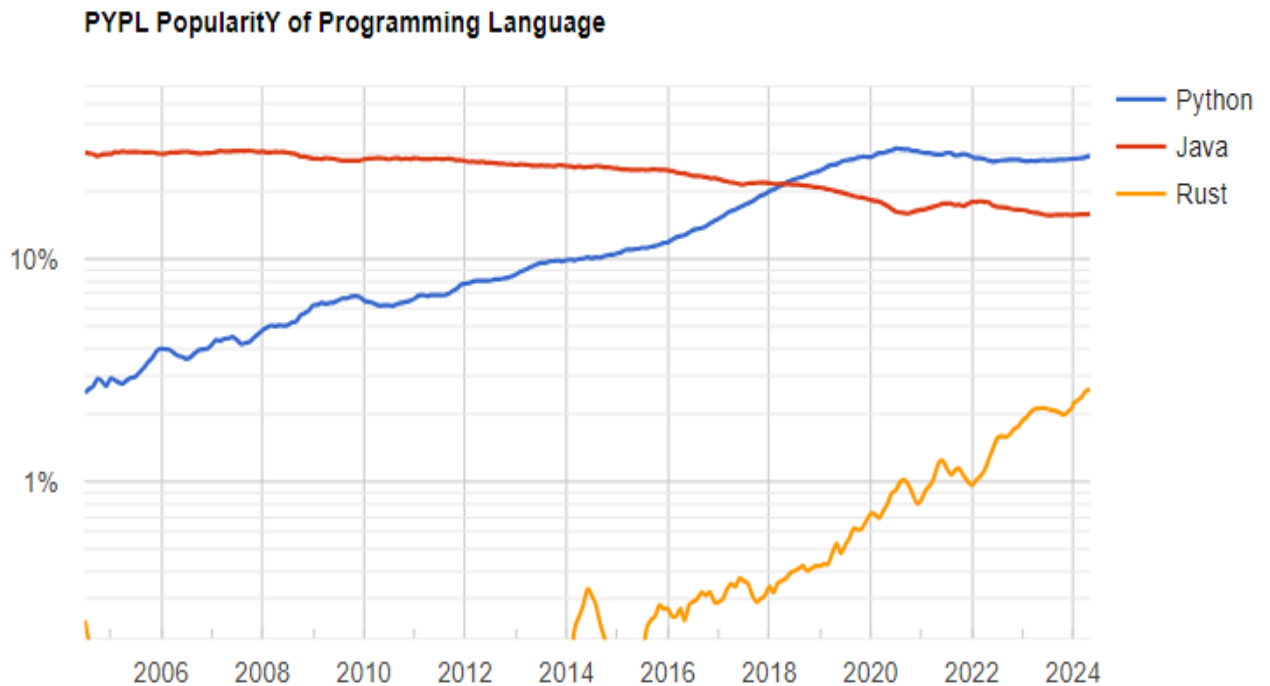


Рисунок 3.1 – Рейтинг популярних мов програмування

Крім того, Python використовується майже в усіх сферах ІТ-технологій, зокрема:

- Data Science;
- науково-математичні обчислення;
- веб-розробка;
- фінанси та торгівля;
- системна автоматизація та адміністрування;
- комп'ютерна графіка;
- розробка базової гри;
- тестування безпеки та проникнення;
- загальні сценарії та сценарії для окремих програм;
- картографування та географія (програмне забезпечення ГІС).

Таким чином, Python дає більше можливостей для роботи.

Переваги Python.

1. Легкість розуміння. Одна з головних переваг Python полягає в тому, що його легко вивчити просто використовувати. Синтаксис Python, на відміну від більшості комп'ютерних мов, читається як англійська, тому його не так напружено вивчати, як інші мови програмування. Він був названий на честь «Літаючого цирку» Монті Пайтона, тому ви знаєте, що хтось мав почуття гумору, розробляючи цей код, і таким чином зробив його простим у використанні. Він чудово справляється зі складністю, тож є можливість зосередитися на вивченні переваг програмування на Python, а не на дрібних деталях. Безкоштовний та має відкритий код.

2. Гнучкість. Python не тільки простий у вивченні, але й гнучкий. Існує понад 125 000 сторонніх бібліотек Python, які дозволяють використовувати Python для машинного навчання, веб-обробки та навіть біології. Крім того, його орієнтовані на дані бібліотеки, такі як pandas, NumPy і matplotlib, роблять його дуже здатним обробляти, маніпулювати та візуалізувати дані — тому його вважають за краще в аналізі даних. Він настільки зручний, що його часто називають «швейцарським армійським ножом» комп'ютерних мов.

3. Використовується в багатьох галузях промисловості. Завдяки перевагам Python і його гнучкості ви можете знайти його майже в будь-якій галузі. Нові програмісти на Python не обмежуються лише наукою про дані.

Де Python використовується в реальному житті? Python використовується в веб-розробці, аналізі даних, штучному інтелекті, наукових обчисленнях, мережевому програмуванні та автоматизації. Він використовується в різних

галузях, включаючи фінанси, охорону здоров'я, маркетинг і розваги, допомагаючи професіоналам оптимізувати робочі процеси, аналізувати великі набори даних і створювати потужні алгоритми та програми.

4. Великі бібліотеки. Python пропонує широкий спектр бібліотек, які можна використовувати в різних програмах. Бібліотеки — це колекції ресурсів, які допомагають нам оптимізувати розробку програм. Замість того, щоб писати кожен фрагмент коду з нуля, ми можемо використовувати бібліотеки, які містять багато попередньо написаних функцій і класів.

Деякі популярні бібліотеки Python включають:

- NumPy та SciPy - для наукових обчислень із широким набором функцій, алгоритмів та оптимізацій, включаючи лінійну алгебру та статистику;
- Keras, Seaborn, TensorFlow і SciKit-Learn - для машинного навчання, штучного інтелекту, обробки природної мови та нейронних мереж;
- Scrapy - для науки про дані дозволяє створити ефективний веб-сканер і збирач даних;
- Pandas - для аналізу даних, включаючи очищення даних і маніпулювання як реляційними, так і позначеними даними
- Matplotlib і Plotly - для візуалізації даних і побудови графіків.

5. Пише менше, робить більше. Python має дуже стислий синтаксис. Це помітно вірно навіть у порівнянні з іншими мовами програмування високого рівня, такими як Java. Простий синтаксис Python у поєднанні з великим набором бібліотек допомагає нам робити більше з меншою кількістю рядків коду. Це економить багато часу на розробку та є однією з причин популярності Python.

6. Портативність. Портативність є ще однією з сильних сторін Python . Портативність означає здатність програми працювати в різних операційних системах (ОС). Якщо ваша програма не містить системних викликів, ви

можете запускати свою програму Python у Windows, Mac OS і Linux, не змінюючи код програми. Все, що вам потрібно зробити, це використовувати інтерпретатор Python, який підходить для вибраної вами платформи. Портативність Python значною мірою пояснюється використанням інтерпретатора замість компілятора. Як інтерпретатори, так і компілятори перетворюють вихідний код у машинний код. Однак вони роблять це в різний час і різними способами. Інтерпретатори перетворюють вихідний код під час виконання програми, а компілятори перетворюють його перед виконанням програми. Зокрема, вихідний код Python перетворюється на проміжну форму під назвою байт-код, який можна виконати на будь-якій платформі, яка має інтерпретатор Python. Навпаки, компілятор перетворював би вихідний код у непереносимий машинний код, який міг бути виконаний лише на певній платформі.

7. Покращена структура. Python працює з фреймворками Django, Pyramid, Pylons і Flask. З Django - найбільшою мірою. Цей фреймворк простий, безпечний і швидкий, відрізняється високою надійністю і стабільністю в роботі. Тож не дивно, що розробники зазвичай віддають перевагу саме йому.

Отже, основні переваги Python такі: більш продуманий дизайн, порівняльна простота для вивчення, покращена структура, більш читабельний стек, нескладний синтаксис, прості і зрозумілі інструменти налагодження, універсальність, велика підтримка і активна спільнота

3.2 Створення програмної реалізації

Проксі-сервер виконує роль посередника між клієнтом і цільовим сервером. Проксі-сервер діє «від особи» клієнта і, в залежності від поставленої задачі, може виконувати різні перетворення даних.

Завдання роботи полягає у розробці програмного модуля управління проксі-фермою, використовуючи тільки стандартні бібліотеки Python.

Перед початком розробки були сформульовані наступні критерії функціональності додатка:

- кожне нове з'єднання клієнта з проксі-сервером, повинно ініціювати нове з'єднання з цільовим сервером;
- кожен пакет даних, що знаходиться на проксі-сервері від клієнта, повинен пересилатися на цільовий сервер;
- кожен пакет даних, що знаходиться на проксі-сервері від цільового сервера, повинен пересилатися відповідному клієнту;
- підтримка роботи з кількома клієнтами;
- висока швидкість передачі;
- малий об'єм використаних ресурсів.

Розглянемо основні елементи розробленого програмного модуля управління проксі-фермою (додаток В).

Клас Forward - встановлює з'єднання між проксі-сервером і цільовим сервером (рис.3.2).

```
class Forward:

    def __init__(self):

        self.forward = socket.socket(socket.AF_INET,
socket.SOCK_STREAM)

    def start(self, host, port):

        try:

            self.forward.connect((host, port))

            return self.forward

        except Exception, e:

            print e

            return False
```

Рисунок 3.2 - Клас Forward

Клас TheServer - основний клас додатку (рис.3.3).

```
class TheServer:

    input_list = []

    channel = {}

    def __init__(self, host, port):

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

        self.server.setsockopt(socket.SOL_SOCKET,
socket.SO_REUSEADDR, 1)

        self.server.bind((host, port))

        self.server.listen(200)
```

Рисунок 3.3 - Клас TheServer

Метод TheServer.main_loop() (рис.3.4). Список input_list містить усі доступні сокети, керування якими здійснюється за допомогою select.select(). Першим у список додається серверний пакет самого проксі-сервера. Якщо нове підключення до цього сокету буде ініціювати виклик методу on_accept().

Якщо поточний сокет зі списку inputready, повернутого select.select(), не передбачає нового з'єднання, то маємо справу з поточними даними (можливо з цільного сервера, можливо, від клієнта). Якщо розмір даних дорівнює нулю, то це запит на закриття з'єднань, у протилежному випадку пакет необхідно переслати у відповідне місце призначення.

```

def main_loop(self):

    self.input_list.append(self.server)

    while 1:

        time.sleep(delay)

        ss = select.select

        inputready, outputready, exceptready =
ss(self.input_list, [], [])

        for self.s in inputready:

            if self.s == self.server:

                self.on_accept()

                break

            self.data = self.s.recv(buffer_size)

            if len(self.data) == 0:

                self.on_close()

                break

            else:

                self.on_recv()

```

Рисунок 3.4 - Метод TheServer.main_loop()

Метод TheServer.on_accept() - цей метод встановлює нове з'єднання з цільовим сервером, а також приймає підключення поточного клієнта (рис.3.5).

```

def on_accept(self):
    forward = Forward().start(forward_to[0],
forward_to[1])

    clientsock, clientaddr = self.server.accept()

    if forward:

        print clientaddr, "has connected"

        self.input_list.append(clientsock)

        self.input_list.append(forward)

        self.channel[clientsock] = forward

        self.channel[forward] = clientsock

    else:

        print "Can't establish connection with
remote server.",

        print "Closing connection with client
side", clientaddr

        clientsock.close()

```

Рисунок 3.5 - Метод TheServer.on_accept()

Обидва сокети додаються в список `input_list` для подальшої обробки в методі `main_loop()`. Обидва сокети також зберігаються в словнику каналу для розміщення кінцевих точок призначення (клієнт $\Leftrightarrow$ цільового сервера).

Обидва сокети додаються в список `input_list` для подальшої обробки в методі `main_loop()`. Обидва сокети також зберігаються в словнику каналу для розміщення кінцевих точок призначення (клієнт $\Leftrightarrow$ цільового сервера).

Метод `TheServer.on_recv()` - цей метод виконує обробку даних (за необхідності) і пересилає їх за місцем призначення (клієнт $\leq$ проксі-сервер $\Rightarrow$ цільовий сервер) (рис.3.6).

```
def on_recv(self):

    data = self.data

    # here we can parse and/or modify the data
    before send forward

    print data

    self.channel[self.s].send(data)

if __name__ == '__main__':

    server = TheServer('', 9090)

    try:

        server.main_loop()

    except KeyboardInterrupt:

        print "Ctrl C - Stopping server"

        sys.exit(1)
```

Рисунок 3.6 - Метод `TheServer.on_recv()`

Метод `TheServer.on_close()` - цей метод відкриває з'єднання між проксі-сервером і цільовим сервером, а також між проксі-сервером і клієнтом. Відповідні об'єкти видаляються (рис.3.7).

```

def on_close(self):

    print self.s.getpeername(), "has disconnected"

    #remove objects from input_list

    self.input_list.remove(self.s)

    self.input_list.remove(self.channel[self.s])

    out = self.channel[self.s]

    # close the connection with client

    self.channel[out].close() # equivalent to do
self.s.close()

    # close the connection with remote server

    self.channel[self.s].close()

    # delete both objects from channel dict

    del self.channel[out]

    del self.channel[self.s]

```

Рисунок 3.7 - Метод TheServer.on_close()

3.3 Тестування програмної реалізації

Для тестування проксі-ферми було використано такі компоненти.

1. Сервер ПК- Tower (1 шт.), підключаємо до нього 4 модеми.

Технічні характеристики Tower:

- Intel, Core i5-12500,
- Thermaltake, Core V1 Snow Edition / Black edition,
- материнська плата Gigabyte, H610I,
- оперативна пам'ять ddr4 і найменший SSD.

2. Модеми 4G/LTE - Huawei e3372h 153 (4 шт.):
3. USB-концентратор для підключення модемів (1 шт.).
4. USB-подовжувач для зручності (1 шт.).
5. Роутер для підключення ПК до Інтернету – TP-LINK TL-WR840N.
(1шт.)



Рисунок 3.8 – Компоненти проксі-ферми

Для тестування програмної реалізації встановили MyProxy.ru за допомогою менеджера пакетів Python, pip. Виконали таку команду:

```
pip install myproxy.py
```

Основне налаштування та конфігурація. Після встановлення myproxy.ru можна почати налаштовувати його для базового використання. Щоб запустити проксі-сервер із налаштуваннями за замовчуванням запустили:

```
myproxy
```

Ця команда запускає проксі-сервер на локальній машині. За замовчуванням сервер працює на 127.0.0.1(localhost) і порт 8899. Можливо змінити ці параметри за замовчуванням, вказавши IP-адресу та порт як аргументи командного рядка. Наприклад:

```
myproxy --hostname 0.0.0.0 --port 9000
```

Тепер, коли запущено MyProху.ру, налаштували базовий проксі-сервер.

Простий приклад.

Налаштувати свій проксі: налаштувати запити HTTP або HTTPS для маршрутизації через проксі-сервер. Наприклад, якщо використовувати бібліотеку запитів Python, то можна зробити це за допомогою:

```
import requests

proxies = {
    'http': 'http://127.0.0.1:8899',
    'https': 'http://127.0.0.1:8899',
}
response = requests.get('http://example.com',
proxies=proxies)
print(response.text)
```

Цей сценарій направляє запит через проксі-сервер, що працює на localhostпорту 8899.

Після налаштування для використання запустили свій сценарій. Успішно отримали дані через проксі. Можна перевірити це, перевіривши журнали myproху.ру, де побачили вхідні запити.

Було протестовано проксі-ферму у реальних сценаріях засобами Proху Checker [15]. Вона має високу якість, що значно краще, ніж у багатьох інших (рис.3.9).

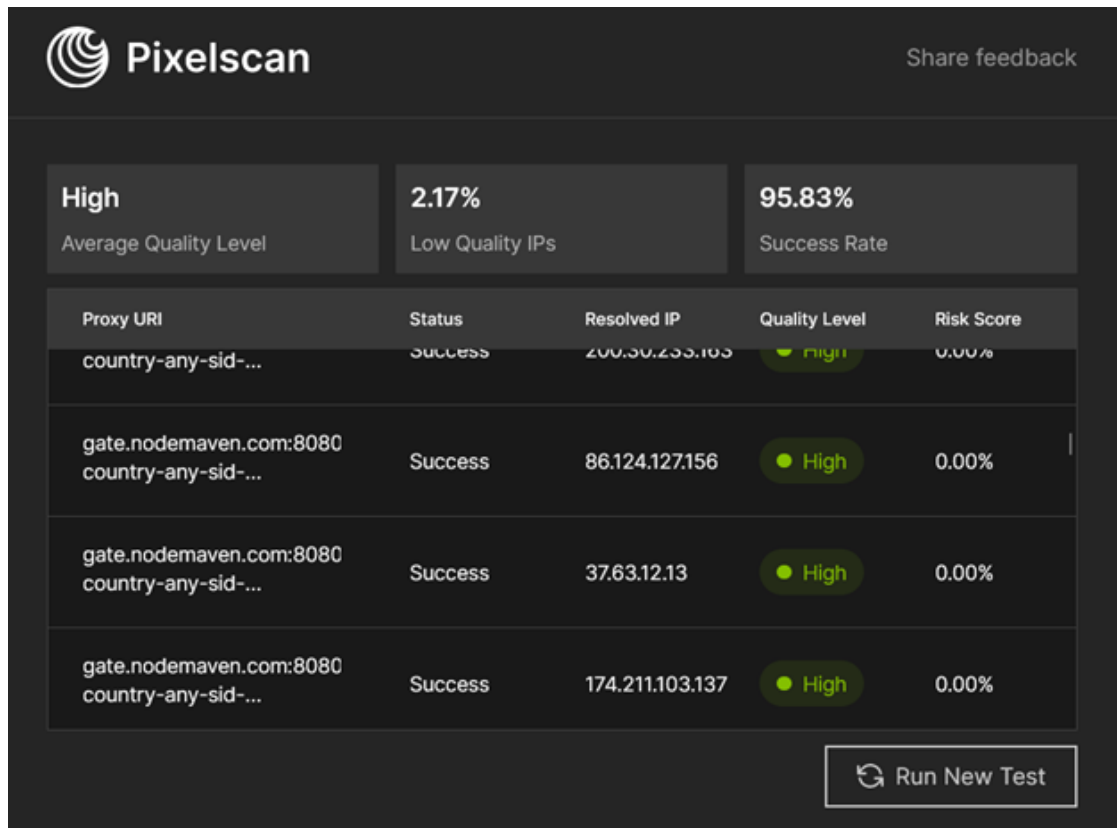


Рисунок 3.9 - Результати тестування

3.4 Висновок до розділу 3

Отже, за результатами розробки у якості програмного засобу реалізації проксі-ферми обрано об'єктно-орієнтовану мову програмування Python. У якості середовища розробки обрано Visual Studio. Створено програмний модуль реалізації проксі-ферми. За результатами розробки створено програмний модуль управління проксі-фермою, проведено тестування обраної апаратної реалізації, яке підтвердило працездатність та оптимальність візначених рішень.

ВИСНОВКИ

У результаті виконання бакалаврської дипломної роботи було досягнуто наступних результатів.

Достатньо актуальною задачею, є створення власної мобільної проксі-ферми, розв'язання якої надає ряд переваг, а саме: контроль трафіку; заощадження коштів; адаптування для конкретного GEO; можливість заробітку на їх оренду. У якості завдання обрано створення програмного забезпечення для розгортання мобільної проксі-ферми.

Розробка програмного модуля проксі-ферми 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 засоби розробки програмного модуля проксі-ферми. Розроблені UML діаграми послідовності роботи проксі-ферми. Деталізовані класи програмного забезпечення програмного модуля проксі-ферми.

Отже, за результатами розробки у якості програмного засобу реалізації проксі-ферми обрано об'єктно-орієнтовану мову програмування Python. У якості середовища розробки обрано Visual Studio. Створено програмний модуль реалізації проксі-ферми. За результатами розробки створено програмний модуль управління проксі-фермою, проведено тестування обраної апаратної реалізації, яке підтвердило працездатність та оптимальність визначених рішень.

Результати розробки програмного модуля управління проксі-фермою дозволяють підвищити надійності, зменшити витрати та спростити керування

проксі-фермою за рахунок уникнення використання зовнішнього програмного забезпечення та простої організації особистого програмного модуля. Результати розробки можливо порадити для використання усім користувачам-початківцям проксі-ферм.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Farming & Multiaccounting Techniques: Enabling the Proxies.
<https://en.trafficcardinal.com/post/farming-multiaccounting-techniques-enabling-the-proxies>
2. Pixelscan. <https://pixelscan.net>
3. Whoer. <https://whoer.net>
4. Fake.Vision. <http://fake.vision>
5. CheckerProxy.net. <https://checkerproxy.net>
6. Getipintel.net. <https://getipintel.net>
7. Час створити власний проксі. <https://en.trafficcardinal.com/post/time-to-make-your-own-proxy>
8. Стати афілійованим маркетологом.
<https://en.trafficcardinal.com/post/becoming-an-affiliate-marketer>
9. Початок роботи з мобільними проксі.
<https://en.trafficcardinal.com/post/getting-started-with-mobile-proxies>
10. Як працюють мобільні проксі. <https://proxy24.pro/indexua.html>
11. Why Choose Python? Discover Its Core Advantages!
<https://www.simplilearn.com/why-learn-python-a-guide-to-unlock-your-python-career-article>
12. Why learn Python? 5 advantages and disadvantages
<https://www.educative.io/blog/why-learn-python>
13. Programming languages: Python could soon overtake C and Java as most popular. <https://www.zdnet.com/article/programming-languages-python-could-soon-overtake-c-and-java-as-most-popular/>
14. The PYPL Popularity of Programming Language Index is created by analyzing how often language tutorials are searched on Google.
<https://pypl.github.io/PYPL.html>
15. PROXY CHECKER. <https://proxy6.net/en/checker>

ДОДАТКИ

ДОДАТОК А

ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬНазва роботи: Розробка програмного модуля управління проксі-фермоюТип роботи: бакалаврська дипломна робота
(БДР, МКР)Підрозділ кафедра комп'ютерних наук, ФІТА
(кафедра, факультет)

Показники звіту подібності Unichesk

Оригінальність 87,7% Схожість 12,3%

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.

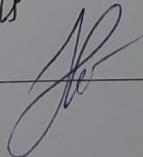
Ознайомлені з повним звітом подібності, який був згенерований системою Unichesk щодо роботи.

Автор роботи



Мацько О.С.

Керівник роботи



Колодний В.В.

ДОДАТОК Б

ІНСТРУКЦІЯ КОРИСТУВАЧА

Встановлення MyProxy.py.

Щоб почати користуватися myproxy.py, спочатку потрібно встановити його. Це можна легко зробити за допомогою менеджера пакетів Python, pip. Відкрийте термінал або командний рядок і виконайте таку команду:

```
pip install myproxy.py
```

Основне налаштування та [конфігурація](#)

Після встановлення myproxy.py можна почати налаштовувати його для базового використання. Щоб запустити проксі-сервер із налаштуваннями за замовчуванням, просто запустіть:

```
myproxy
```

Ця команда запускає проксі-сервер на локальній машині. За замовчуванням сервер працює на 127.0.0.1(localhost) і порт 8899. Можливо змінити ці параметри за замовчуванням, вказавши IP-адресу та порт як аргументи командного рядка. Наприклад:

```
myproxy --hostname 0.0.0.0 --port 9000
```

Тепер, коли запущено MyProxy.py, можете налаштувати базовий проксі-сервер.

Простий приклад.

Налаштуйте свій проксі: у вашому сценарії налаштуйте запити HTTP або HTTPS для маршрутизації через проксі-сервер. Наприклад, якщо використовуєте бібліотеку запитів Python, ви можете зробити це за допомогою:

```
import requests

proxies = {
    'http': 'http://127.0.0.1:8899',
    'https': 'http://127.0.0.1:8899',
}
response = requests.get('http://example.com',
proxies=proxies)
print(response.text)
```

Цей сценарій направляє запит через проксі-сервер, що працює на localhostпорту 8899.

Тестування та перевірка: після налаштування для використання запустіть свій сценарій. Якщо все налаштовано правильно, маєте успішно отримати дані через проксі. Можна перевірити це, перевібивши журнали тургоху.ру, де повинні побачити вхідні запити.

ДОДАТОК В

ЛІСТИНГ ПРОГРАМИ

```
Import socket

import select

import time

import sys

buffer_size = 4096

delay = 0.0001

forward_to = ('smtp.zaz.ufsk.br', 25)


class Forward:

    def __init__(self):

        self.forward = socket.socket(socket.AF_INET,
socket.SOCK_STREAM)

    def start(self, host, port):

        try:

            self.forward.connect((host, port))

            return self.forward

        except Exception, e:

            print e

            return False
```

```
class TheServer:

    input_list = []

    channel = {}

    def __init__(self, host, port):

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

        self.server.setsockopt(socket.SOL_SOCKET,
socket.SO_REUSEADDR, 1)

        self.server.bind((host, port))

        self.server.listen(200)

def main_loop(self):

    self.input_list.append(self.server)

    while 1:

        time.sleep(delay)

        ss = select.select

        inputready, outputready, exceptready =
ss(self.input_list, [], [])

        for self.s in inputready:

            if self.s == self.server:

                self.on_accept()
```

```

        break

        self.data = self.s.recv(buffer_size)

        if len(self.data) == 0:

            self.on_close()

            break

        else:

            self.on_recv()


def on_accept(self):

    forward = Forward().start(forward_to[0],
forward_to[1])

    clientsock, clientaddr = self.server.accept()

    if forward:

        print clientaddr, "has connected"

        self.input_list.append(clientsock)

        self.input_list.append(forward)

        self.channel[clientsock] = forward

        self.channel[forward] = clientsock

    else:

        print "Can't establish connection with
remote server.",

```

```

        print "Closing connection with client
side", clientaddr

```

```

        clientsock.close()

```

```

def on_close(self):

```

```

    print self.s.getpeername(), "has disconnected"

```

```

    #remove objects from input_list

```

```

    self.input_list.remove(self.s)

```

```

    self.input_list.remove(self.channel[self.s])

```

```

    out = self.channel[self.s]

```

```

    # close the connection with client

```

```

    self.channel[out].close() # equivalent to do
self.s.close()

```

```

    # close the connection with remote server

```

```

    self.channel[self.s].close()

```

```

    # delete both objects from channel dict

```

```

    del self.channel[out]

```

```

    del self.channel[self.s]

```

```

def on_recv(self):

```

```

    data = self.data

```

```
        # here we can parse and/or modify the data
before send forward

        print data

        self.channel[self.s].send(data)

if __name__ == '__main__':

    server = TheServer('', 9090)

    try:

        server.main_loop()

    except KeyboardInterrupt:

        print "Ctrl C - Stopping server"

        sys.exit(1)
```

ДОДАТОК Г

ГРАФІЧНА ЧАСТИНА

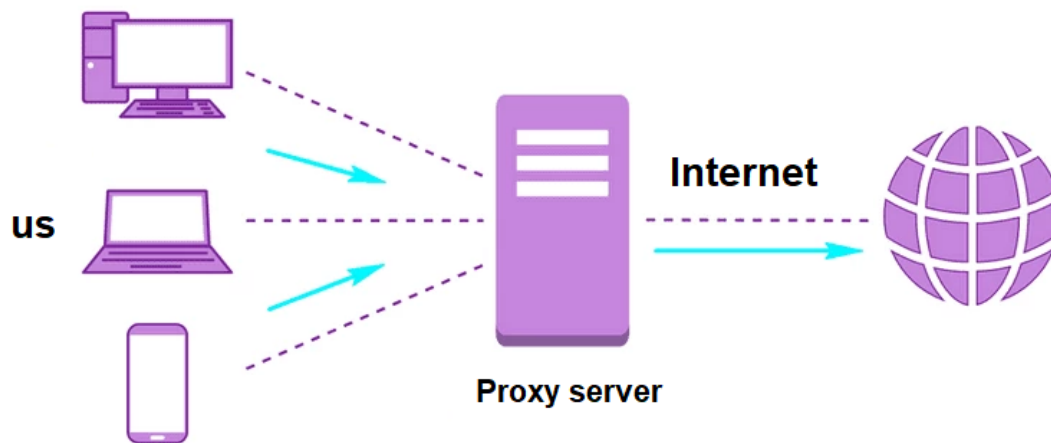


Рисунок Г.1 - Основні елементи проксі

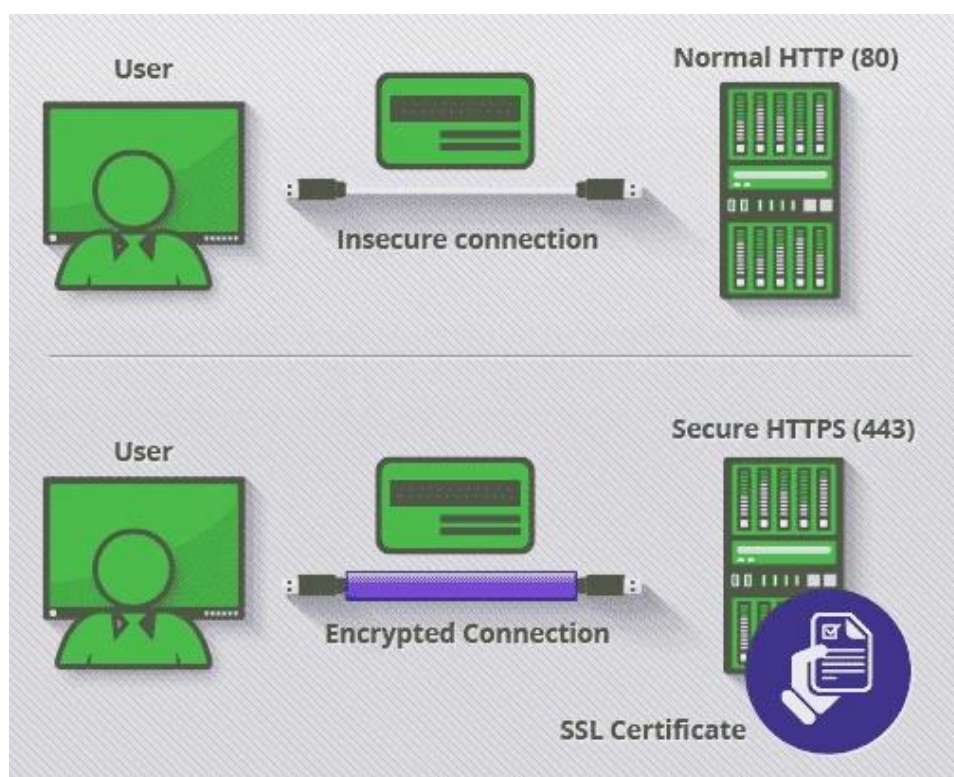


Рисунок Г.2 - Сертифікат SSL

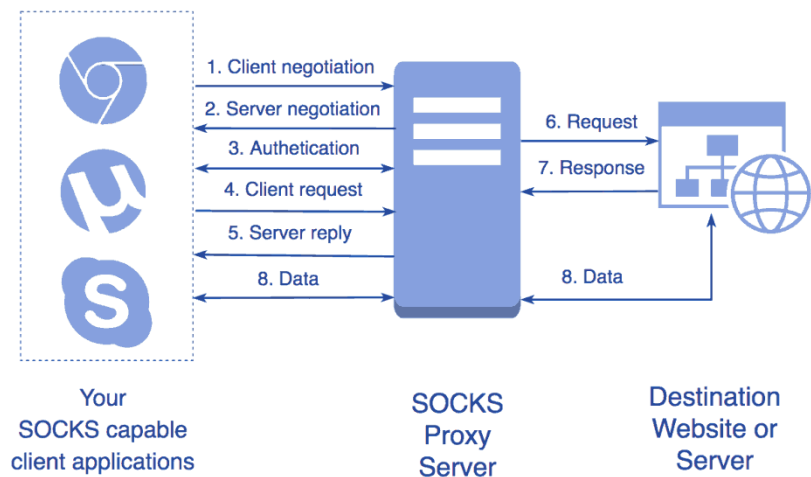


Рисунок Г.3 - Проксі SOCKS

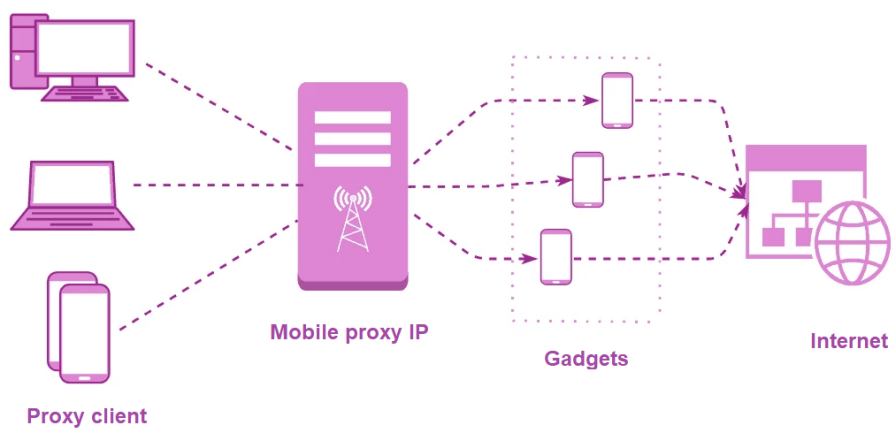


Рисунок Г.4 – Структура мобільної проксі-ферми



Рисунок Г.5 - Структура віртуальної машини

Description of the 4G proxies setup

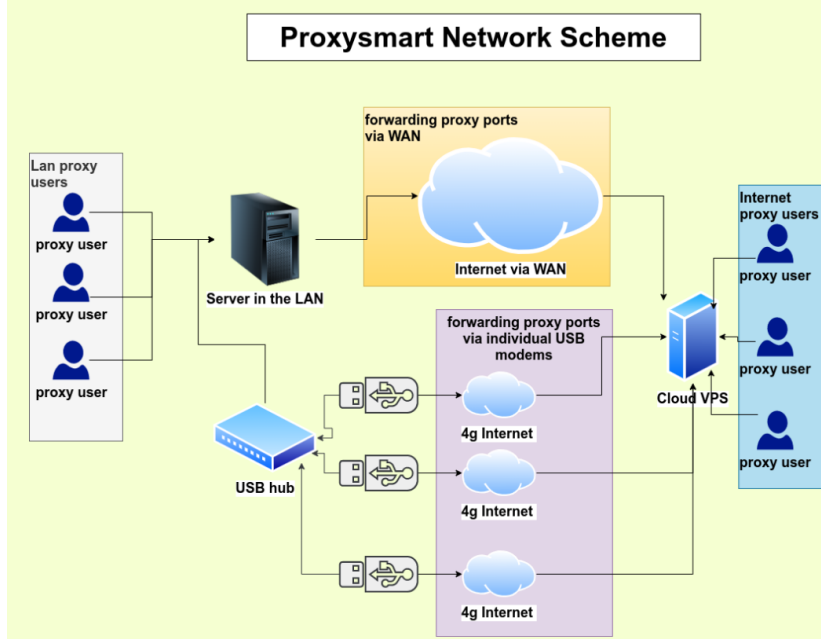


Рисунок Г.6 - Варіант організації проксі-ферми

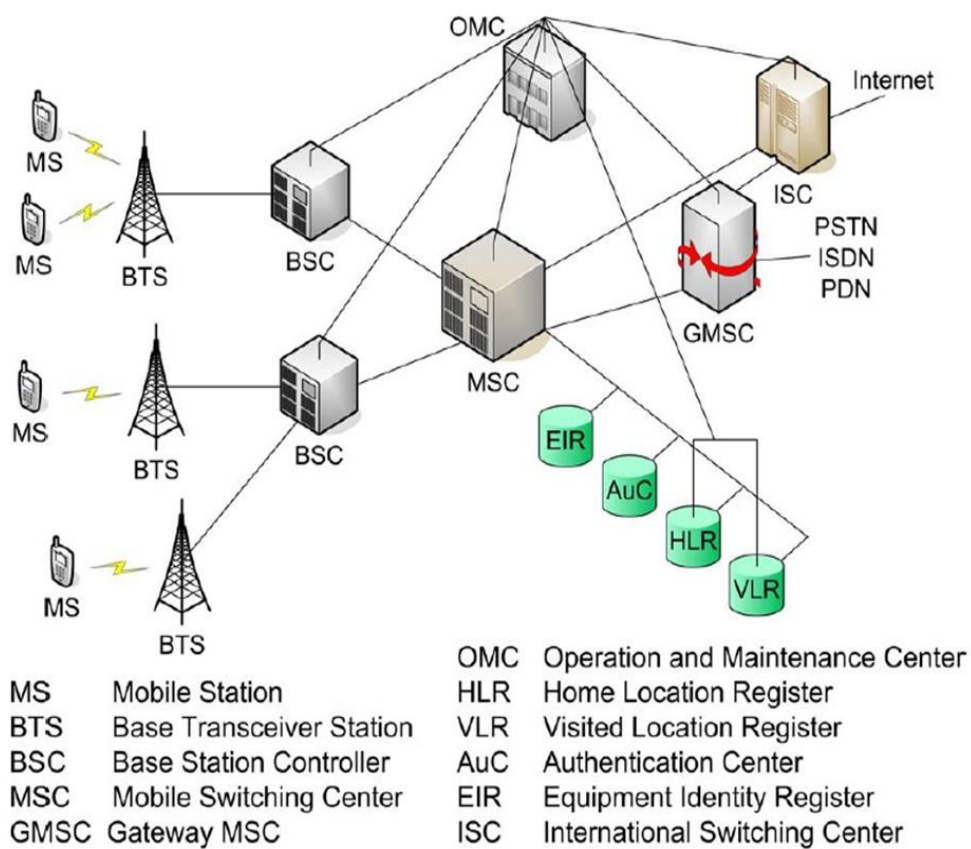


Рисунок Г.7 - Архітектура мережі GSM 4G

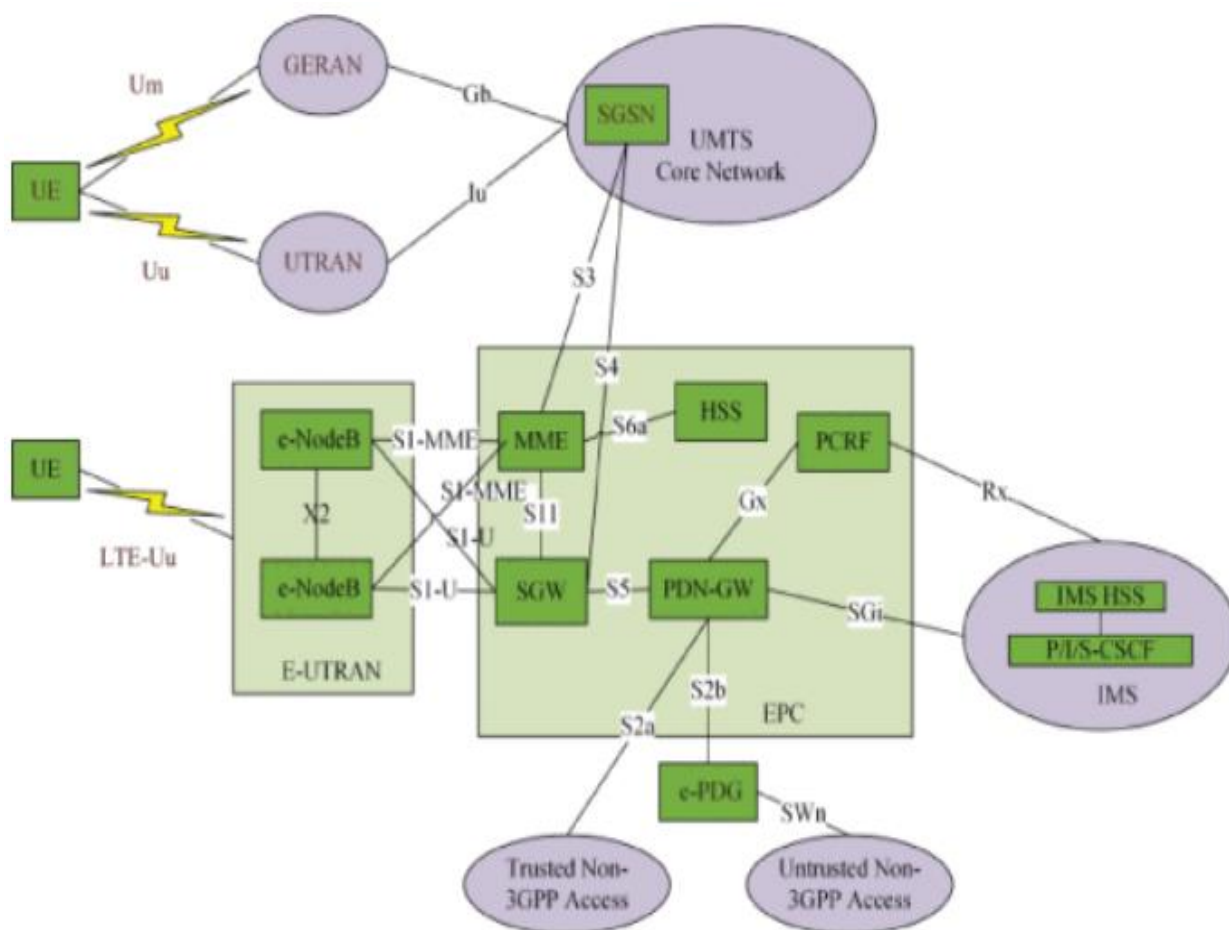


Рисунок Г.8 - Мережева архітектура LTE-Advanced

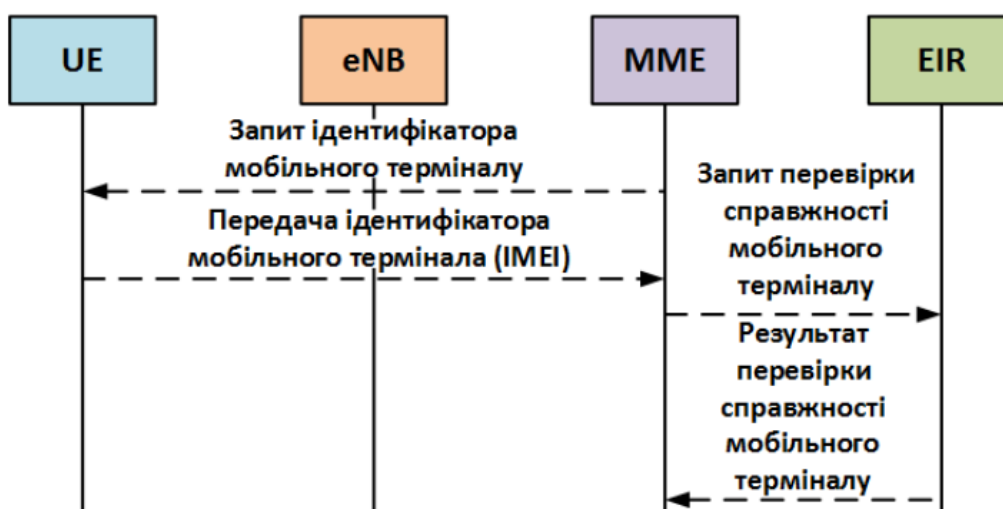


Рисунок Г.9 – Процедура автентифікації мобільного терміналу

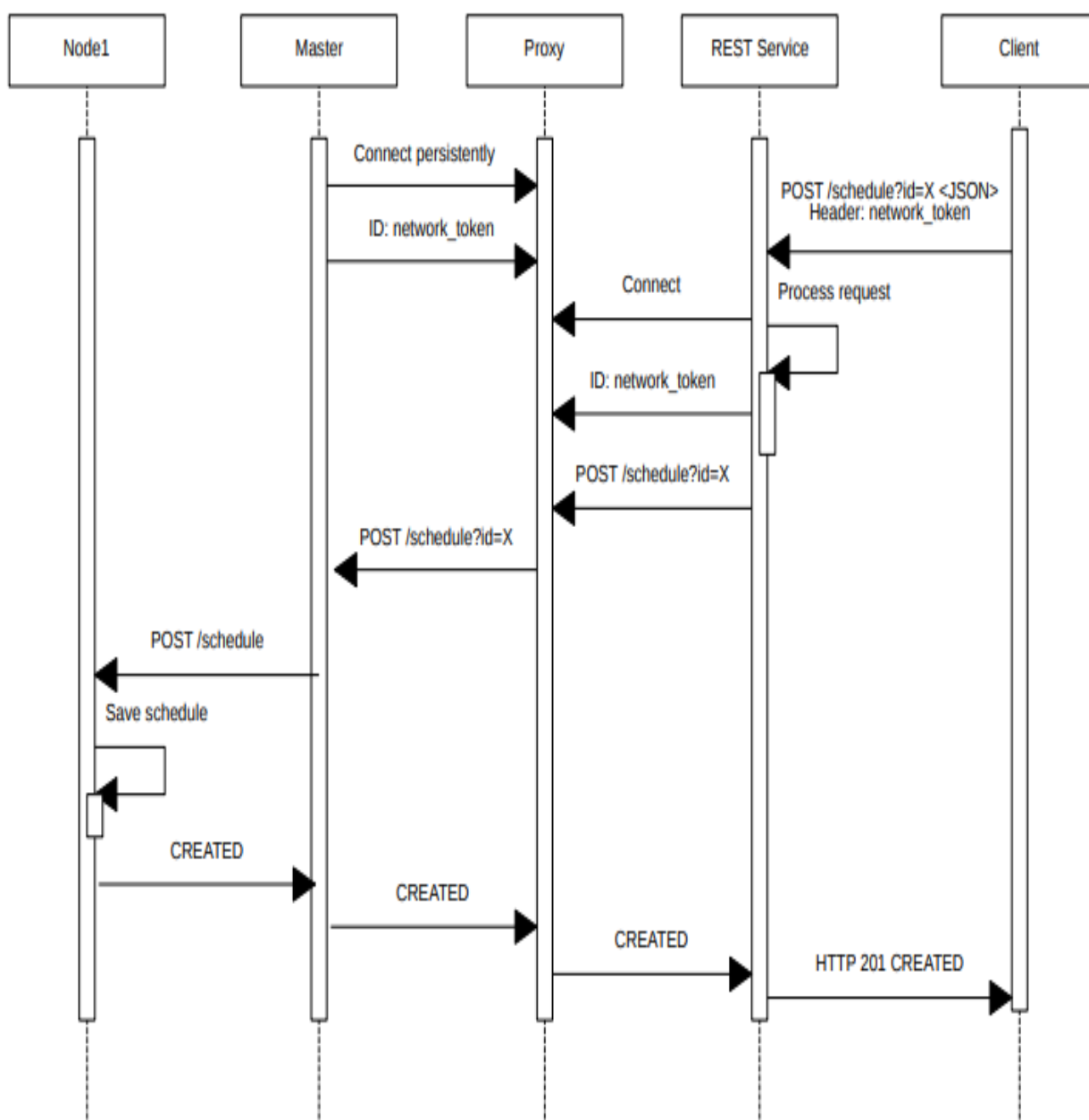


Рисунок Г.10 - UML діаграма послідовності додавання інтервалу операції проксі-сервером

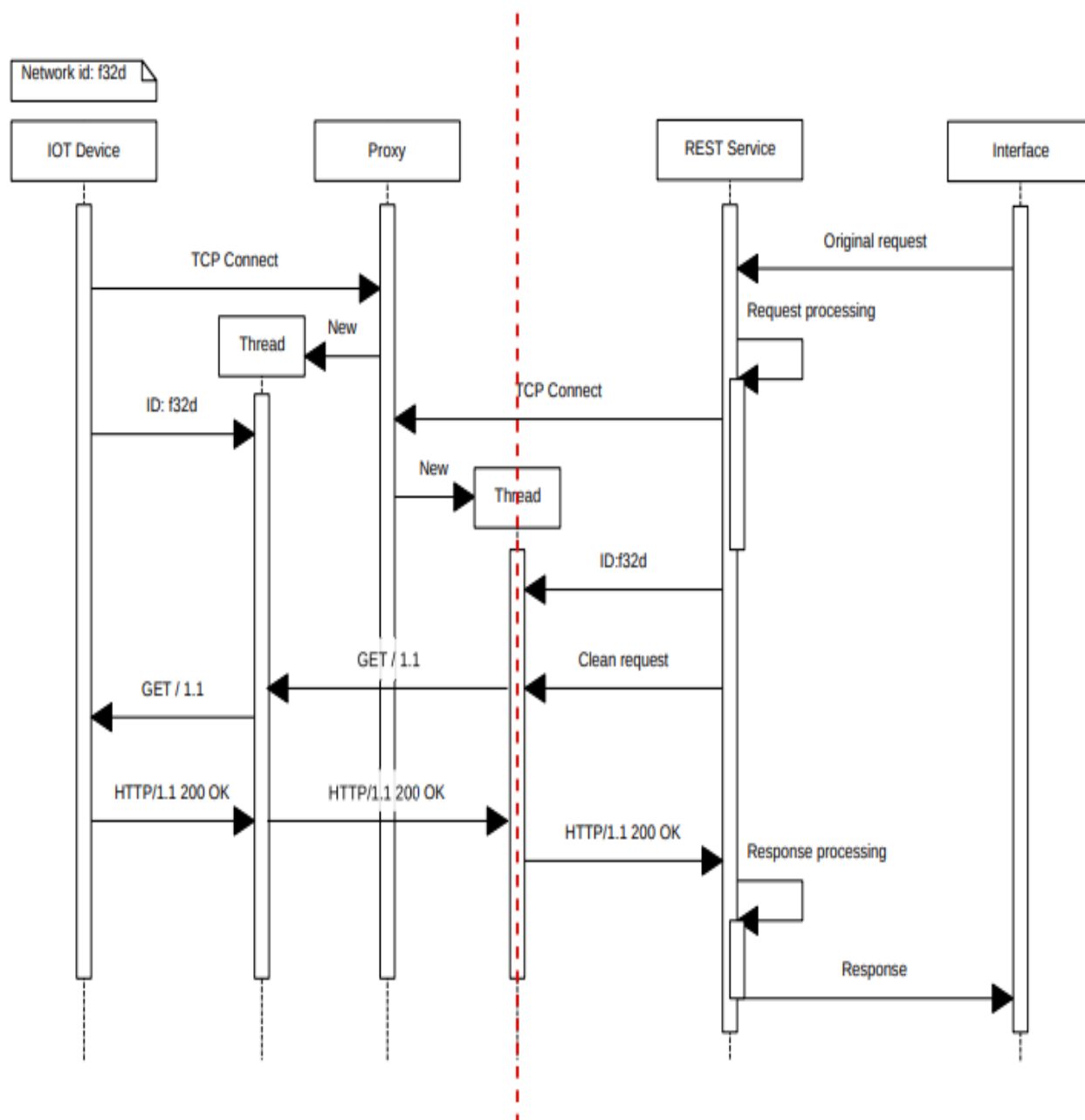


Рисунок Г.11 - UML діаграма послідовності прикладу роботи проксі-сервера

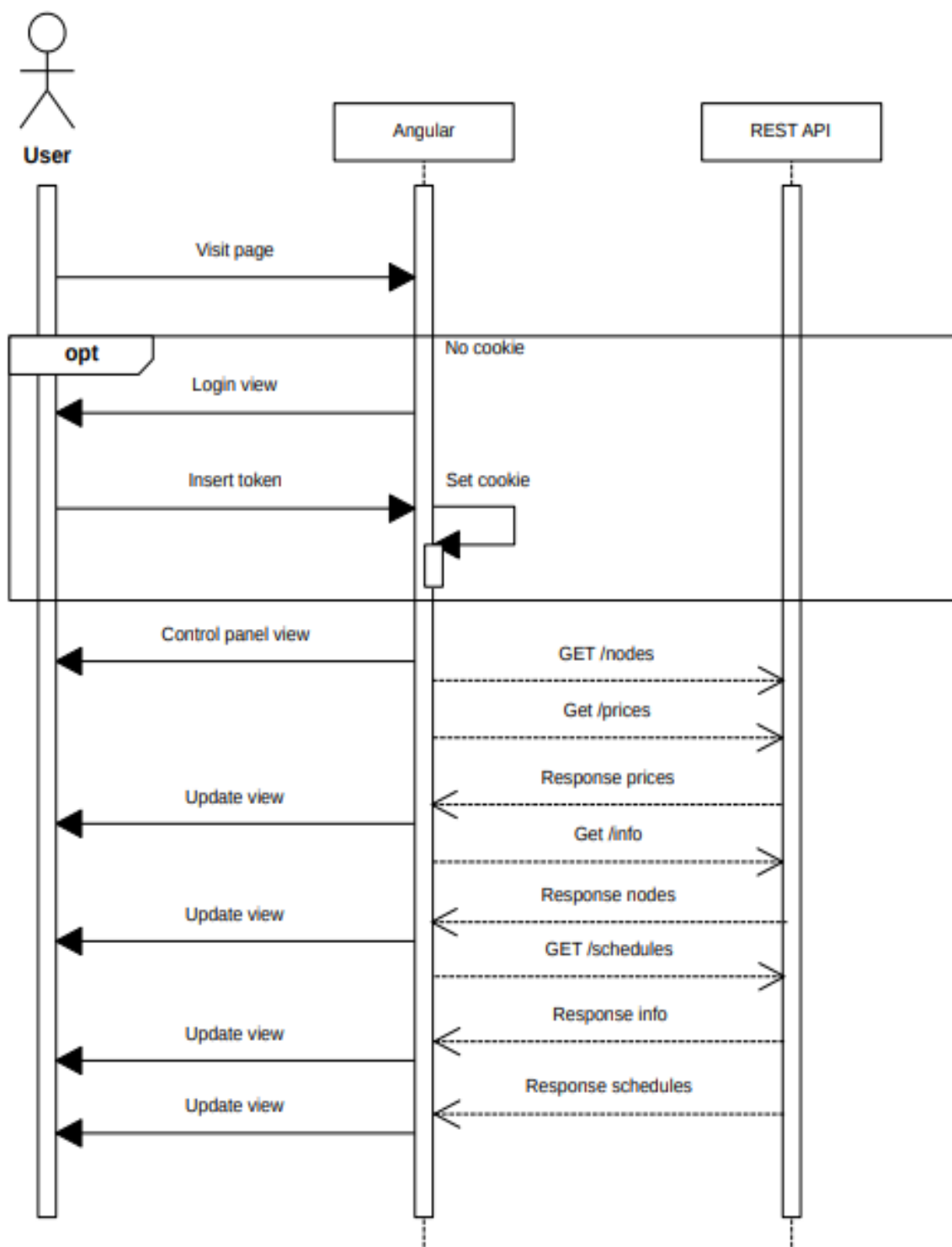


Рисунок Г.12 - Діаграма послідовності обробки веб-додатків