**Структура розробленого програмного продукту**

Програмний продукт написано мовою TypeScript, об’єкти топології мережі описані мовою Protobuf для полегшення роботи зі зберіганням та відтворенням мережі з файлу.

Код програми розподілено по різним файлам в залежності від функціональності, яку вони реалізують. Концептуальна модель наведена на рис 2.1.

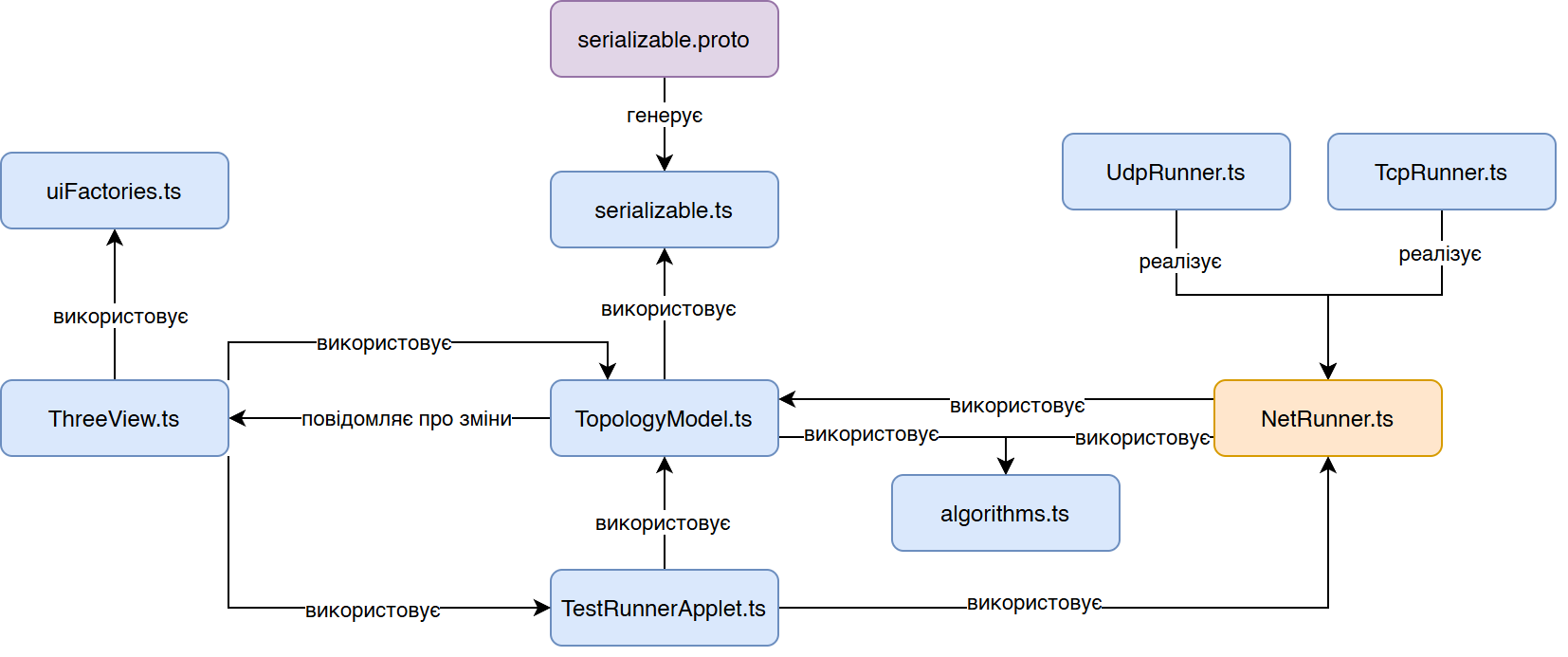


Рис. 2.1 – Концептуальна модель програмного продукту

На моделі видно, що керуючим модулем є клас ThreeView, який відповідає за забезпечення взаємодії користувача з продуктом (виведення графу на екран, обробка команд тощо), інші модулі підключаються до нього прямо або опосередковано:

1. **uiFactories.ts** – містить визначення елементів графічного інтерфейсу (діалогові вікна та меню).
2. **serializable.proto** – опис сутностей топології мережі: об’єкти підключення, робочої станції, комунікаційного вузла.
3. **serializable.ts** – згенерований з опису сутностей файл з функціями створення та кодування об’єктів.
4. **algorithms.ts** – містить реалізацію алгоритму Беллмана-Форда.
5. **TopologyModel.ts** – модель топології мережі, містить функції для створення об’єктів, їх редагування та підписки на їхні зміни.
6. **NetRunner.ts** – містить опис інтерфейсу підпрограми моделювання пересилання повідомлень.
7. **TcpRunner.ts, UdpRunner.ts** – реалізації інтерфейсу NetRunner для протоколів TCP та UDP.
8. **TestRunnerApplet.ts** – підпрограма, що дозволяє виконувати тести та експортувати результати у форматі таблиць Excel.

**Опис інтерфейсу розробленого програмного продукту**

Графічний інтерфейс складається з двох частин: меню та області побудови мережі (рис 2.2).

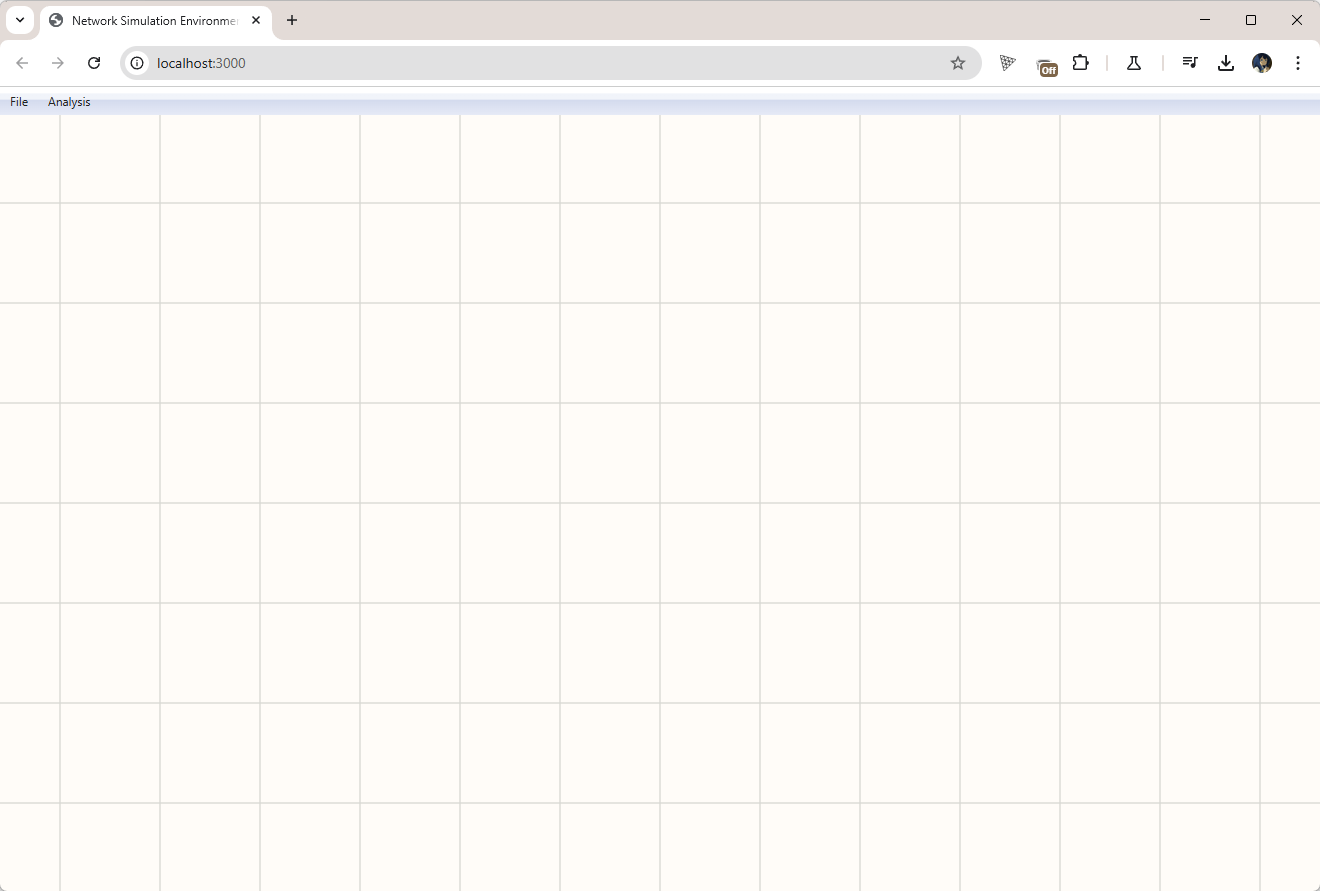


Рис 2.2 – Головний екран програмного продукту

В меню доступні функції збереження та завантаження мережі з файлу на диску (рис. 2.3), а також функції аналізу (рис. 2.4).

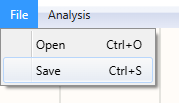


Рис. 2.3 – Меню роботи з файлами

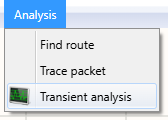


Рис. 2.4 – Меню утіліт аналізу

Створювати елементи топології мережі користувач може через меню керування топологією мережі (рис. 2.5), що відкривається по натисканню правої кнопки миші в пустому місці.

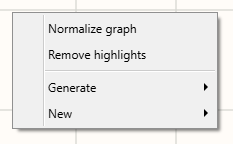


Рис. 2.5 – Меню керування топологією мережі

Елементи можна створити автоматично через генератор регіональних мереж (рис. 2.7) або самостійно, обравши з переліку (рис. 2.6) потрібний елемент та налаштувавши його (рис. 2.8 та 2.9).

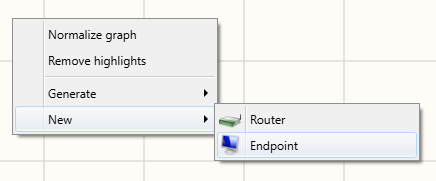


Рис. 2.6 – Перелік доступних для створення елементів

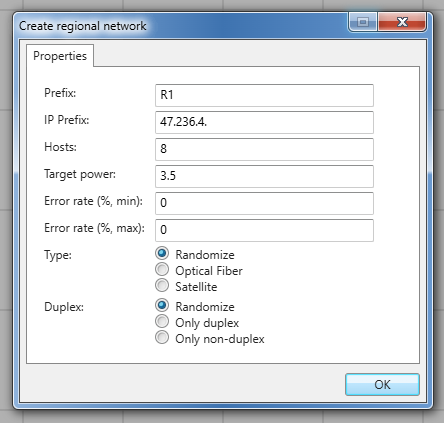


Рис. 2.7 – Вікно генератора регіональних мереж

Генератор локальних мереж створює задану користувачем кількість комунікаційних вузлів та призначає їм назви за префіксом (Prefix). Кількість вузлів задається користувачем в полі Hosts, а кількість каналів між ними визначається за автоматично, користувачу лише треба вказати порібну ступінь (Target power), ваги каналів будуть визначені випадково із допустимого набору. Ймовірність помилки також буде призначена випадково, але користувач має можливість вказати проміжок дозволених значень у полях Error rate. За умовчанням тип каналу та його дуплексність визначається випадково, але користувач може обмежити це і вказати налаштування каналів вручну в полях Type та Duplex.

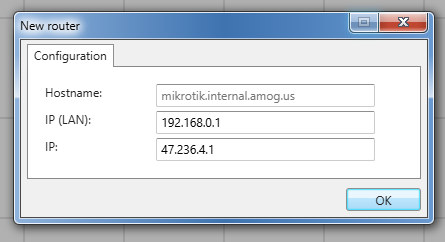


Рис 2.8 – Вікно створення комунікаційного вузла (роутера)

При створені комунікаційного вузла в ручному режимі, потрібно вказати його ім’я та IP-адреси. Зовнішні IP-адреси та ім’я при цьому мають бути унікальними для всієї мережі, а внутрішня IP-адреса унікальною в межах локальної мережі.

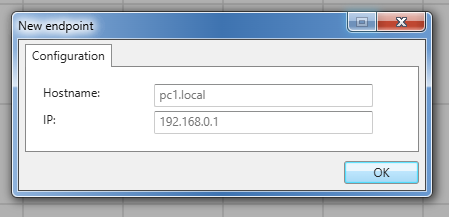


Рис. 2.9 – Вікно створення робочої станції

Робоча станція створюється так само як і комунікаційний вузол, але має лише внутрішню IP-адресу.

При натисканні правою кнопкою миші на елемет відкривається меню керування елементом (рис. 2.10).

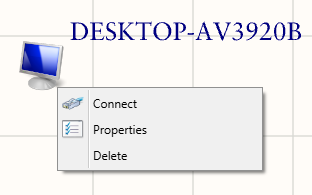


Рис. 2.10 – Меню керування елементом

З меню можна видалити елемент або відкрити його налаштування, що мають такий самий вигляд як і при створенні. Також з меню можна перейти у режим створення підключення.

У режимі створення підключення потрібно натиснути на елемент, до якого треба підєднатись. Тоді відкриється вікно створення каналу (рис. 2.11).

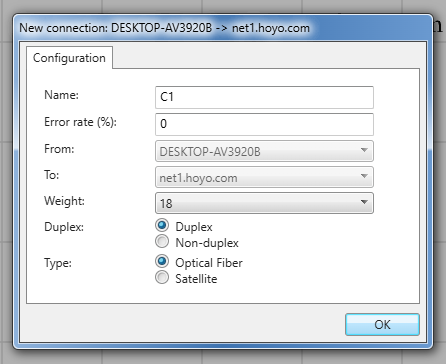


Рис. 2.11 – Вікно створення каналу

Каналу можна призначити тип, дуплексність, вагу та ймовірність помилки. Також можна обрати ім’я, яке має бути унікальним серед усіх об’єктів топології. Після створення каналу, на топології мережі він буде показаний як лінія (рис. 2.12).

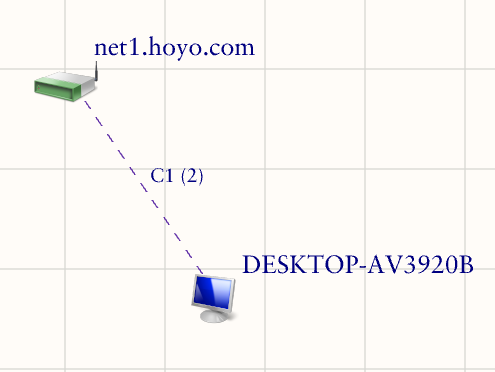


Рис. 2.12 – Топологія мережі з каналом повнодуплексним супутниковим каналом С1

При натисканні на канал правою кнопкою миші відкривається меню керування каналом (рис. 2.13), через яке можна канал видалити або відредагувати.

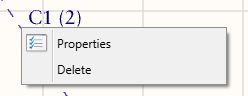


Рис. 2.13 – Меню керування каналом

Канали різного типу мають відмінний вигляд, приклади наведені в таблиці 2.1.

Таблиця 2.1

|  |  |  |
| --- | --- | --- |
| Вигляд | Дуплексність | Тип |
|  | Так | Оптоволоконний |
|  | Супутниковий |
|  | Ні | Оптоволоконний |
|  | Супутниковий |

У випадку якщо мережа стане занадто заплутаною та складною для сприйняття, можна скористатися функцією нормалізації графа топології у меню (рис. 2.5). Приклад роботи показано на рис. 2.14 та 2.15.

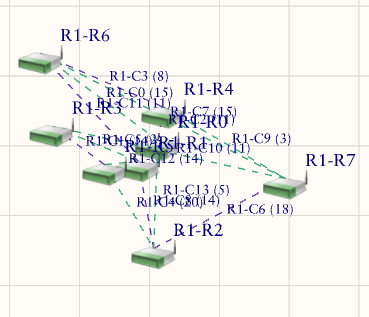


Рис 2.14 – Заплутана мережа

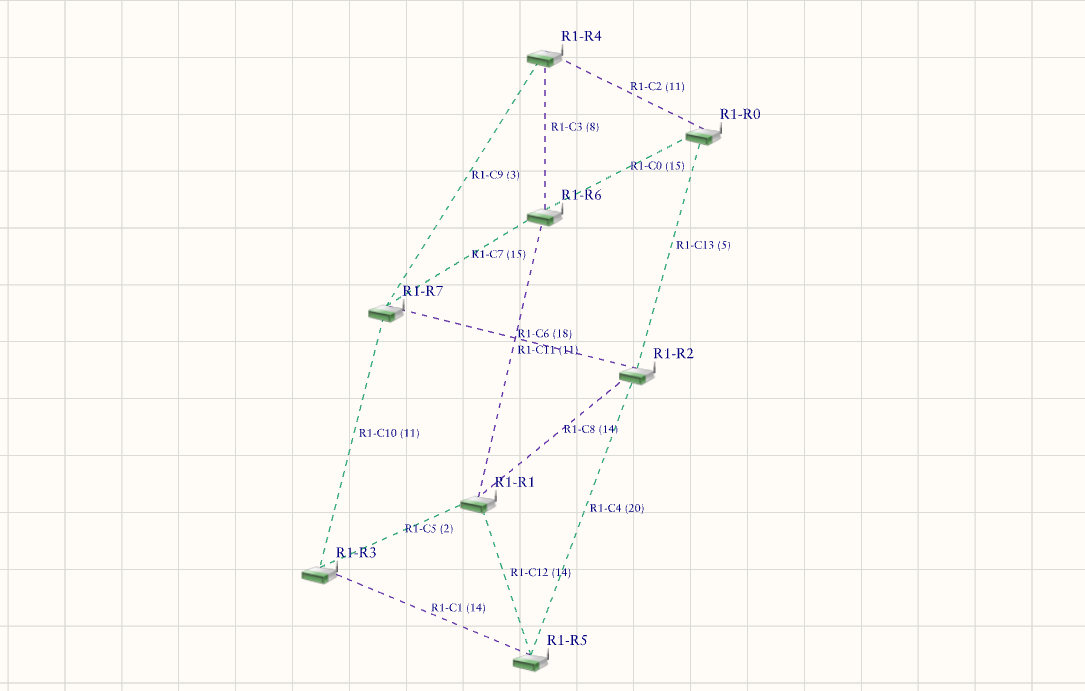


Рис. 2.15 – Мережа після нормалізації

Через меню утіліт для аналізу (рис. 2.4) можна побудувати найкоротший шлях між двома вузлами (рис. 2.16).

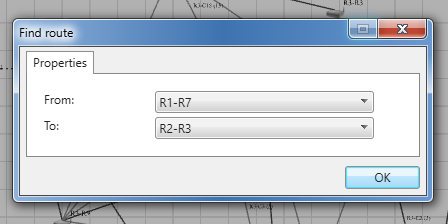


Рис. 2.16 – Налаштування пошуку маршруту

Після вибору вузлів, на топології мережі маршрут буде виділено червоним кольором (рис. 2.17). Червоний колір можна за бажанням прибрати, натиснувши на Remove highlights у меню (рис. 2.4).

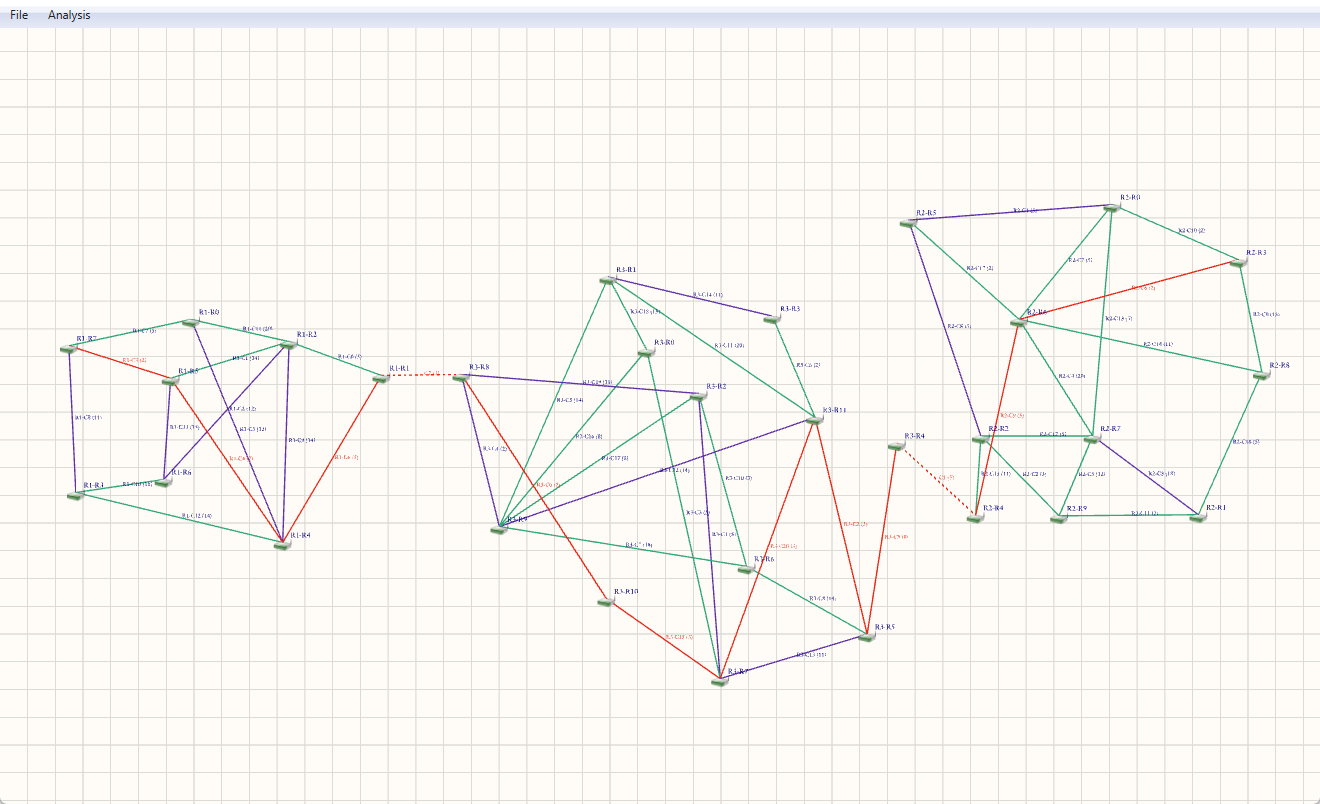


Рис 2.17 – Найкоротший шлях між двома вузлами

Іншою утілітою для аналізу є засіб відправки повідомлення, що відкривається (рис. 2.18) по натисканню на Trace packet у меню керування топологією мережі (рис. 2.4).

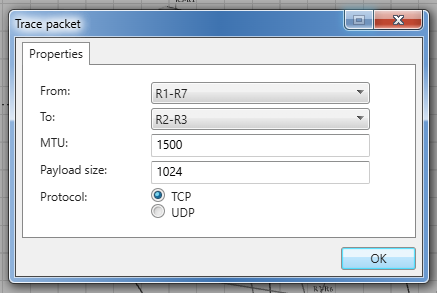


Рис. 2.18 – Вікно утіліти для аналізу відправки повідомлення

При натисканні на OK, утіліта моделює відправку повідомлення та виводить результати у форматі JSON (рис. 2.19).

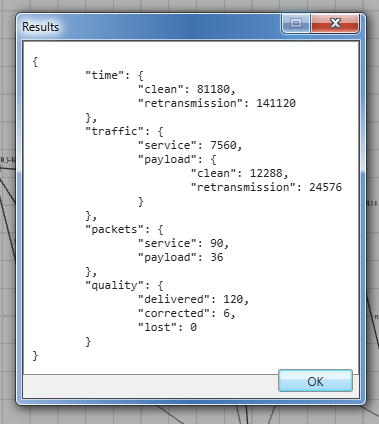


Рис 2.19 – Результати моделювання відправки повідомлення

Для проведення тестів з можливістю масового експорту результатів в таблицю Excel використовується засіб транзитивного аналізу (рис 2.20), що відкривається по натисканню на Transient analysis у меню керування топологією мережі (рис. 2.4).

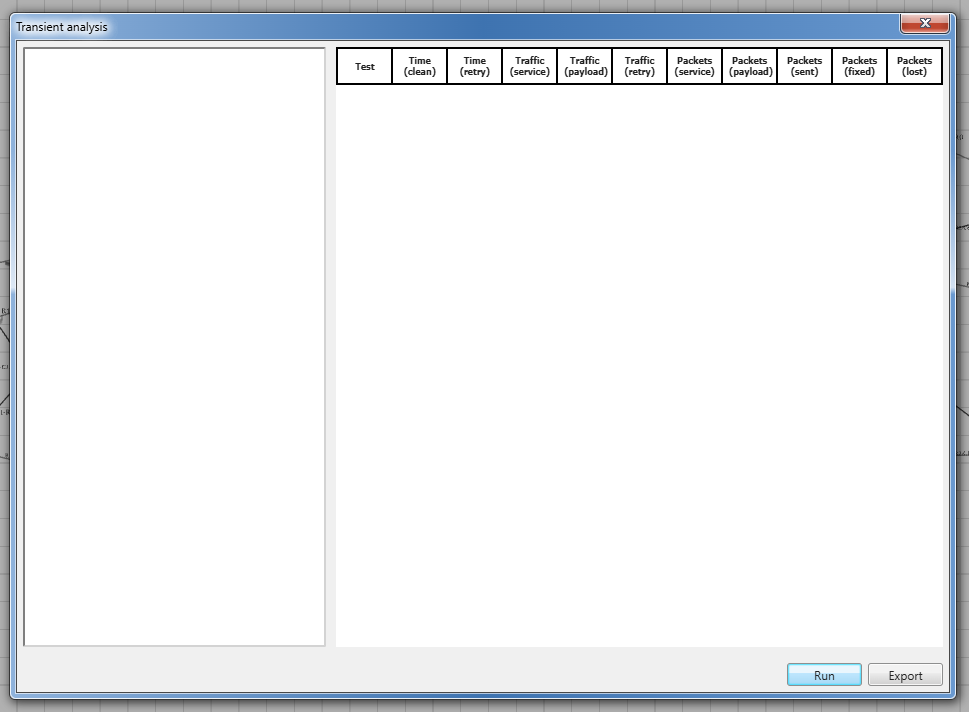


Рис. 2.20 – Екран аналізатору

Екран аналізатору поділяється на дві частини: зліва користувач має ввести сценарій тестування мовою CoffeeScript, а справа будуть відображені результати.

Сценарій тестування має доступ до моделі та додає тести, які аналізатор виконає по натисканню користувачем кнопки Run. Приклад сценарію тесту наведено в лістингу 2.1.

Лістинг 2.1

|  |
| --- |
| (addTest(hostA + ":" + hostB + " $", "tcp", hostA, hostB, 4470, 192000, 2) for hostB in Model.hostnames when hostB isnt hostA) for hostA in Model.hostnames |

Наведений вище сценарій тестування створить по 2 тести для кожної пари вузлів з наступними параметрами: MTU = 4470, Розмір повідомлення = 192000, Протокол = TCP.

Результати показані на рис. 2.21.

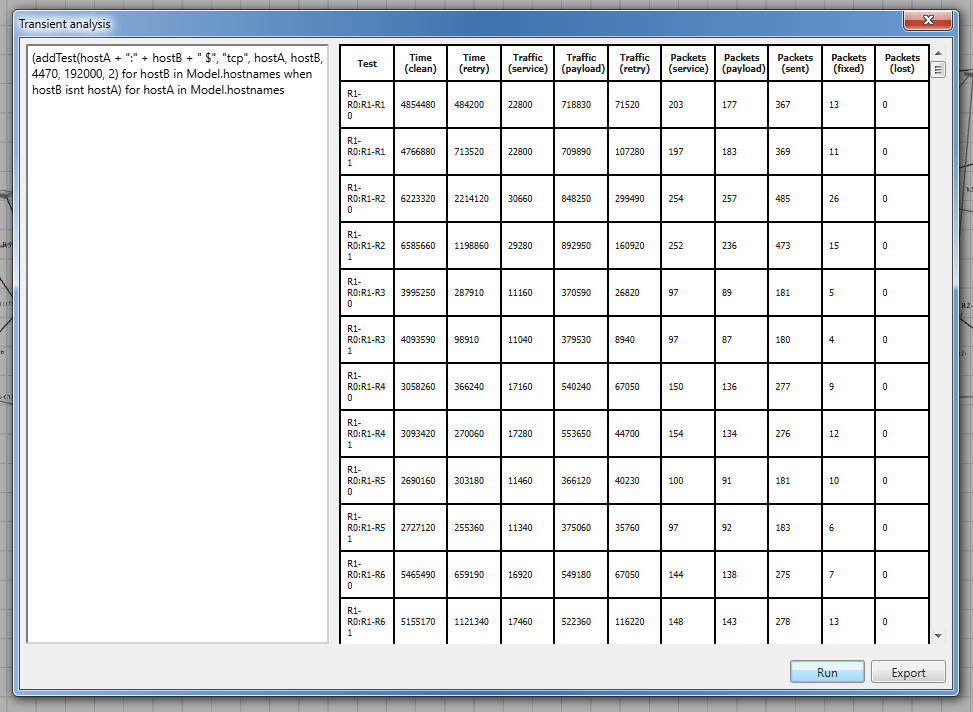


Рис. 2.21 – Екран аналізатору після завершення тестів

Варто зазначити, що аналізатор виводить не усі значення, які були отримані під час тестування, у таблицю через брак вільного місця на екрані. Щоб отримати додаткову інформацію, корисну для проведення досліджень, потрібно завантажити повну версію даних, натиснувши на кнопку Export. Результат експорту показаний на рис. 2.22.

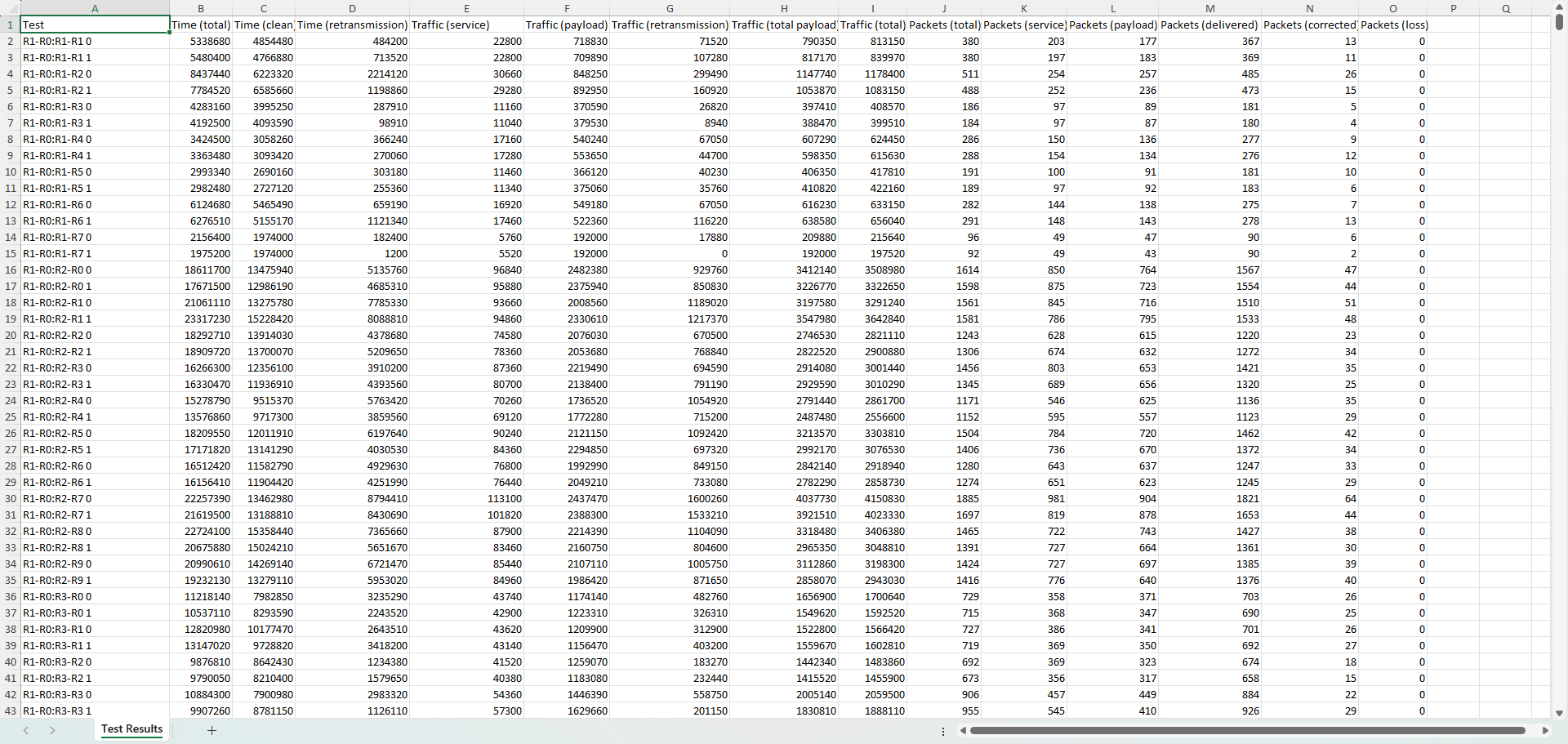


Рис. 2.22 – Вид повних даних тестування у програмі Microsoft Excel

**Дослідницька частина**

У цьому розділі курсового проекту розглянуто результати експерементального дослідження впливу різних властивостей мережі та конфігурації мережевих протоколів на їх ефективність та швидкодію. Основною метою тестування є визначення впливу розміру повідомлень, розміру пакетів, типу з’єднання та ймовірніості виникнення помилок на загальний час передачі, обсяг службого та інформаційного трафику, кількість ретрансмісій та втрачених пакетів. Для проведення експериментів було використано розробленний в рамках курсової роботи програмний продукт, що дозволяє створити план мережі та виконати моделювання відправки пакетів із різними налаштуваннями.

У рамках дослідження було виконано наступне:

1. Моделювання мережі: за допомогою розробленого ПП на кожний тест буде створено модель мережі, що поділяється на три регіональні мережі з 8 або більше вузлами, які з'єднані між собою супутниковими каналами. Параметри каналів будуть обрані випадковим чином ПП, якщо в описі тесту не зазначено інше.
2. Збір та аналіз даних: за допомогою функціоналу моделювання передачі пакетів ПП буде виконано збір інформації про час доставки, кількість втрачених та переданих заново пакетів. Ці дані будуть збережені у таблицю, по ним будуть створені графіки для більш зручного сприйняття інформації, по якій будуть зроблені висновки щодо залежності між параметрами мережі та ефективністю її роботи.
3. Порівняння результатів: на основі зібраної інформації буде виконано порівняння, що дозволить визначити найбільш та найменш сприятливі умови для застосування різних видів протоколів та виявити залежності між цими умовами та швидкістю, стабільністю роботи тощо.
4. Висновки: на основі порівняння будуть зроблені висновки щодо особливостей протоколів TCP та UDP, впливу параметрів мережі на її ефективність.

**Опис процесу тестування передачі повідомлень**

Для проведення тесту буде створено модель мережі за допомогою функції генерування мереж ПП. За варіантом, мережа буде включати у себе три регіональні мережі (з 8 та більше вузлами), що з’єднані між собою супутниковим каналом, а середній ступінь буде становити 3.5. Кожному каналу буде присвоєно випадкову вагу із множини {2, 3, 5, 8, 11, 12, 14, 15, 18, 20} та несупутниковий тип. Ймовірність помилки визначається випадково у проміжку від 0 до 5%. Супутиникові канали, що об'єднують регіональні мережі між собою будуть створені вручну, оскільки в ПП не передбачено генерацію топології мережі цілком. Типові налаштування генератора регіональних мереж зображенні на рис 3.1, типові налаштування супутникового каналу зв’язку між регіональними мережами зображено на рис 3.2, приклад мережі із 3 згенерованими регіональними мережами, об’єднані супутниковими каналами показано на рис. 3.3. Для деяких тестів ці параметри можуть бути змінені, інформація про це буде надана у відповідних розділах.

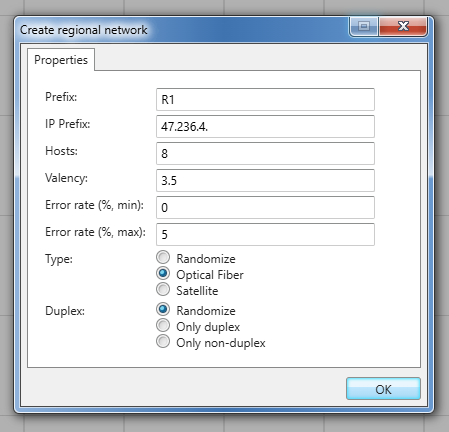


Рис. 3.1 – Налаштування генератора регіональних мереж

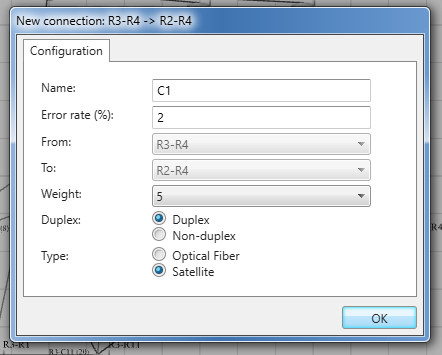


Рис. 3.2 – Налаштування супутникового каналу

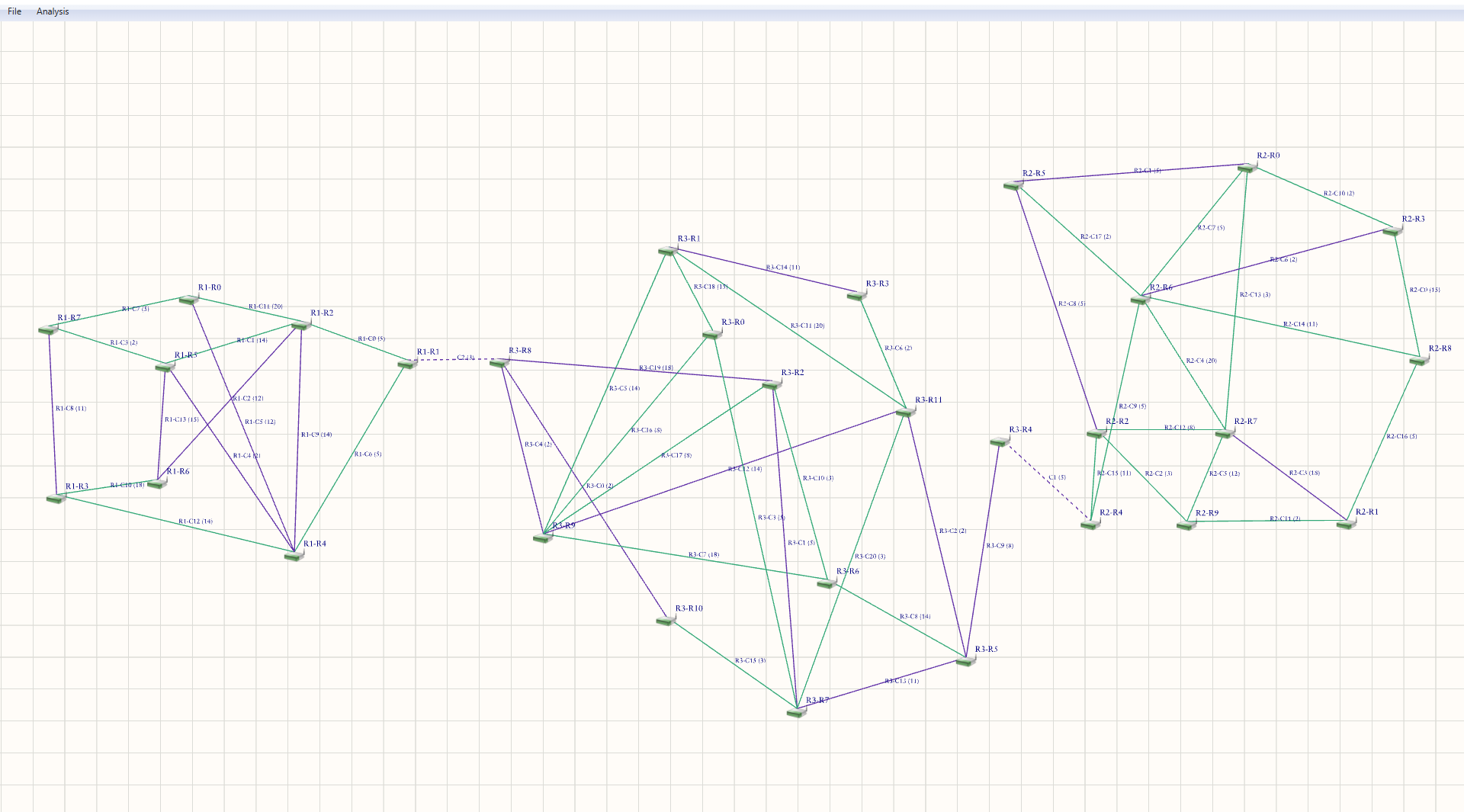


Рис. 3.3. – Приклад сгенерованої мережі

**Тестування передачі повідомлень в залежності від величини повідомлення**

Для визначення залежності часу передачі повідомлення від величини створимо тест із наступними параметрами (рис. 3.4):

* Розмір пакету – 1500
* Розмір повідомлення – від 10 \* 1500 до 50 \* 1500
* Ймовірність виникнення помилки – випадкова, від 0% до 5%
* Ваги каналів – випадкові
* Типи каналів – випадкові

Для тестування був обрані такі вузли, між якими шлях містить велику кількість транзитних ділянок, що знаходяться в різних регіональних мережах (рис 3.5). Результати подані в таблиці 3.1.

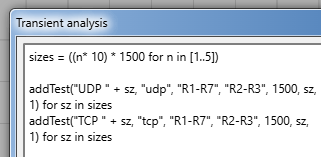


Рис 3.4 – Сценарій тесту

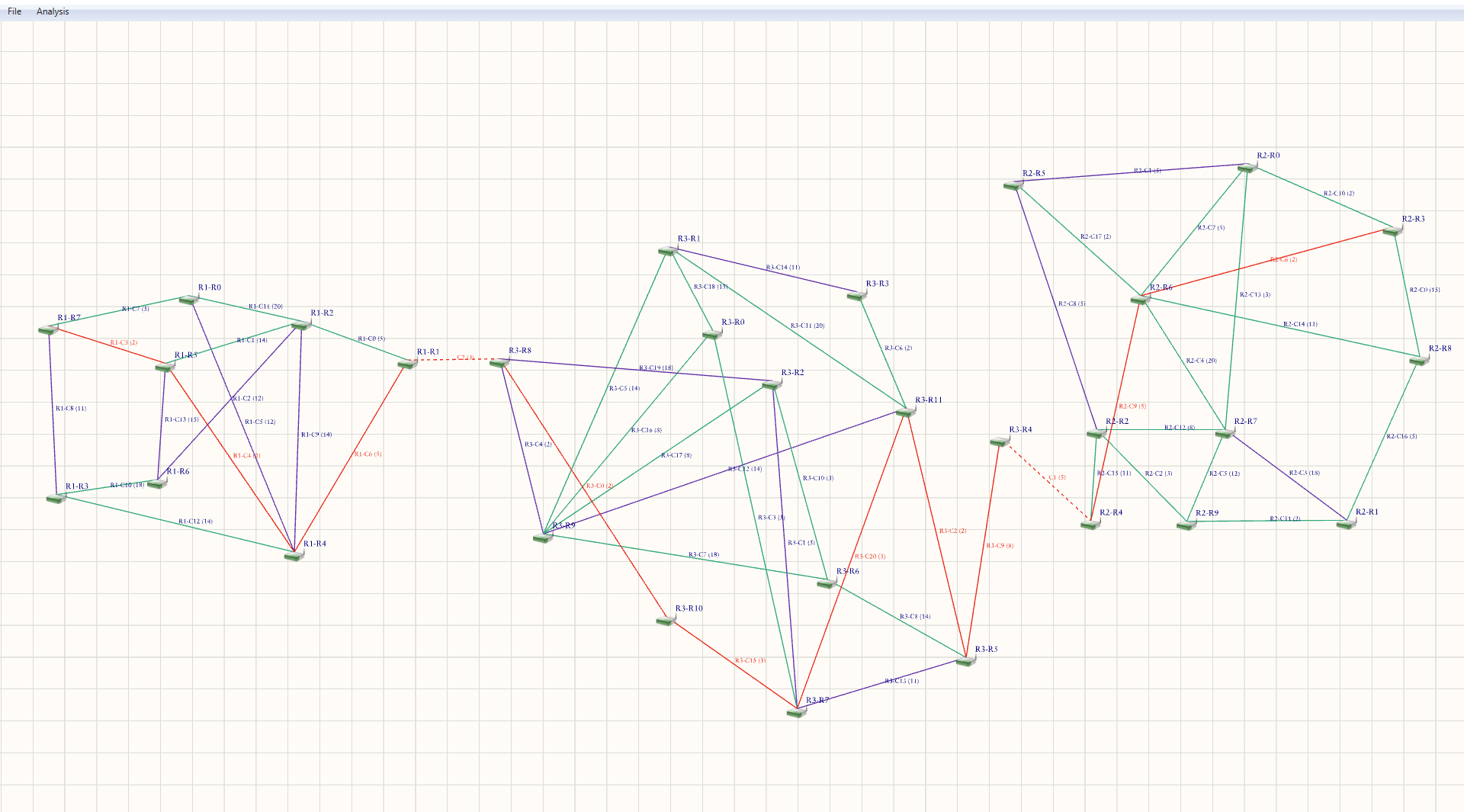


Рис. 3.5 – Шлях між вузлами R1-R7 та R2-R3

Таблиця 3.1

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Протокол | Розмір | MTU | Час | | Службовий трафік | Пакети | | | |
| Всього | Корекція | Всього | Службові | Корекція | Втрата |
| UDP | 15000 | 1500 | 866376 | 0 | 3192 | 114 | 0 | 0 | 2 |
| 30000 | 1740392 | 6412 | 229 | 4 |
| 45000 | 2444800 | 9016 | 322 | 8 |
| 60000 | 3153792 | 11648 | 416 | 12 |
| 75000 | 4163800 | 15372 | 549 | 12 |
| TCP | 15000 | 1039140 | 123540 | 21660 | 361 | 236 | 10 | 0 |
| 30000 | 2507760 | 786720 | 39780 | 663 | 353 | 17 |
| 45000 | 3257280 | 559260 | 51060 | 851 | 449 | 13 |
| 60000 | 4676160 | 1023900 | 70200 | 1170 | 591 | 25 |
| 75000 | 6342720 | 2076300 | 97980 | 1633 | 848 | 49 |

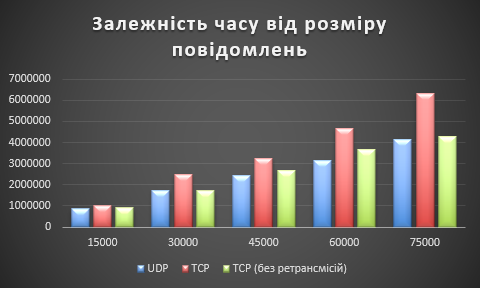


Рис. 3.6 – Залежність часу від розміру повідомлення



Рис 3.7 – Залежність службового трафіку від розміру повідомлення

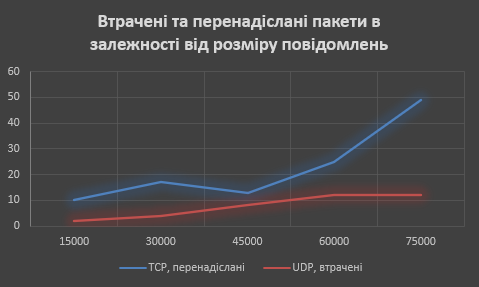


Рис 3.8 – Залежність втрачених та перенадісланих пакетів від розміру повідомлення

**Аналіз:**

1. Втрати пакетів: На відміну від TCP, UDP втрачає пакети, кількість втрачениї пакетів становить від 2 до 12 та лінійно зростає у залежності від розміру повідомлення (рис. 3.8). Це свідчить про те, що цей протокол не забезпечує гарантію доставки і зі збільшенням кількості пакетів зростає кількість таких, що не будуть доставлені.
2. Час передачі (рис 3.6): Бачимо що при використанні протокола UDP маємо лінійну залежність часу передачі повідомлення в залежності від його розміру. При цьому у TCP залежність вже є квадратичною. Але якщо не враховувати час витрачений на повторну передачу пакетів, залежність стане лінійною і час буде не сильно відрізнятись від того, який був би при використанні UDP. Це свідчить про те, що TCP потрібно більше часу на передачу повідомлення через більшу кількість службового трафіку (рис 3.7), але за умови відсутності помилок, швидкість буде не сильно гіршою ніж у UDP.
3. Службовий трафік (рис 3.7): UDP потребує значно меншу кількість службового трафіку ніж TCP, у якого кількість службового трафіку росте значно швидше. Це пов'язано із тим, що протокол TCP використовує службовий трафік для контролю стану з’єднання та передачі інформації про необхідність перевідправки або підтвердження отримання.
4. Кількість перевідправок (рис 3.8): Чим більшим є розмір повідомлення, тим більше TCP потрібно перевідправляти пакети в умовах, коли канали зв’язку не є ідеальними і втрачають дані, саме це і впливає на час доставки повідомлення.

**Висновки:**

Аналіз показав, що UDP має кращі показники швидкодії як і за часом так і за кількістю пакетів. Цей протокол економить ресурси за допомогою зменшення частки службового трафіку, проте це призводить до неможливості забезпечити гарантовану доставку повідомлень: орієнтовно 2% пакетів втрачається. Хоча кількість втрат росте із розміром повідомлення, тести показали що їх кількість у процентному співвідношені є доволі стабільною та знаходиться в проміжку від 1.7% до 2%, що робить втрати прогнозованими.

TCP, якщо не враховувати повторні передачі пакетів, за швидкістю є дещо повільнішим за UDP, але не сильно: в цьому протоколі на початку та кінці з’єднання надсилаються службові пакети, а після передачі фрагментів надходять підтвердження. Проте ці службові пакети є маленькими і вони не сильно збільшують час доставки, якщо повторних передач немає. Якщо ж канали зв’язку не ідеальні, TCP показує значно гіршу швидкодію ніж UDP, адже виконує повторну передачу кожного пакету, що був втрачений.

Отже, UDP є більш швидким протоколом, але TCP є більш надійним і його швидкість може бути близькою до такої, що є у UDP в залежності від кількості помилок. Цю закономірність буде досліджено у наступному тесті.

**Тестування впливу ймовірності виникнення помилок на доставку повідомлень**

Дослідимо як впливає на роботу протоколів якість каналів зв’язку та коректність роботи транзитних вузлів. Для цього тестування візьмемо мережу, що використовувалась у попередніх тестах (рис 3.3), але перед початком кожного тестування змінимо ймовірність виникнення помилки. Використаємо наступні параметри:

* Розмір пакету – 1500
* Розмір повідомлення – 3072000
* Ймовірність виникнення помилки – випадкова, проміжок змінний
* Ваги каналів – випадкові
* Типи каналів – випадкові

Шлях оберемо між двома вузлами, що розташовані далеко один від одного як у попередньому тесті. Результати подані в таблиці 3.2.

Таблиця 3.2

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Прото | Помилка | Час | | Службовий трафік | Пакети | | | |
| Всього | Корекція | Всього | Службові | Корекція | Втрата |
| UDP | 0% | 187760640 | 0 | 688128 | 114 | 0 | 0 | 0 |
| До 2% | 175406760 | 645736 | 229 | 257 |
| До 4% | 170599672 | 628264 | 322 | 409 |
| До 8% | 155270776 | 574728 | 416 | 738 |
| До 16% | 118290120 | 444080 | 549 | 1394 |
| TCP | 0% | 199080000 | 2952000 | 361 | 236 | 10 | 0 |
| До 2% | 213031920 | 27963900 | 3190380 | 663 | 353 | 17 |
| До 4% | 228333540 | 48225720 | 3398280 | 851 | 449 | 13 |
| До 8% | 257551140 | 91498020 | 3871140 | 1170 | 591 | 25 |
| До 16% | 395043060 | 269296380 | 6023580 | 1633 | 848 | 49 |

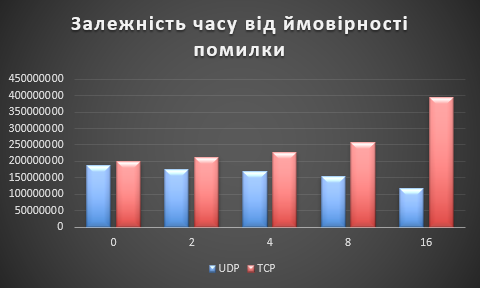


Рис. 3.9 – Залежність часу від ймовірності помилки



Рис. 3.10 – Залежність кількості службового трафіку від ймовірності помилки



Рис. 3.11 – Залежність кількості перенадісланих пакетів в залежності від ймовірності помилки

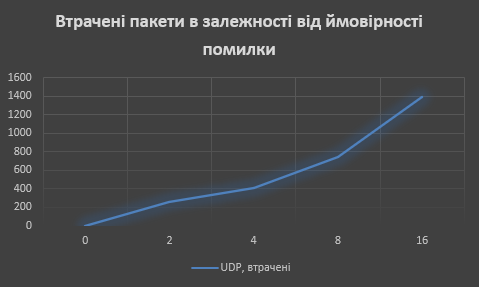


Рис. 3.11 – Залежність кількості втрачених пакетів в залежності від ймовірності помилки

**Аналіз:**

За графіками видно, що для UDP зі зростанням ймовірності помилок лінійно росте кількість втрачних пакетів (3.11), cаме це і впливає на те, що час відправки зменшується (3.9). Це відбувається через те, що якщо пакет, що був втрачений на певному транзитному вузлі не пересилається на наступних та, відповідно, усі вузли витрачають менше часу і трафіку (3.10) на відправку повідомлення. Проте варто звернути увагу на те, що зі зростанням ймовірності помилки зростає і частина повідомлення, що не буде отримана.

У випадку з TCP час доставки збільшується експоненціально зі збільшенням ймовірності помилки, це відбувається через те що у випадку втрати пакета, TCP його перенадсилає ще раз через всі вузли. Цю тенденцію можна побачити на графіку залежності службового трафіку від ймовірності помилки (рис. 3.10) та залежності кількості перенадісланих пакетів (рис. 3.11).

Проте коли кількість помилок близька до 0, час витрачений на доставку протоколами TCP та UDP відрізняється не так радикально. Це підтверджує гіпотезу викладену у попередньому тесті.

**Висновки:**

Аналіз показав, що протокол TCP є не сильно повільнішим за UDP в ідеальних умовах, де кількість помилок близька до 0. Проте в таких умовах використовувати TCP немає сенсу, адже час все одно збільшується без покращення якості: в ідеальних умовах UDP також гарантовано доставить повідомлення.

Протокол TCP ж забезпечує гарантовану доставку повідомлень але ціною експоненціального зросту службового трафіку по мірі погіршення якості зв’язку.

Отже, TCP є менш швидким протоколом, але більш надійним. При виборі між TCP та UDP варто враховувати наскільки швидкість є більш пріоритетною ніж надійність передачі.

**Тестування передачі повідомлень в залежності від велечини пакету**

Дослідимо як впливає на роботу протоколів розмір MTU при сталому розмірі повідомлення. Для цього тестування візьмемо мережу, що використовувалась у попередніх тестах (рис 3.3), але перед початком кожного тестування змінимо MTU. Використаємо наступні параметри:

* Розмір пакету – змінний, значення надані в таблиці 3.3
* Розмір повідомлення – 3072000
* Ймовірність виникнення помилки – випадкова, від 0 до 5%
* Ваги каналів – випадкові
* Типи каналів – випадкові

Шлях оберемо між двома вузлами, що розташовані далеко один від одного як у попередньому тесті. Результати подані в таблиці 3.4.

Таблиця 3.3

|  |  |
| --- | --- |
| MTU | Використання |
| 1280 | Мінімальний розмір пакету IPv6 |
| 1500 | Стандартний розмір пакету в інтернеті |
| 4470 | Розмір пакету в опірних мережах інтернету |
| 9000 | Jumbo-пакети |
| 16384 | Пакети в деяких InfiniBand мережах |

Таблиця 3.4

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Прото | MTU | Час | | Службовий трафік | Пакети | | | |
| Всього | Корекція | Всього | Службові | Корекція | Втрата |
| UDP | 1280 | 166797075 | 0 | 718382 | 25801 | 0 | 0 | 671 |
| 1500 | 166111172 | 612578 | 21839 | 569 |
| 4470 | 163589038 | 205209 | 7067 | 196 |
| 9000 | 164067715 | 102572 | 3709 | 93 |
| 16384 | 161844495 | 55745 | 1918 | 54 |
| TCP | 1280 | 247336268 | 72565680 | 4277280 | 71052 | 35605 | 1866 | 0 |
| 1500 | 247091790 | 71467020 | 4215582 | 61002 | 30462 | 1841 |
| 4470 | 232746936 | 65400810 | 1232022 | 20473 | 10195 | 542 |
| 9000 | 233498511 | 64931700 | 1172628 | 10292 | 5107 | 520 |
| 16384 | 226233134 | 53770736 | 339060 | 5540 | 2771 | 151 |



Рис. 3.12 – Залежність часу доставки від розміру пакета ТСР

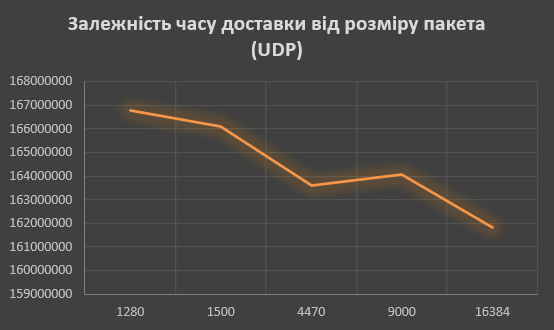


Рис. 3.13 – Залежність часу доставки від розміру пакета UDP



Рис. 3.14 – Залежність службового трафіку від розміру пакета ТСР



Рис. 3.15 – Залежність службового трафіку від розміру пакета UDP



Рис. 3.16 – Залежність кількості перенадсилань від розміру пакета ТСР



Рис. 3.17 – Залежність кількості перенадсилань від розміру пакета UDP

**Аналіз:**

За графіками видно, що збільшення розміру пакету (MTU) позитивно впливає на низку показників. Зокрема, час доставки для обох протоколів (рис. 3.12 та рис. 3.13) зменшився. Для TCP на 9%, для UDP дещо менше, на 3%. Це пояснюється тим, що зі збільшенням MTU зменшується і кількість пакетів, а відповідно спадає і кількість службового трафіку, який як і в ТСР, так і в UDP, залежить від кількості пакетів і потребує часу для доставки.

Цю тенденцію можна прослідкувати на графіках 3.14 та 3.15, на яких показано зменшення службового трафіку зі збільшенням розміру пакета. Також за ними видно, що у TCP службового трафіку більше, відповідно і приріст швидкості у TCP є більшим, ніж у UDP.

Відповідно, кількість помилок також зменшується, що позитивно впливає на кількість перевідправок (рис. 3.16) та втрат (рис. 3.17). У рамках цього тестування кількість перенадсилань зменшилась на 92% (з 1866 до 151 при переключенні з MTU=1280 на MTU=16384), кількість втрат також зменшилась на 92% (з 671 до 54).

Проте варто зазначити, що для UDP кількість втрачної інформації не змінилась, бо MTU не впливає на ймовірність помилки, але впливає на кількість інформації, що передається і, відповідно, може «загубитись» за раз. Тому, хоч загубилось і менше пакетів, процентне співвідношення втрат залишилось таким самим.

**Висновки:**

Збільшення розміру пакету позитивно впливає на швидкодію мережі, в основному через зменшення кількості службового трафіку, об’єм якого залежить від кількості пакетів. Але % втрати інформації при використанні UDP залишився сталим.

**Тестування передачі повідомлень в залежності від типу каналу**

Для дослідження впливу типу каналу на передачу повідомлень створимо тест із наступними параметрами:

* Розмір пакету – 1500
* Розмір повідомлення – 3072000
* Ймовірність виникнення помилки – 0%
* Ваги каналів – випадкові
* Типи каналів – змінні

Шлях оберемо між двома вузлами, що розташовані далеко один від одного як у попередньому тесті. Результати надані в таблиці 3.5.

Таблиця 3.5

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Прото | Тип каналів | Час | | | Службовий трафік | Пакети | | | |
| Всього | Кор. | x | Всього | Службові | Корекція | Втрата |
| UDP | Змішаний | 200278016 | 0 | 0% | 688128 | 24576 | 0 | 0 | 0 |
| Дуплексний | 200278016 | 688128 | 24576 |
| Напівдуплексний | 200278016 | 688128 | 24576 |
| TCP | Змішаний | 291604792 | 45% | 2952000 | 49200 | 24624 |
| Дуплексний | 207948664 | 3.8% | 2952000 | 49200 | 24624 |
| Напівдуплексний | 416578274 | 108% | 2952000 | 49200 | 24624 |

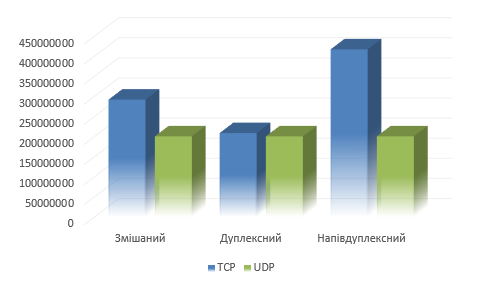


Рис. 3.18 – Залежність часу доставки від типу каналів

**Аналіз:**

Тип каналу не впливає на швидкість передачі даних протоколом UDP, час передачі якого залишився сталим.

TCP повільніше UDP приблизно на 3.8% за умови, що всі канали зв’язку дуплексні. У змішаному середовищі (50% ймовірність того, що канал стане дуплексним) TCP виявився повільніше вже на 45%, а у напівдуплексному на 108%. Це свідчить про те, що TCP є більш чутливим до неможливості одночасного пересилання даних в обидві сторони.

**Висновки:**

TCP показує себе не так гарно як UDP у середовищах із напівдуплексними каналами, але, як було показано у попередніх тестах, є більш надійним.

**Висновки**

В рамках даного курсового проекту було детально проаналізовано протоколи TCP та UDP, було визначено їхню поведінку при різних налаштуваннях передачі, що дозволяє зробити наступні висновки:

1. **Швидкість:**
   1. UDP є швидшим за TCP незалежно від типу каналу: протокол показав себе краще у дуплексному, напівдуплексному та змішаному середовищі на 3.5%, 108% та 45% відповідно. Час доставки збільшується по мірі зростання розміру повідомлення, але лінійно. Таку швидкість можна пояснити відносною простотою протоколу: в ньому зовсім немає механізму синхронізації, що унеможливлює забезпечети гарантовану доставку та порядок надходження пакетів. Це суттєво впливає на цілісність даних, але також і значно зменшує накладні витрати на синхронізацію між вузлами.
   2. TCP має гірші показники по швидкості, навіть у середовищі із дуплексними каналами. Проте в ідеальних умовах (ймовірність помилки = 0%) ця різниця не є дуже великою і пов’язана у збільшеній кількості службового трафіку, який потрібен для синхронізації між вузлами та встановлення/закінчення з’єднання. Оскільки при синхронізації службові пакети відправляються в обидві сторони, протокол TCP має суттєво гіршу ефективність у середовищах із напівдуплексними каналами: у рамках тестування було визначено, що час відправки може збільшитись більше, ніж в два рази (на 108%). Також на швидкість впливає і ймовірність помилок, оскільки TCP перенадсилає пакет, коли той був втрачений: при збільшенні ймовірності помилок з 0 до 16% час доставки повідомлення зріс на 50% (табл. 3.2). Збільшення розміру пакету (MTU) позитивно впливає на час доставки (табл 3.4).
2. **Службовий трафік:**
   1. UDP не генерує службові пакети, вся службова інформація знаходиться у заголовках пакетів з корисним навантаженням, від чого кількість службового трафіку є низькою та прямо залежить від кількості пакетів: зі збільшенням MTU та/або зменшенням довжини повідомлення кількість службового трафіку зменшується.
   2. TCP генерує службові пакети під час встановлення та завершення з’єднання а також для синхронізації між вузлами. Кількість службового трафіку росте із кількістю пакетів, тому збільшення MTU та зменшення розміру повідомлення також як і в UDP зменшує кількість службового трафіку. Зі збільшенням ймовірності помилки, кількість службового трафіку зростає, адже виконуються перенадсилання, на які теж потрібно відправити пакети з підтвердженнями отримання, до того ж про втрачені пакети вузол-отримувач також має повідомляти синхронізаційним пакетом.
3. **Завадостійкість та надійність:**
   1. UDP не має жодних механізмів перенадсилання. Відповідно, із зростанням ймовірності помилки лінійно збільшується частка повідомлення, яка не буде доставлена (рис. 3.11).
   2. TCP забезпечує гарантію доставки повідомлень у тому порядку, у якому вони були відправлені. Це досягається за допомогою механізмів синхронізації та перевідправки, що суттєво збільшують кількість трафіку у порівнянні з UDP (рис. 3.10). Час доставки також експоненціально зростає зі збільшенням ймовірності помилки (рис. 3.9).
4. **Сфера застосування:**
   1. UDP можна використовувати у випадках, коли заради швидкості можна знехтувати цілісністю повідомлення. Наприклад, для передачі мультмедіа-потоків: втрата або отримання в неправильному порядку індивідуальних пакетів навряд чи суттєво вплине на сприйняття відео або аудіо. Також UDP можна використовувати для протколів, де розмір повідомлення не перевищує розмір пакету, бо в такому випадку можна тривіально реалізувати свою реалізацію перенадсилання, що працюватиме на таймаутах. Таким чином працює протокол DNS, який невеликі запити та відповіді надсилає через UDP одним пакетом, тим самим зберігаючи час, який би був витрачений на відкриття з’єднання та синхронізацію, яку тут можна замінити повтором запиту у випадку відсутності відповіді.
   2. TCP потрібно використовувати у випадках, коли потрібна гарантія доставки усіх фрагментів повідомлення в правильному порядку. Така потреба виникає при передачі файлів або виконанні запитів, розмір яких може бути більше розміру пакету. Наприклад, протоколи HTTP, SMTP, SSH, FTP використовують TCP, адже інформація, яку вони передають легко може бути більшою за розмір пакету, але при цьому втрата або пошкодження навіть одного біту корисного навантаження може призвести до некоретних результатів.

Отже, розглянуті протоколи мають свої переваги та недоліки, але обидва можуть бути використані ефективно у різних сферах застосування. Для задач, де гарантована доставка та правильний порядок отримання пакетів не є важливими, UDP буде доречним вибором. В решті випадків доцільно розглянути можливість використання більш надійного, хоч і повільнішого, протоколу TCP.

**Список використаної літератури**

Мартинова О.П. Комп'ютерні мережі: конспект лекцій.

Орлова М.М. Теоретичні відомості.

Dynamic Routing Protocols: OSPF, EIGRP, RIPv2, IS-IS, BGP [Електронний ресурс] – Режим доступу до ресурсу: <https://community.cisco.com/t5/networking-knowledge-base/dynamicrouting-protocols-ospf-eigrp-ripv2-is-is-bgp/ta-p/4511577>

Bellman–Ford Algorithm [Електронний ресурс] – Режим доступу до ресурсу: <https://www.geeksforgeeks.org/bellman-ford-algorithm-dp-23/>

Комп’ютерні мережі: навчальний посібник [Текст] / Азаров О.Д., Захарченко С.М., Орлова М.М., Тарасенко В.П.. – Підручник. - Вінниця: ВНТУ. – 2022. – 385 с.

Комп’ютерні мережі. Ч. 2 : навчальний посібник / І. Р. Арсенюк, А. А. Яровий. – Вінниця : ВНТУ, 2010. – 145 с.