**ЗМІСТ**

[ПЕРЕЛІК](#_Toc421274321) СКОРОЧЕНЬ 2

[ВСТУП](#_Toc421274322) 3

[1.ОСНОВНІ ТЕОРЕТИЧНІ ВІДОМОСТІ](#_Toc421274323) 4

[2.МОДИФІКОВАНИЙ АЛГОРИТМ ДЕЙКСТРИ](#_Toc421274325) 7

[3.ОПИС РОЗРОБЛЕНОЇ ПРОГРАМИ](#_Toc421274329) 10

[3.1. Інтерфейс програми](#_Toc421274326) 17

3.2. Приклад роботи програми\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_18

3.3. Результати роботи програми 19

ВИСНОВКИ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_23

СПИСОК ЛІТЕРАТУРИ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_24

# 

# ПЕРЕЛІК СКОРОЧЕНЬ

ЕОМ – електрона обчислювальна машина;

МПД – Мережі передачі даних;

UDP – User Datagram Protocol, іноді, розшифровують як Unreliable Datagram Protocol;

TCP – Transmission Control Protocol;

BECN - Backward Explicit Congestion Notification;

FECN - Forward Explicit Congestion Notification.

# ВСТУП

На даний момент використання технологій використання моделювання мережі передачі даних (МПД) є необхідним, адже створення ефективної топології мережі є умовою подальшого розвитку і нормального фунціонування її. Моделювання МПД дозволяє виявити та виправити всі допущені помилки як у організації структури мережі, так і у виборі режимів маршрутизації. Використання моделювання МПД дозволяє ще на етапі розробки дослідити завантаження каналів передачі даних, час передачі повідомлень, дозволяє проаналізувати вже існуючі мережі більш точно, оскільки модель дозволяє аналізувати весь маршрут передачі даних.

Для розподілу трафіку в МПД використовуються різні алгоритми маршрутизації, що дозволяють оптимізувати декілька параметрів МПД і підвищити її характеристики. Кожний алгоритм маршрутизації має свою область застосування та свої переваги і недоліки.

Таким чином, для створення максимально ефективної структури та організації МПД необхідно використовувати можливості моделювання, що дозволять оптимізувати систему. З цього випливають певні вимоги до засобів моделювання. Вони повинні максимально точно відображати параметри, що імітуються і бути максимально гнучкими у задані налаштувань.

1. ОСНОВНІ ТЕОРЕТИЧНІ ВІДОМОСТІ

Маршрут - це послідовність портів маршрутизаторів, які повинен пройти пакет від джерела до адресата. Сам маршрутизатор адреси не має, а кожний порт маршрутизатора має числову і локальну адресу тієї мережі, до складу якої він входить. Тип і формат числових адрес залежить від стека комунікаційних протоколів, який використовується в об’єднаній мережі. Здебільшого числові адреси складаються з номера мережі, в якій знаходиться адресат, та номера кінцевого вузла в цій мережі.

Маршрутизація – це процес визначення маршруту передачі даних в мережу. Для підтримки маршрутизації в мережі має використовуватись протокол з підтримкою маршрутизації. Такий протокол визначає формат пакету, в заголовку якого вказується адреси отримувача та відправника та інші поля, що можуть знадобитись для виконання маршрутизації.

Є два типи маршрутизації: статична і динамічна.

Алгоритм маршрутизації визначає спосіб заповнення таблиці маршрутизації. Існує велика кількість алгоритмів маршрутизації та їх програмних реалізацій, але їх кінцеве завдання – заповнення таблиці маршрутизації коректною актуальною інформацією.

Статичні алгоритми маршрутизації використовують таблиці маршрутизації, що створюються системним адміністратором і, як правило, не змінюються під час роботи мережі.

Динамічні алгоритми з деяким періодом оновлюють таблиці маршрутизації. Маршрутизатори, що використовують дані алгоритми обмінюються інформацією про відстані та таким чином враховують завантаженість мережі. Але це може призвести до використання значного обсягу пропускної спроможності під службові повідомлення.

Існує два методи пакетної комутації: дейтаграмний (датаграмний) і спосіб віртуальних з'єднань.

Дейтаграмний метод ефективний для передачі коротких повідомлень. Він не вимагає громіздкої процедури встановлення з'єднання між абонентами.

Пакети надходять на прийом не в тій послідовності, в якій вони були передані, тому доводиться виконувати функції, пов'язані із збіркою пакетів. Отримавши дейтаграму, вузол комутації направляє її в бік суміжного вузла, максимально наближеного до адресата. Коли суміжний вузол підтверджує отримання пакету, вузол комутації стирає його у своїй пам'яті. Якщо підтвердження не отримано, вузол комутації відправляє пакет в інший суміжний вузол, і так до тих пір, поки пакет не буде відправлений.

Всі вузли, що оточують даний вузол комутації, ранжуються за ступенем близькості до адресата, і кожному присвоюється перший, другий або інший ранг. Пакет спочатку надсилається у вузол першого рангу, при невдачі - у вузол другого рангу тощо. Ця процедура називається алгоритмом маршрутизації. Існують алгоритми, коли вузол передачі вибирається випадково, і тоді кожна дейтаграма йтиме за випадковою траєкторії. Дейтаграмний режим використовує протокол UDP.

Дейтаграмний метод працює швидко, оскільки ніяких попередніх дій перед відправленням даних проводити не потрібно. Однак при такому методі важко перевірити факт доставки пакета вузлу призначення.

Режим віртуального з’єднання передбачає попереднє встановлення маршруту передачі всього повідомлення від відправника до одержувача за допомогою спеціального службового пакету - запиту виклику.

Для цього пакету вибирається маршрут, який у разі згоди одержувача цього пакету на з'єднання закріплюється для проходження по ньому всього повідомлення. Пакет запиту на з'єднання як би прокладає через мережу шлях, по якому підуть всі пакети, що відносяться до цього виклику.

Метод називається віртуальним тому, що тут не комутується реальний фізичний тракт (як, наприклад, в телефонній мережі), а встановлюється логічний зв'язок між відправником та одержувачем, - тобто комутується віртуальний (уявний) тракт.

У віртуальній мережі абоненту-одержувачу направляється службовий пакет, який прокладає віртуальне з'єднання. У кожному вузлі цей пакет залишає розпорядження виду: пакети k-го віртуального з'єднання, що прийшли з i-го каналу, слід направляти в j-й канал. Тим самим віртуальне з'єднання існує тільки в пам'яті керуючого комп'ютера. Дійшовши до абонента-отримувача, службовий пакет запитує у нього дозвіл на передачу, повідомивши, який обсяг пам'яті знадобиться для прийому. Якщо цей комп'ютер має потрібну пам'ять і є вільним, то абоненту-відправнику надсилається згода на передачу повідомлення. Отримавши підтвердження, абонент-відправник приступає до передачі повідомлення звичайними пакетами. Режим віртуального каналу використовує протокол TCP.

1. МОДИФІКОВАНИЙ АЛГОРИТМ ДЕЙКСТРИ

Традиційні методи маршрутизації в мережах з комутацією пакетів передбачають вибір найкоротшого шляху для доставки іформаційного пакета з точки відправлення у точку призначення навіть, якщо існують інші, хоч і дещо гірші маршрути. Це призводить до того, що у випадку, коли найкоротший шлях є перевантаженим, інформаційні пакети продовжують надсилатися цим шляхом, нагромаджуючи черги. Зменшити або усунути перевантаження найкоротшого шляху можна за рахунок тимчасового переходу на інший маршрут, який знаходять за модифікованим алгоритмом Дейкстри. Після зниження завантаженості найкоротшого шляху до допустимого рівня відновлюється передача пакетів найкоротшим шляхом. Прикладами повідомлень про переванта- жені шляхи є повідомлення про перевантаження каналу у зворотному напрямку BECN (Backward Explicit Congestion Notification) та повідомлення про перевантаження каналу в прямому напрямку FECN (Forward Explicit Congestion Notification).

Модифікація алгоритму полягає у вилученні з графу топології мережі суміжної з коренем дерева вершини, через яку проходить перевантажений найкоротший шлях. Не будь-яку вершину можливо видалити з мережі без порушення її роботи. Вершина не може бути видалена:

* якщо вона є кінцевою в гілці (існує тільки один зв’язок);
* якщо через вершину пролягає єдиний можливий шлях до іншої вершини.

Суть модифікованого алгоритму можна описати за допомогою таких кроків.

1. Знаходження дерева найкоротших шляхів.

1.1.  Задається топологія мережі у вигляді зваженої матриці суміжностей, що зберігається у масиві A[count][count], де count – загальна кількість вершин в мережі.

1.2.  Створюється масив X[count], в якому буде відображатися факт знаходження найкоротшого шляху з вершини s (кореня дерева) до кожної з решти вершин (1 – шлях знайдено, 0 – шлях не знайдено).

1.3.  За алгоритмом Дейкстри знаходиться дерево найкоротших шляхів D з вершини s до всіх решти вершин.

2. Формування основної маршрутної таблиці.

3. Знаходження альтернативних дерев найкоротших шляхів, що рекомендуються до використання при перевантаженні, та формування маршрутних таблиць відповідно до модифікованого алгоритму.

Для всіх вершин, суміжних з коренем дерева:

3.1. Створюється масив B[count][count], що є копією масиву A[count][count] і буде використовуватись для опису топології мережі з видаленою вершиною;

3.2. В циклі по j, де j∈1..count, виконуємо прохід по елементах матриці B[count][count] і занулюємо ваги B[j][n] та B[n][j], які показують вартість переходу між перевантаженою вершиною n та іншими вершинами;

3.3 За алгоритмом Дейкстри знаходиться дерево найкоротших шляхів Da для мережі з визначеною вершиною n;

3.4 Формується масив Xn[count], що відображає знайдені найкоротші шляхи з вершини s до кожної з решти вершин;

3.5. Масив Xn[count] порівнюється зі створеним на кроці 1 масивом Xn[count], в результаті порівняння знаходяться елементи що відрізняються (вершини, для яких відсутній альтернативний шлях);

3.6. Для вершин знайдених на кроці 3.5, дерево альтернативних шляхів Da знайдене на кроці 3.3, доповнюється шляхами з дерева D, знайденого на кроці 1.3, в результаті чого отримуємо дерево найкротших шляхів модифікованого алгоритму Dм.

3.7. Формування маршрутної таблиці, знайдених на основі модифікованого алгоритму Dм.

13→1 : [13,"7","10","28","5","2","4",1] Minimal weight:→34

13→2 : [13,"7","10","28","5",2] Minimal weight:→27

13→3 : [13,"7","10","28","5",3] Minimal weight:→26

13→4 : [13,"7","10","28","5","2",4] Minimal weight:→30

13→5 : [13,"7","10","28",5] Minimal weight:→24

13→6 : [13,"7","10","28","5","3",6] Minimal weight:→30

13→7 : [13,7] Minimal weight:→3

13→8 : [13,"7","10",8] Minimal weight:→8

13→9 : [13,"7","10","8",9] Minimal weight:→10

13→10 : [13,"7",10] Minimal weight:→7

13→11 : [13,"7","10","8",11] Minimal weight:→9

13→12 : [13,"7","10","8","9",12] Minimal weight:→12

13→14 : [13,"7","10","28","18",14] Minimal weight:→17

13→15 : [13,"7","10","28","18",15] Minimal weight:→18

13→16 : [13,"7","10","28","18","20",16] Minimal weight:→21

13→17 : [13,"7","10","28","18","20",17] Minimal weight:→19

13→18 : [13,"7","10","28",18] Minimal weight:→15

13→19 : [13,"7","10","28","18","14",19] Minimal weight:→22

13→20 : [13,"7","10","28","18",20] Minimal weight:→16

13→21 : [13,"7","10","28","25",21] Minimal weight:→42

13→22 : [13,"7","10","28","25","21",22] Minimal weight:→44

13→23 : [13,"7","10","28","25","21","22",23] Minimal weight:→47

13→24 : [13,"7","10","28","25","21","26",24] Minimal weight:→46

13→25 : [13,"7","10","28",25] Minimal weight:→35

13→26 : [13,"7","10","28","25","21",26] Minimal weight:→44

# ОПИС РОЗРОБЛЕННОЇ ПРОГРАМИ

Розроблена програма представляє мережу передачі даних, яка складається з вузлів та каналів зв’язку. Кожен вузол мережі генерує повідомлення. В вузлі-джерелі повідомлення розбивається на пакети та відправляється в мережу. В вузлі-приймачі пакети збираються в повідомлення.

Програма реалізує наступні функції:

1. Створення мережі враховуючи відповідну кількість комунікаційних вузлів і ступінь (середню кількість каналів, що підключені до комунікаційного вузла). Кожний канал МПД має сукупність параметрів: вага лінії, тип (дуплексний, напівдуплексний) та ймовірність виникнення помилок.
2. Визначення найкоротших шляхів між одним з вузлів мережі та всіма іншими вузлами мережі використовуючи алгоритм локальної адаптивної маршрутизації.
3. Визначення маршрутів з мінімальною кількістю транзитних ділянок.
4. Моделювання роботи мережі в двох режимах.

Розробленна програма має інтерфейс, представленний на рис. 4.1.

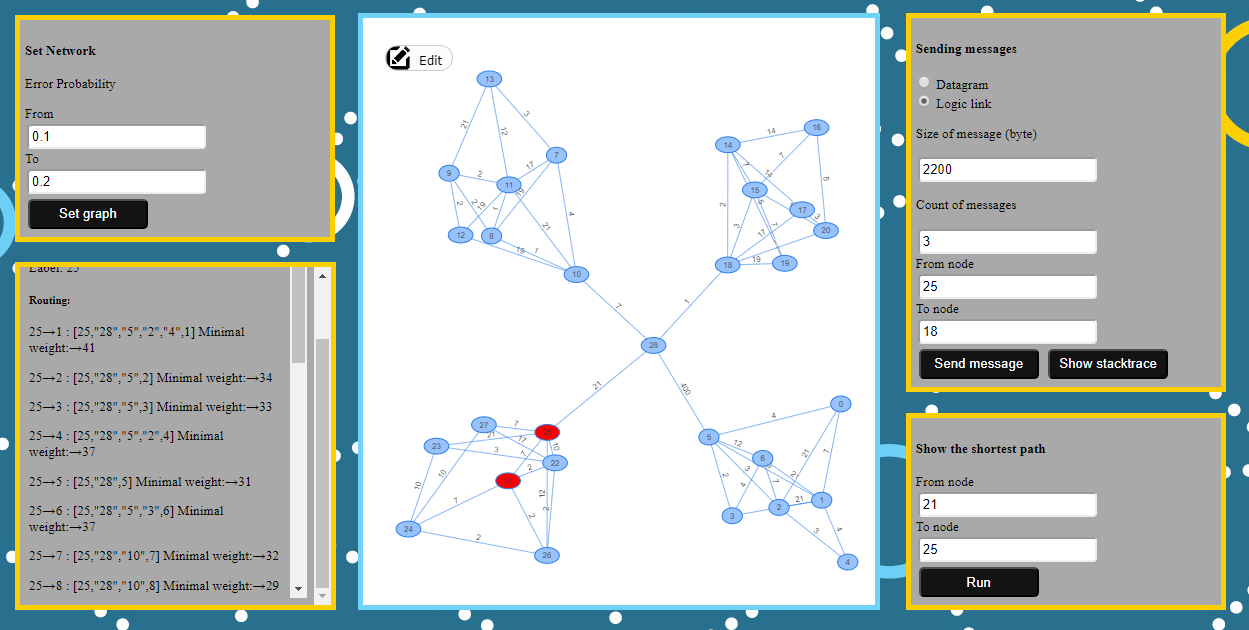


Рис. 4.1 Інтерфейс програми

На головному екрані міститься:

1. Блок для задання можливості видалення помилки і утворення графу мережі (рис. 4.2).

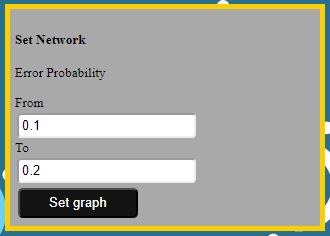


Рис. 4.2. Задання можливості видалення помилки і утворення графу мережі

2. Блок для знаходження найкоротшого маршруту між заданими вузлами в мережі (рис. 4.3).

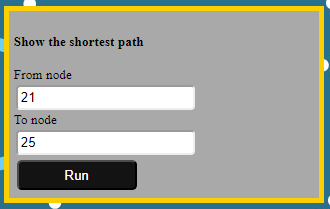


Рис. 4.3. Знаходження найкоротшого маршруту між заданими вузлами в мережі

3. Блок для вибору режиму передачі повідомлення, задання його розміру та маршрут передачі та відображення передачі повідомлення (рис. 4.4).

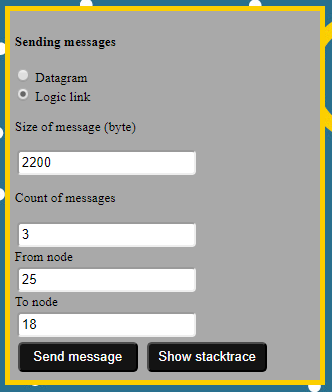


Рис. 4.4. Вибір режиму передачі повідомлення, задання його розміру та маршрут передачі та відображення передачі повідомлення

4. Блок для відображення передачі повідомлення (рис. 4.5).

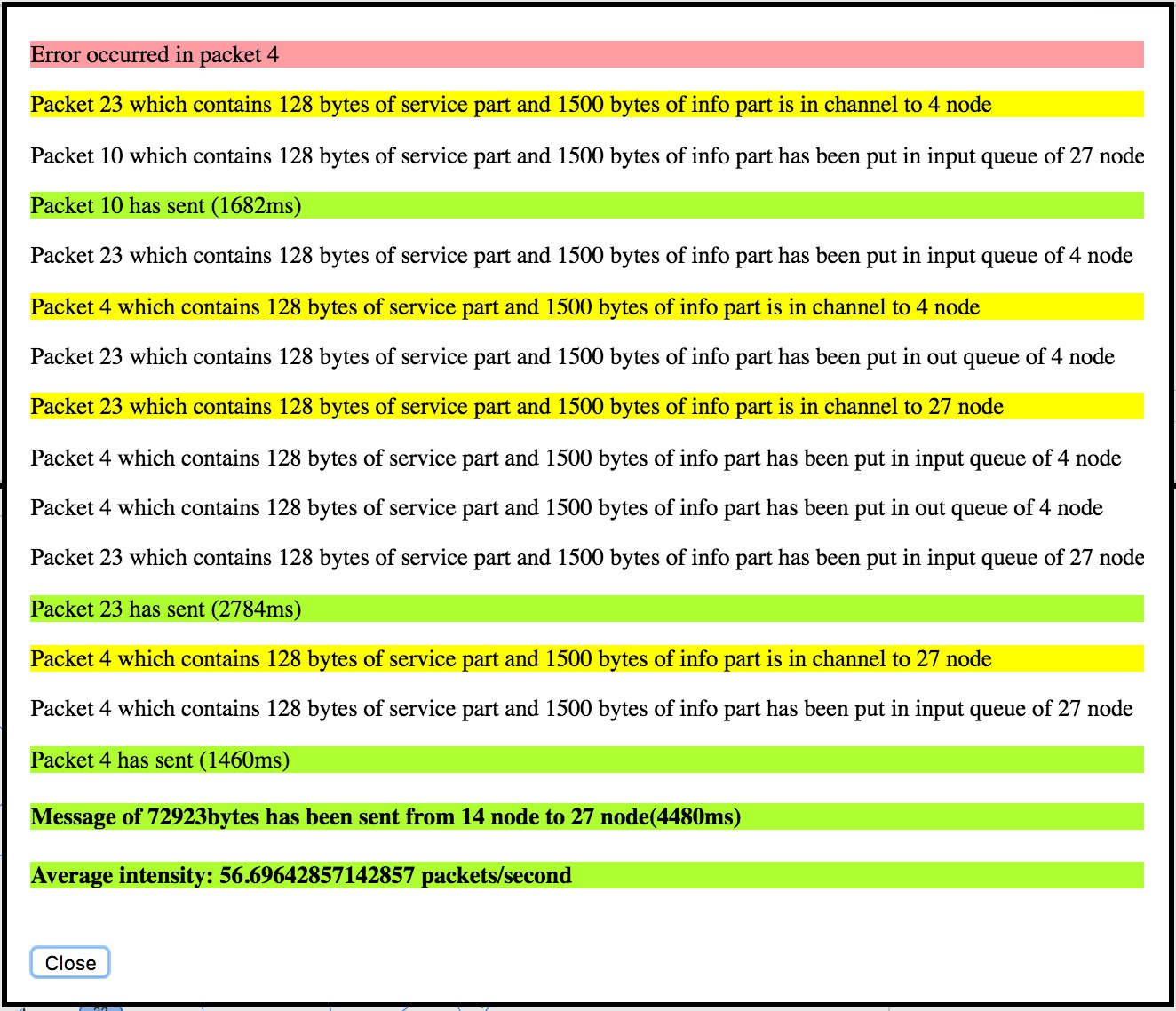


Рис. 4.5. Відображення передачі повідомлення(після передачі)

При натисканні на вузол інтерфейс відображає блоки для додавання вузла, додавання зв’язку, можливості зміни характеристик вузла, а також його видалення (рис 4.6).

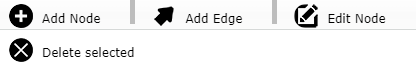


Рис. 4.6. Відображення блоку роботи з вузлами

При натисканні на зв’язок інтерфейс відображає блоки для додавання вузла, додавання зв’язку, можливості зміни характеристик зв’язку, а також його видалення (рис 4.7).

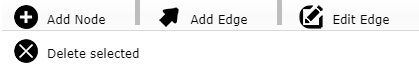


Рис. 4.7. Відображення блоку роботи з зв’язками

При натисканні на кнопку “Add node” потрібно натиснути на екран, щоб утворився новий вузол і задати його параметри (рис 4.8).

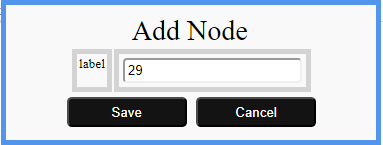


Рис. 4.8. Опції вузла

При натисканні на кнопку “Add edge” потрібно натиснути на один вузол і перетягнути курсор на інший щоб між ними утворився зв’язок, а потім задати його параметри, такі як дуплексний канал чи напівдуплексний, увімкнутий чи вимкнутий, імовірність помилки при передачі повідомлення через нього, а також його вагу (рис 4.9).

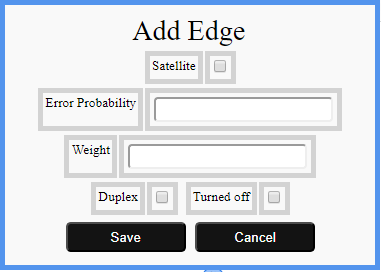


Рис. 4.9. Опції зв’язку

При натисканні на кнопку “Edit node” потрібно натиснути на вузол. Після натискання появиться вікно як на рис. 4.10.

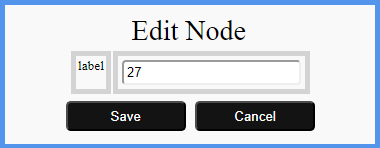


Рис. 4.10. Зміна параметрів вузла

При натисканні на кнопку “Edit edge” потрібно натиснути на зв’язок. Після натискання появиться вікно як на рис. 4.11.

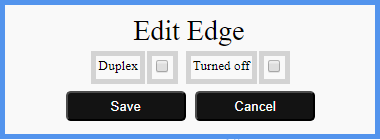


Рис. 4.11. Зміна параметрів зв’язку

При натисканні кнопки Set graph програма створює 4 регіональні мережі, які пов’язані між собою одним вузлом та 4 каналами, як на рис. 4.12.

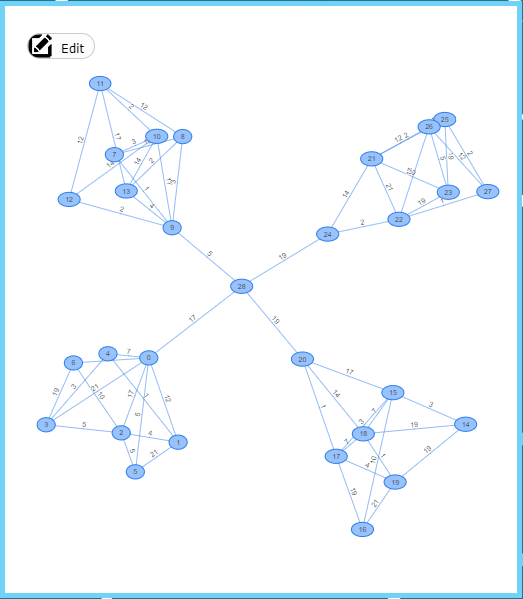


Рис. 4.12. Створення мережі

Крім того при натисканні на вузол відображається додаткове вікно з інформацією про вузол, а також його таблицю маршрутизації (рис. 4.13).

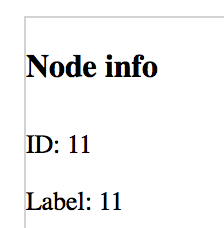


Рис. 4.13. Іноформація про вузол

Приклад таблиці маршрутизації відображений в інтерфейсі програми для 24 вузла.

22 --> 1 : [22,"8","10","14","6",1] min weight: 24

22 --> 2 : [22,"8","10","0",2] min weight: 17

22 --> 3 : [22,"8",3] min weight: 7

22 --> 4 : [22,"8","10","0","2",4] min weight: 19

22 --> 5 : [22,"8","10","0","2",5] min weight: 19

22 --> 6 : [22,"8","10","14",6] min weight: 20

22 --> 7 : [22,"8","32",7] min weight: 18

22 --> 8 : [22,8] min weight: 2

22 --> 9 : [22,"8","3",9] min weight: 11

22 --> 10 : [22,"8",10] min weight: 4

22 --> 11 : [22,"8","10","14","6","1",11] min weight: 28

22 --> 12 : [22,"8","10","0","2","5",12] min weight: 23

22 --> 13 : [22,"8","3","15","17","28",13] min weight: 29

22 --> 14 : [22,"8","10",14] min weight: 15

22 --> 15 : [22,"8","3",15] min weight: 12

22 --> 16 : [22,"8","32","7",16] min weight: 23

22 --> 17 : [22,"8","3","15",17] min weight: 16

22 --> 18 : [22,"8","10","0","2","4",18] min weight: 28

22 --> 19 : [22,"8","3","9",19] min weight: 25

22 --> 20 : [22,"8","10","14","6","1",20] min weight: 29

22 --> 21 : [22,"8","10","0",21] min weight: 22

22 --> 23 : [22,"8","32",23] min weight: 25

22 --> 24 : [22,"8","10","0","2","4",24] min weight: 33

22 --> 25 : [22,"8","10","14","6",25] min weight: 24

22 --> 26 : [22,"8","32","7",26] min weight: 23

22 --> 27 : [22,"8","10","14","6",27] min weight: 35

22 --> 28 : [22,"8","3","15","17",28] min weight: 20

22 --> 29 : [22,"8","10","0","2",29] min weight: 28

22 --> 30 : [22,"8","3","15",30] min weight: 26

22 --> 31 : [22,"8","3",31] min weight: 19

22 --> 32 : [22,"8",32] min weight: 13

22 --> 33 : [22,"8","10","14","6","1",33] min weight: 26

Результат передачі повідомлення в режимі віртуального з’єднання зобрвжений на рис. 4.14, а в дейтаграмному на рис. 4.15:

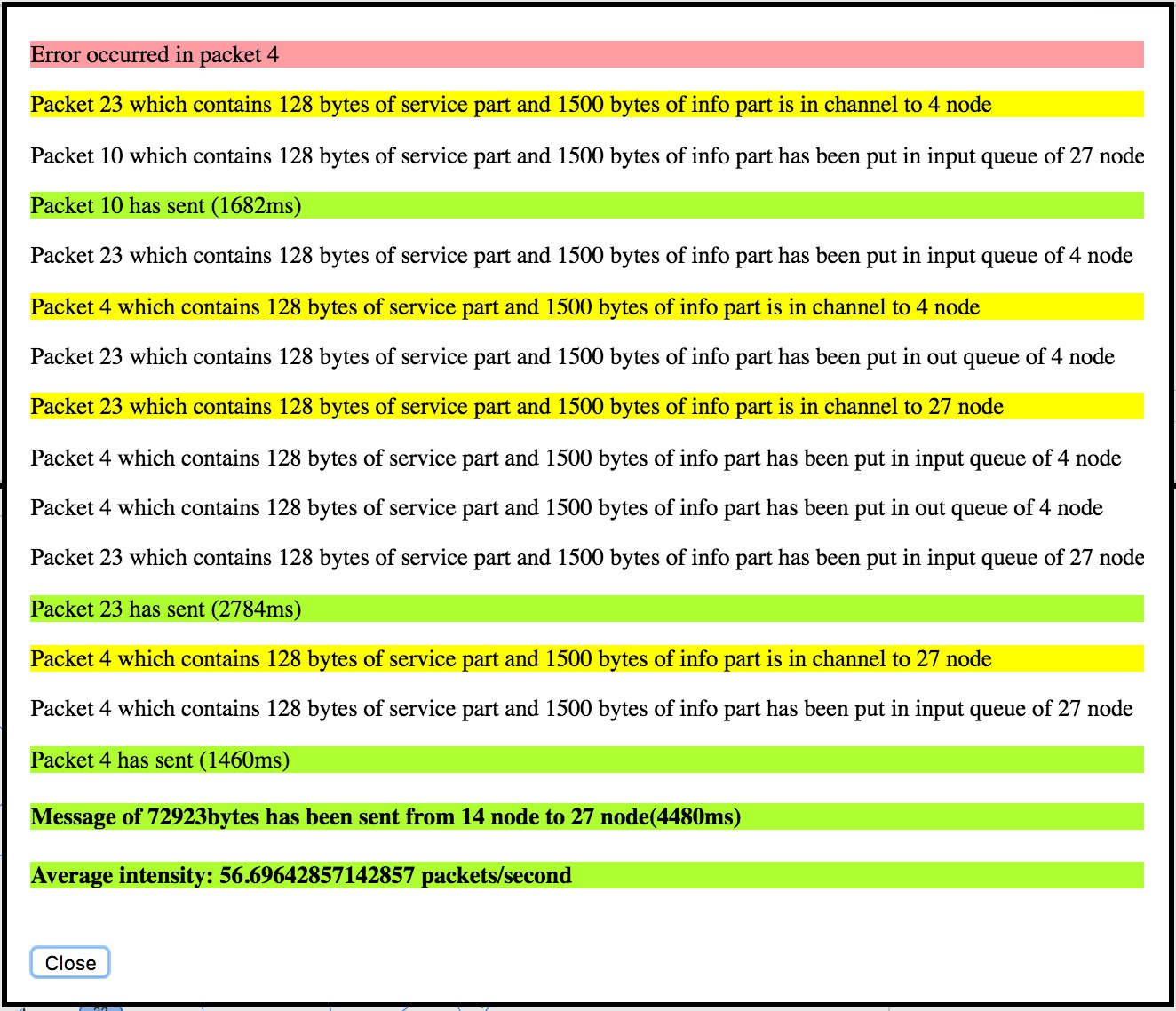


Рис. 4.14. Передача повідомлення в режимі віртуального з’єднання



Рис. 4.15. Передача повідомлення в дейтаграмному режимі

В залежності від режиму та розміру повідомлення були отримані різні дані, які занесені у таблиці 4.1, 4.2 та 4.3. Тестування виконувалось з генерацією повідомлення безпосередньо від відправника до отримувача, які розташовані у різних регіональних мережах.

В таблиці 4.1 представлена статистика передачі двох повідомлень по заданому маршруту в дейтаграмному режимі та режимі віртуального

з’єднання при передачі 32768 байт інформації.

Таблиця 4.1

Результати передачі повідомлення розміром 32768 байт в режимі віртуального з’єднання та дейтаграмному режимі

|  |  |  |
| --- | --- | --- |
| **Параметри** | **Режим віртуального з’єднання** | **Дейтаграмний** |
| Стартовий вузол | 30 | 30 |
| Кінцевий вузол | 32 | 32 |
| Час, мс | 2345 | 521 |
| Кількість службових пакетів | 44 | 22 |
| Розмір інформ частини повідомлення | 32768 | 32768 |
| Tрафік службової частини повідомлення | 5632 | 2816 |
| Імовірність помилки | 0.1 - 0.2 | 0.1 - 0.2 |
| Середня інтенсивність відправлення пакетів, пакетів / с | 46.48 | 184.26 |

Передамо повідомлення розміром 65535 байт. Отримані дані представлені у таблиці 4.2.

Таблиця 4.2

Результати передачі повідомлення розміром 65535 байт в режимі віртуального з’єднання та дейтаграмному режимі

|  |  |  |
| --- | --- | --- |
| **Параметри** | **Режим віртуального з’єднання** | **Дейтаграмний** |
| Стартовий вузол | 11 | 11 |
| Кінцевий вузол | 12 | 12 |
| Час, мс | 5247 | 794 |
| Кількість службових пакетів | 88 | 44 |
| Розмір інформ частини повідомлення | 65535 | 65535 |
| Tрафік службової частини повідомлення | 11264 | 5632 |
| Імовірність помилки | 0.1 - 0.2 | 0.1 - 0.2 |
| Середня інтенсивність відправлення пакетів,  пакетів / с | 55.65 | 243.07 |

Збільшимо розмір повідомлення, що передається до 131072 байт. Дані представлені в таблиці 4.3.

Таблиця 4.3

Результати передачі повідомлення розміром, 131072 байт в режимі віртуального з’єднання та дейтаграмному режимі

|  |  |  |
| --- | --- | --- |
| **Параметри** | **Режим віртуального з’єднання** | **Дейтаграмний** |
| Стартовий вузол | 4 | 4 |
| Кінцевий вузол | 9 | 9 |
| Час, мс | 8946 | 983 |
| Кількість службових пакетів | 176 | 88 |
| Розмір інформ частини повідомлення | 131072 | 131072 |
| Tрафік службової частини повідомлення | 22528 | 11264 |
| Імовірність помилки | 0.1 - 0.2 | 0.1 - 0.2 |
| Середня інтенсивність відправлення пакетів,  пакетів / с | 35.99 | 306.20 |

Дивлячись на отримані результати маршрутизаціі, можна побачити, що при передачі у дейтаграмному режимі час доставки повідомлення значно менший, ніж при передачі у режимі віртуального з’єднання. Це досягається завдяки тому, що в дейтаграмному режимі частини повідомлення передаються паралельними шляхами, також це зумовлено простотою організації алгоритму дейтаграмної передачі, відсутністю конфігурації каналу та підтверджень з’єднання та роз’єднання у ній.

Також на час передачі повідомлення впливає імовірність виникнення помилки в каналі. Чим більшою вона буде тим більшим буде час передачі повідомлення.

ВИСНОВКИ

Під час виконання курсового проекту, я ознайомився з роботою алгоритмів маршрутизації і принципом роботи мережевого рівня мережі. Кінцевим продуктом стала програма, яка моделює процес передачі інформації в МПД.

Змоделювавши передачу даних, можна проаналізувати ефективність МПД застосовуючи різні режими. Більш ефективним виявився дейтаграмний режим, оскільки завдяки специфіці алгоритму інформація передаватиметься швидше.

Протестувавши режим логічного з’єднання і дейтаграмний режим, виявилось що для передачі інформації в дейтаграмному режимі часу необхідно менше ніж у логічному, що являється перевагою. Також у цьому режимі не монополізується канал звязку і мережа використовується більш продуктивно. З іншого боку, дейтаграмний режим не гарантує того, що доставка повідомлення буде виконана коректно, оскільки деякі пакети можуть “загубитись’’ в мережі. Повідомлення в режимі логічного з’єднання передаються повільніше, але в цьому режимі гарантується передача всіх пакеті. Ще одним недоліком цього режиму є більша кількість службової інформації. Зменшення часу передачі повідомлень також зумовлена забиванням черг, адже всі пакети передаються лише по одному найкоротшому шляху.

СПИСОК ЛІТЕРАТУРИ

1. Блек Ю. Мережі ЕОМ. Протоколи стандарти інтерфейси: Пер. з англ. / Блек Ю. – М.: Мир, - 1990.
2. Комп’ютерні мережі: навчальний посібник / Азаров О.Д., Захарченко С.М., Кадук О.В., Орлова М.М., Тарасенко В.П.. – Вінниця: ВНТУ. – 2013. – 371 с.

3 Семенов Ю.А. Протоколы и ресурсы Internet / Семенов Ю.А.: М. – 1996.

4 Столлінгс В. Комп’ютерні системи передачі даних с. 6-те видання 928 с.