

КВІТКО О.С.,
ДОРОШЕНКО К.С.,
ПОЛТОРАК В.П.

ДОСЛІДЖЕННЯ МЕТОДІВ ДИНАМІЧНОГО ПРОГРАМУВАННЯ У КОРПОРАТИВНИХ МЕРЕЖАХ

Розглянуто можливості використання методів динамічного програмування в мережах TCP/IP. Наведено результати експериментальних досліджень використання алгоритму динамічного програмування для вирішення задачі пошуку оптимального шляху (маршруту) в корпоративній мережі.

The article describes the usage of dynamic programming methods in TCP/IP networks. There are results of experimental using a dynamic programming algorithm in solving the optimal path (route) search on the corporate network.

Постановка задачі в загальному вигляді та її актуальність

Сучасні корпоративні мережі, побудовані на IP-протоколах, використовуються не тільки для класичної передачі даних, але й для обміну звуковими даними, проведення мультимедійних конференцій, оперативного управління, прослуховування музичних записів, перегляду відео кліпів, мережових ігор та інших прикладних програм реального часу. Тому все актуальнішою стає проблема пошуку оптимального шляху в процесі маршрутизації, та мінімізація ресурсів, необхідних для пошуку оптимального шляху. Дивлячись із цієї точки зору, дослідження методу моделювання процесу маршрутизації на основі методу динамічного програмування заслугоує на прискіпливішу увагу.

Мета дослідження

Довести доцільність використання методу динамічного програмування для моделювання процесу пошуку оптимального шляху в мережі.

Викладення основного матеріалу досліджень

Як відомо існує кілька параметрів оцінювання протоколів маршрутизації, серед них масштабування, швидкість обміну інформацією та ресурсоємність. Існує також параметр для оцінки самого алгоритму, який лежить в основі протоколу динамічної маршрутизації, що має назву O-нотація. Асимптотична нотація великого O, відома також як нотація Ландау, – розповсюджена математична нотація для формального запису асимп-

тотичної поведінки функцій. Вона широко вживається в теорії складності обчислень, інформатиці та математиці.

Теорія складності обчислень – підрозділ теоретичної інформатики, який займається дослідженням складності алгоритмів для розв'язання задач на основі формально визначених моделей обчислювальних пристроїв. Складність алгоритмів вимірюється за необхідними ресурсами, в основному, це тривалість обчислень або необхідний об'єм пам'яті. В окремих випадках досліджуються інші міри складності, такі як розмір мікропроцесора, або кількість процесорів, необхідна для роботи паралельних алгоритмів.

Основна задача досліджень в теорії складності обчислень полягає у класифікації всіх розв'язуваних задач. Зокрема, робляться спроби відокремити множину задач з ефективними алгоритмами розв'язання від множини важко розв'язуваних задач.

Нехай A – алгоритм розв'язання деякого класу задач, а N – розмірність окремої задачі цього класу. N може бути, наприклад, розмірністю оброблюваного масиву, числом вершин оброблюваного графа тощо. Позначимо $f_A(N)$ функцію, що дає верхню межу максимального числа основних операцій (додавання, множення і т. д.), які повинен виконати алгоритм A , вирішуючи задачу розмірності N . Говоритимемо, що алгоритм A поліноміальний, якщо $f_A(N)$ зростає не швидше, ніж деякий поліном від N . В іншому разі A – експоненціальний алгоритм.

Виявляється, що між цими класами алгоритмів є істотна різниця: при великих розмі-

рностях задач (які, зазвичай, найцікавіші на практиці), поліноміальні алгоритми можуть бути виконані на сучасних комп'ютерах, тоді як експоненціальні алгоритми для практичних розмірностей задач, як правило, не можуть виконуватися повністю. Зазвичай розв'язання задач, що породжують експоненціальні алгоритми, пов'язано з повним перебором всіх можливих варіантів, і зважаючи на практичну неможливість реалізації такої стратегії, для їх розв'язання розробляються інші підходи.

Наприклад, якщо навіть існує експоненціальний алгоритм для знаходження оптимального розв'язку деякої задачі, то на практиці застосовуються інші, ефективніші поліноміальні алгоритми для знаходження не обов'язково оптимального, а лише прийнятного розв'язку (наближеного до оптимального).

Поліноміальні алгоритми також можуть істотно розрізнятися залежно від ступеня полінома, що апроксимує залежність $f_A(N)$. Розглянемо оцінювання часової складності алгоритму. Така оцінка проводиться із застосуванням згаданої вище O – нотації. Будемо говорити, що $f_A(N)$ зростає як $g(N)$ для великих N і записувати це $f_A(N) = O(g(N))$, якщо існує така додатна константа $a > 0$, що $\lim_{N \rightarrow \infty} f_A(N) / g(N) = a$. Тоді оцінка $O(g(N))$ називається асимптотичною часовою складністю алгоритму.

Оцінка $O(g(N))$ для функції $f_A(N)$ застосовується, коли точне значення $f_A(N)$ невідоме, а відомий лише порядок зростання часу, затрачуваного на розв'язання задачі розмірністю N за допомогою алгоритму A . Точні значення $f_A(N)$ залежать від конкретної реалізації, тоді як $O(g(N))$ є характеристикою самого алгоритму.

В наступній таблиці наведено поширені асимптотичні складності, типові для задач управління мережами.

Складність	Коментар	Приклади
$O(1)$	Сталій час роботи не залежно від розміру задачі	Очікуваний час пошуку в хеші

$O(\log n)$	Логарифмічне зростання – подвоєння розміру задачі збільшує час роботи на сталу величину	Обчислення x^n ; двійковий пошук в масиві з n елементів
$O(n^2)$	Квадратичне зростання – подвоєння розміру задачі вчетверо збільшує необхідний час	Елементарні алгоритми сортування
$O(n^3)$	Кубічне зростання – подвоєння розміру задачі збільшує необхідний час у вісім разів	Звичайне множення матриць
$O(cn)$	Експоненціальне зростання – збільшення розміру задачі на 1 призводить до c -кратного збільшення необхідного часу; подвоєння розміру задачі підносить необхідний час у квадрат	Деякі задачі комівояжера, алгоритми пошуку повним перебором

Для алгоритму Беллмана-Форда складність можна записати так $O(m \cdot n)$. Фактично це є часова складність алгоритму, що означає, що за час $O(m \cdot n)$ алгоритм знаходить найкоротші маршрути від однієї вершини до решти вершин. Щодо алгоритму Дейкстри, то його складність значною мірою залежить від способу реалізації цього алгоритму. В найпростішому випадку, коли для пошуку вершин переглядається вся підмножина вершин, а для збереження величин використовується масив, час роботи алгоритму буде $O(n^2 + m)$.

Щоб визначити функцію складності досліджуваного нами алгоритму динамічного програмування для початку необхідно проаналізувати цей алгоритм. Основною частиною роботи даного алгоритму є процедура розширення найкоротших шляхів, які відомі на даний момент, шляхом додавання в них по ще одному ребру. Вибір ребра, що додається, здійснюється на основі доведення, що всі часткові маршрути найкоротшого маршруту також вважаються найкоротшими маршрутами. Таким чином, для матриці розмірністю $n \cdot n$ мінімальна вага будь-якого шляху з вершини i в вершину j , що включає не більше m ребер визначається так:

$$l_{ij}^{(m)} = \min_{1 \leq k \leq n} \left\{ l_{ik}^{(m-1)} + w_{kj} \right\},$$

де k – проміжна вершина, що входить в оптимальний маршрут, w_{kj} – вага ребра, що веде від вершини k до кінцевої вершини j .

Оскільки алгоритм включає три вкладені цикли, доцільно говорити, що час його роботи $O(n^3)$. Після аналізу графіку функції (рис. 1), що відповідає даному порядку складності, було прийнято рішення ввести в формулу складності деякий коефіцієнт q , що показуватиме зміну кількості ітерацій даного алгоритму в залежності від кількості ребер у графі.

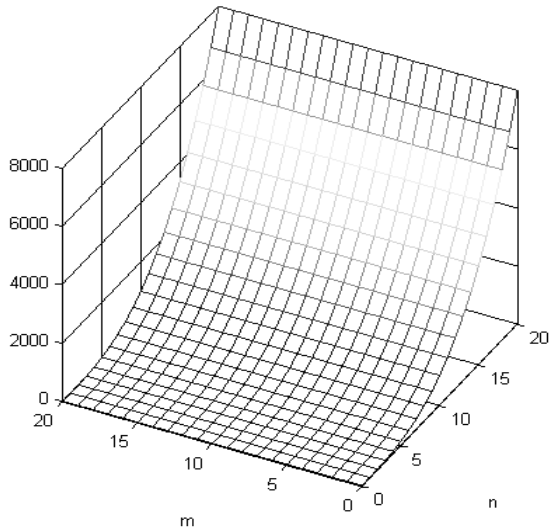


Рис.1 Графік функції $O(n^3)$.

Виразимо цей коефіцієнт відношенням $q = \frac{n}{m}$, тоді час роботи алгоритму $O(qn^3)$.

Після аналізу графіка функції, що показує складність алгоритму з урахуванням коефіцієнта q (рис. 2), можна помітити, що даний коефіцієнт може суттєво впливати на показник складності алгоритму. Якщо $0 < q < 1$ очевидно, що показник складності алгоритму може істотно зменшуватись при $q \rightarrow 0$.

Відношення $\frac{n}{m}$ є показником, що характеризує ступінь заповненості матриці сполучень, інакше кажучи, дане відношення характеризує фізичну структуру досліджуваної мережі. Наприклад, якщо у мережі n вузлів, то мінімальна кількість ребер, необхідних щоб з'єднати ці вузли так, щоб до кожного вузла призначення було два маршрути, становить $m=n$.

В такому випадку значення $q = \frac{n}{m} = \frac{n}{n} = 1$.

А максимальна кількість ребер, що дозволить організувати повністю зв'язану топологію (full-mesh topology), становить $m = \frac{n(n-1)}{2}$, що приведе до того, що відношення матиме вигляд $q = \frac{n}{m} = \frac{2n}{n(n-1)} = \frac{2}{n-1}$.

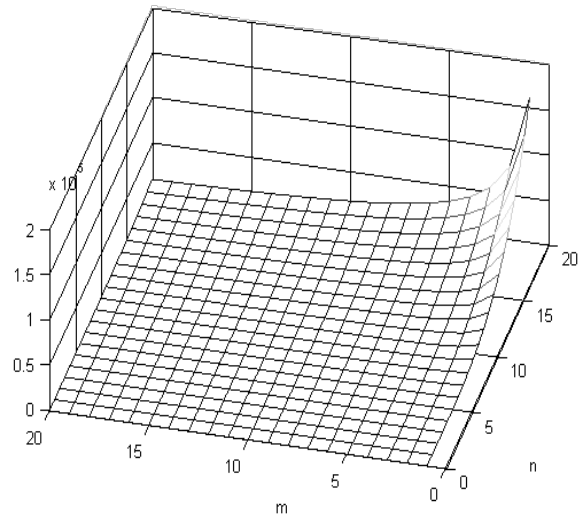


Рис.2 Графік функції $O(qn^3)$.

Керуючись тим фактом, що в реальному житті побудова повністю зв'язаної топології мережі надто дороге рішення, будемо говорити, що $m < \frac{n(n-1)}{2}$. А також тим, що при $q=1$ складність даного алгоритму досягатиме значення $O(n^3)$. А отже визначимо діапазон значень коефіцієнта q , при яких алгоритм буде мати найменшу складність – $\frac{2}{n-1} < q < 1$.

Оскільки в дослідженні ми прив'язуємось саме до даної реалізації алгоритму, щоб оцінити його здатність адекватно моделювати процес маршрутизації, тобто вирішувати задачу пошуку оптимального маршруту, необхідно оцінити його складності з практичної точки зору. Для цього було проведено ряд досліджень аналітичного характеру, по результатам яких ми отримали дані, що унаочнюють залежність часу роботи алгоритму від топології досліджуваної мережі (рис.3).

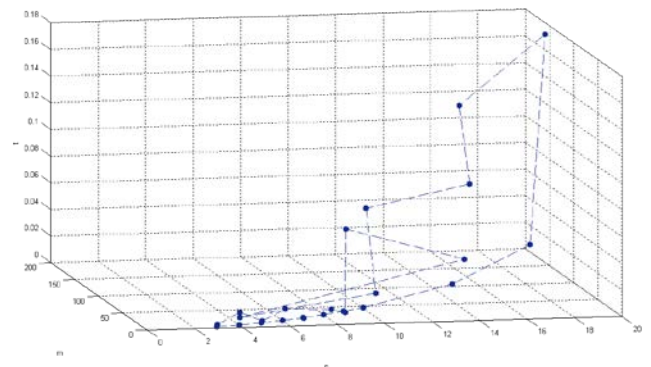


Рис.3 Графік залежності часу виконання від структури мережі

Аналізуючи результати дослідження можна прослідкувати деяку схожість з теоретичними даними. Завдяки проведеним розра-

хункам і побудові графіка, що зображений вище, ми довели, що час роботи програми залежить не лише від кількості вузлів у топології, але й від кількості ребер, що з'єднують ці вузли.

Огляд існуючих рішень

Метод динамічного програмування широко відомий і потужний математичний метод сучасної теорії управління. Динамічне програмування дозволяє вирішувати задачі, розбиваючи їх на підзадачі, аналогічні початковій задачі, і об'єднуючи в подальшому рішення цих підзадач. Тобто оптимальне рішення підзадач меншого розміру може бути використано для рішення початкової задачі. Ключовою умовою для застосування динамічного програмування є наявність підзадач, що перекриваються.

Для кожного кінцевого користувача важливо, щоб мережа виконувала функції важливі саме для нього. Тому існує велика кількість критеріїв оцінювання роботи мережі передачі даних. Для проведення моделювання необхідно виконати вибір критерію якості для оцінювання роботи алгоритму. При виборі критерію оцінювання роботи мережі передачі даних перевага була надана показнику, що з нашої точки зору найбільш точно покаже швидкість та надійність функціонування досліджуваного алгоритму. В даному дослідженні критерієм оцінювання функціонування мережі обрано час відгуку кінцевого вузла на запит вузла-відправника. Для цього було використано загально відому утиліту Ping, що використовується для перевірки з'єднань в мережах на основі TCP/IP. Якщо не вказано інакше, Ping пересилає пакети встановленого розміру, як правило, не великі у 32 байта. Утиліта відправляє запити (ICMP Echo-Request) протоколу ICMP [2] вказаному вузлу мережі і фіксує отримані відповіді (ICMP Echo-Reply). Час між відправкою запиту і отриманням відповіді (RTT, від англ. Round Trip Time) дозволяє визначити як двосторонні затримки (RTT) по маршруту, так і частоту, з якою пакети втрачаються.

Таким чином метою, що була поставлена в процесі дослідження, є моделювання ситуації за якої в мережі відбувається мінімізація критерію оптимізації, а саме часу, що витрачається на передачу вказаного пакету з пункту А в пункт В.

За основу даного дослідження було взято статтю [3], в якій було представлено метод моделювання процесу маршрутизації на ос-

нові методу динамічного програмування. Власне там же можна знайти алгоритм функціонування програми та результати досліджень автора. Для реалізації цього алгоритму було використано засоби середовища MATLAB, в тому числі мова програмування MATLAB. В даній праці було побудовано математичну модель мережі, із наперед визначеною конфігурацією, процес маршрутизації в ній було реалізовано на основі алгоритму динамічного програмування. Моделювання процесу маршрутизації виконано засобами програмного середовища MATLAB 7.0. В даній роботі було використано вже існуючу програмну реалізацію даного алгоритму, проте адаптовано його під роботу з іншими вхідними даними.

Дослід №1

Моделювання телекомунікаційної мережі, в нашому випадку, відбувалося на рівні Network моделі OSI, та, на відміну від моделі розглянутої у статті [3], в процесі даного дослідження нами було побудовано модель телекомунікаційної мережі, яка складалась з десяти вузлів (маршрутизаторів), які з'єднані між собою каналами різної пропускності (bandwidth).

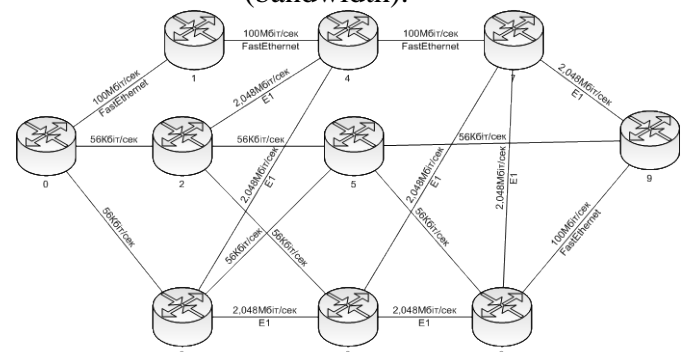


Рис. 4. Схема мережі

Результати роботи методу динамічного програмування порівнювалися із традиційними методами такими, як протоколи маршрутизації OSPF, RIP та їх комбінації.

В залежності від технології Канального рівня (Network Access layer) моделі TCP/IP значення пропускності (bandwidth) міняється. В таблиці 1. представлено значення пропускності (bandwidth) в залежності від різних технологій Канального рівня, що використовувались при розробці даної моделі.

Таблиця 1.

Найменування технології	Формат	Пропускність (Bandwidth)
100BaseT	Ethernet кадр	100 Мб/сек
PPP_E1	ip3 дейтаграма	2.048 Мб/сек

PPP_56K	ipv4 дейтаграма, ipx пакет	56Кб/сек
---------	----------------------------	----------

Після перетворення величин пропускних здатностей у ціни каналів схема набуде наступного вигляду:

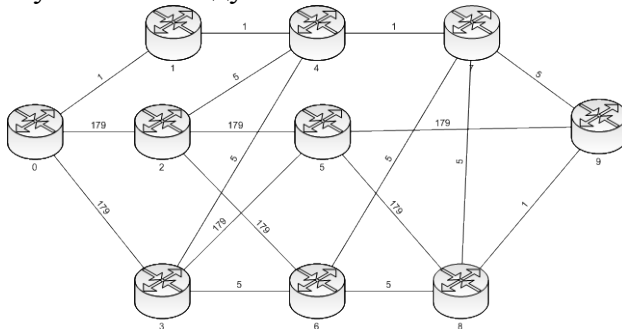


Рис. 5. Схема мережі, представлена у вигляді графа

Для розв'язання задачі маршрутизації інформаційного пакету з вузла 0 до вузла 9, потрібно описати отриманий граф матрицею сполучень:

$$C = \begin{pmatrix} 999 & 1 & 179 & 179 & 999 & 999 & 999 & 999 & 999 & 999 \\ 1 & 999 & 999 & 999 & 1 & 999 & 999 & 999 & 999 & 999 \\ 179 & 999 & 999 & 999 & 5 & 179 & 179 & 999 & 999 & 999 \\ 179 & 999 & 999 & 999 & 5 & 179 & 5 & 999 & 999 & 999 \\ 999 & 1 & 5 & 5 & 999 & 999 & 999 & 1 & 999 & 999 \\ 999 & 999 & 179 & 179 & 999 & 999 & 999 & 999 & 179 & 179 \\ 999 & 999 & 179 & 5 & 999 & 999 & 999 & 5 & 5 & 999 \\ 999 & 999 & 999 & 999 & 1 & 999 & 5 & 999 & 5 & 5 \\ 999 & 999 & 999 & 999 & 999 & 179 & 5 & 5 & 999 & 1 \\ 999 & 999 & 999 & 999 & 999 & 179 & 999 & 5 & 1 & 999 \end{pmatrix}$$

де числами 999 позначені неіснуючі сполучення у графі.

Подавши таку матрицю на вхід програми, отримаємо:

result =

CurrentTime: 1

CurrentVertex: 1

NextVertex: 2

Cost: 1

result =

CurrentTime: 2

CurrentVertex: 2

NextVertex: 5

Cost: 1

result =

CurrentTime: 3

CurrentVertex: 5

NextVertex: 8

Cost: 1

result =

CurrentTime: 4

CurrentVertex: 8

NextVertex: 10

Cost: 5

Отриманий результат буде занесений у таблицю маршрутизації (Routing Table) на вузлі 0 і буде застосовуватись для передачі всіх пакетів, адресованих у вузол 9. Аналогічні розрахунки будуть виконані маршрутизатором для всіх інших вузлів призначення, які наявні в його Topology Database (база даних в пам'яті маршрутизатора, що містить інформацією про всі маршрути). Різниця між Topology Database і Routing Table полягає у тому, що перша містить інформацію про всі маршрути з вузла 0 в вузол 9, а друга зберігає тільки оптимальні маршрути з вузла 0 в вузол 9. По результатам роботи програми в середовищі MATLAB 7.0 можна зробити висновки про оптимальний шлях з вузла 0 в вузол 9: 0 – 1 – 4 – 7 – 9.

Для перевірки результату, отриманого за допомогою програми, було використано середовище моделювання телекомунікаційних мереж OPNet, оскільки даний пакет моделювання широко розповсюджений і відомий різноманітністю технологій, що можуть бути промодельовані в даному середовищі [4]. Модель мережі наведена на наступному рисунку:

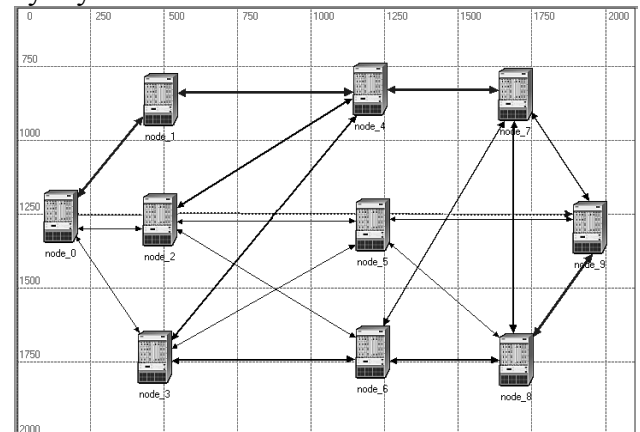


Рис. 6. Модель мережі в пакеті OPNet

Для моделювання вузлів мережі були використані узагальнені блоки ethernet4_slip8_gtwy, які можуть моделювати роботу довільного IP-маршрутизатора. Зв'язки між вузлами мережі промодельовані за допомогою каналів PPP (Point-to-Point Protocol). Для задання ціни каналу, яка враховується при моделюванні процесу прийняття рішення про маршрутизацію пакетів, використовується атрибут Reference Bandwidth. На основі побудованої топології мережі в середовищі OPNet було здійснено настройку існуючих протоколів маршрутизації трьома різними способами. В першому

випадку було настроєно протокол маршрутизації OSPF (Open Shortest Path First), який враховує значення пропусковості кожного з'єднання для пошуку оптимального маршруту. Для знаходження оптимального (у сенсі швидкості передачі даних) шляху між вузлами «node_0» та «node_9», які відповідають вершинам 0 та 9 графа, зображеного на Рис. 2, було задано напрям руху трафіку ICMP (Ping Traffic) від вузла «node_0» до вузла «node_9».

List of traversed IP interfaces:

IP Address	Hop Delay	Node Name
192.0.14.2	0,00000	Enterprise Network.node_0
192.0.13.1	0,00304	Enterprise Network.node_1
192.0.2.1	0,00401	Enterprise Network.node_4
192.0.12.2	0,00285	Enterprise Network.node_7
192.0.18.1	0,00215	Enterprise Network.node_9

Якщо ICMP пакети проходять через ці вузли, очевидно, що шлях node_0 -> node_1 -> node_4 -> node_7 -> node_9 обрано оптимальним для даної мережі на проміжку між вузлом 0 та вузлом 9. А отже результат моделювання в середовищі OPNet підтвердив результати моделювання з допомогою MATLAB 7.0.

Висновок: програма написана по алгоритму динамічного програмування підходить для вирішення задачі маршрутизації в більших мережах, тобто добре масштабується на більші мережі. Крім того, даний алгоритм підходить для вирішення задачі маршрутизації з урахуванням пропусковості лінки як показника якості для вибору оптимального маршруту.

Дослід №2

В другому випадку процес пошуку оптимального шляху в моделі здійснювався по протоколу динамічної маршрутизації RIP (Routing Information Protocol). Даний протокол для своєї роботи використовує досить просту метрику, що вимірюється кількістю проміжних вузлів від відправника до адресата. Як відомо протокол маршрутизації RIP не враховує значення пропусковості (bandwidth) для вибору оптимального маршруту в телекомунікаційній мережі, рішення про оптимальність маршруту приймається на основі метрики, що вимірюється в кількості проміжних вузлів від відправника до одержувача. А отже можна сказати, що показником якості для роботи такої мережі буде відстань, що обчислюється в хопах (кількість кроків до одержувача), і значенням пропусковості

(bandwidth) можна знехтувати. Тому модель такої мережі можна представити у вигляді показаному на рисунку 7.

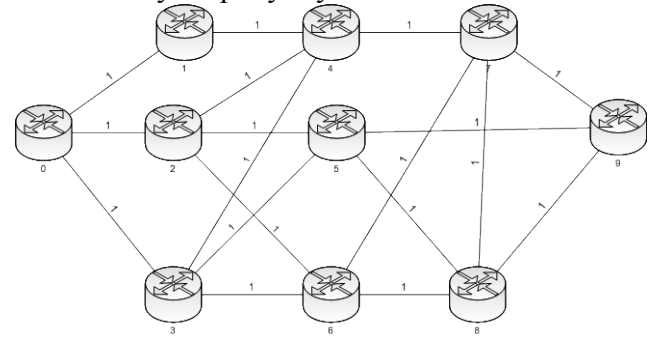


Рис. 7. Схема мережі у вигляді графа з одиничними вагами зв'язків між вузлами

Для розв'язання задачі маршрутизації інформаційного пакету з вузла 0 до вузла 9, потрібно описати отриманий граф матрицею сполучень:

$$C = \begin{pmatrix} 999 & 1 & 1 & 1 & 999 & 999 & 999 & 999 & 999 & 999 \\ 1 & 999 & 999 & 999 & 1 & 999 & 999 & 999 & 999 & 999 \\ 1 & 999 & 999 & 999 & 1 & 1 & 1 & 999 & 999 & 999 \\ 1 & 999 & 999 & 999 & 1 & 1 & 1 & 999 & 999 & 999 \\ 999 & 1 & 1 & 1 & 999 & 999 & 999 & 1 & 999 & 999 \\ 999 & 999 & 1 & 1 & 999 & 999 & 999 & 999 & 1 & 1 \\ 999 & 999 & 1 & 1 & 999 & 999 & 999 & 1 & 1 & 999 \\ 999 & 999 & 999 & 999 & 1 & 999 & 1 & 999 & 1 & 1 \\ 999 & 999 & 999 & 999 & 999 & 1 & 1 & 1 & 999 & 1 \\ 999 & 999 & 999 & 999 & 999 & 1 & 999 & 1 & 1 & 999 \end{pmatrix}$$

де 1 – вузли з'єднані безпосередньо, 999 – безпосередній зв'язок між вузлами відсутній.

Подавши таку матрицю на вхід програми, отримаємо:

```
result =
  CurrentTime: 2
  CurrentVertex: 1
  NextVertex: 3
  Cost: 1
```

```
result =
  CurrentTime: 3
  CurrentVertex: 3
  NextVertex: 6
  Cost: 1
```

```
result =
  CurrentTime: 4
  CurrentVertex: 6
  NextVertex: 10
  Cost: 1
```

Отже оптимальний шлях з вузла 0 в вузол 9, вирахований програмою на основі алгоритму динамічного програмування: 0 -> 2 -> 5 -> 9. Для перевірки цих даних в середовищі OPNet необхідно налаштувати для даної топології протокол маршрутизації RIP (Routing

Information Protocol). В результаті виконаних налаштувань найбільш оптимальним був обраний шлях:

0 -> 2 -> 5 -> 9.

List of traversed IP interfaces:

IP Address	Hop Delay	Node Name
192.0.11.1	0,00000	Enterprise Network.node_0
192.0.10.1	0,11449	Enterprise Network.node_2
192.0.28.2	0,02117	Enterprise Network.node_5
192.0.18.1	0,02153	Enterprise Network.node_9

Отже, результат моделювання підтвержено в пакеті OPNet. Для даного випадку вибір шляху 0-3-5-9 як оптимального також має сенс, оскільки по обраному критерію якості обидва маршрути рівнозначні.

Висновок: Програма, написана по алгоритму динамічного програмування придатна для розрахунку процесу маршрутизації на основі такого показника якості, як кількість проміжних вузлів від відправника до одержувача.

Дослід №3

Щоб максимально приблизити дане дослідження до реальності, було вирішено промоделювати роботу даної топології з двома різними протоколами динамічної маршрутизації. У випадку, коли маршрутизатор працює по кільком різним протоколам, для прийняття рішення про оптимальність маршруту використовується додатковий критерій, що називається адміністративною відстанню АВ (Administrative distance), який характеризує ступінь довіри до того чи іншого джерела інформації про маршрут. Адміністративна дистанція виражається числовим параметром, який може приймати значення від 0 до 255. Кожен протокол маршрутизації має своє значення адміністративної дистанції, при чому чим менше значення АД, тим більш достовірною вважається інформація про маршрут.

Таблиця 2.

Назва протоколу	Значення АД
Мережа безпосередньо з'єднана з даним вузлом	0
Статичний маршрут	1
Маршрут отриманий по OSPF	110
Маршрут отриманий по RIP	120
Походження маршруту невідоме	255

Для моделювання процесу маршрутизації по двом критеріям необхідно в алгоритм програми додати блок, що дозволить враховувати значення АД. Вхідними даними для програми будуть дві матриці: матриця сполучень (як і в випадку пошуку оптимального маршруту по одному критерію) та матриця адміністративних дистанцій (щоб мати змогу оцінювати ступінь довіри до протокола).

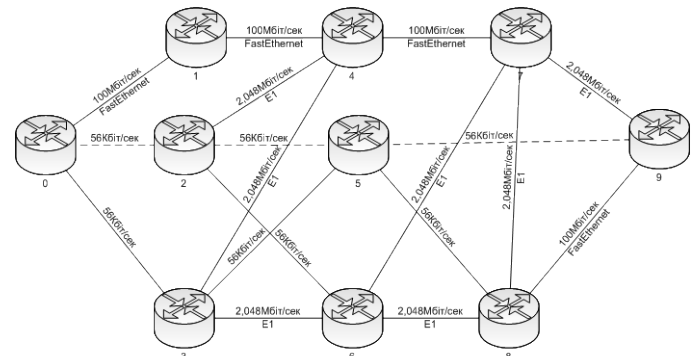


Рис. 8. Модифікована схема мережі

В даній праці для перевірки роботи програми з урахуванням двох критеріїв якості, необхідно модифікувати схему мережі, наведену на рис. 4 таким чином, щоб ділянки, які в попередніх дослідках були обрані оптимальними по протоколу OSPF або RIP, продовжували працювати по тому ж протоколу. В модифікованій мережі було використано комбінацію протоколів RIP та OSPF, таким чином, що на ділянці 0 – 2 – 5 – 9 настроєно RIP, а на всіх інших – OSPF.

Якщо врахувати особливості роботи модифікованого алгоритму програми (мається на увазі, що для вибору оптимального маршруту два критерії якості для кожного з'єднання додаються і уже по мінімальному значенню суми обирається оптимальний маршрут), цілком логічно було припустити, що результати дослідів будуть відрізнятися від тривіального рішення.

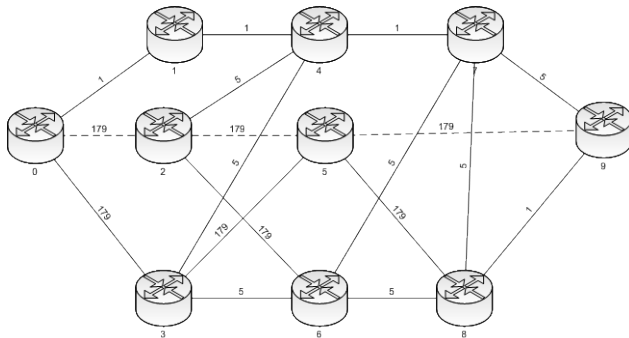


Рис. 9. Модифікована схема мережі у вигляді графу

Вхідними даними для моделювання в середовищі MATLAB 7.0, як уже було сказано вище, є дві матриці.

<i>C</i>	999	1	179	179	999	999	999	999	999	999
	1	999	999	999	1	999	999	999	999	999
	179	999	999	999	5	179	179	999	999	999
	179	999	999	999	5	179	5	999	999	999
	999	1	5	5	999	999	999	1	999	999
	999	999	179	179	999	999	999	999	179	179
	999	999	179	5	999	999	999	5	5	999
	999	999	999	999	1	999	5	999	5	5
	999	999	999	999	999	179	5	5	999	1
	999	999	999	999	999	179	999	5	1	999
<i>D</i>	999	110	120	110	999	999	999	999	999	999
	110	999	999	999	110	999	999	999	999	999
	120	999	999	999	110	120	110	999	999	999
	110	999	999	999	110	110	110	999	999	999
	999	110	110	110	999	999	999	110	999	999
	999	999	120	110	999	999	999	999	110	120
	999	999	110	110	999	999	999	110	110	999
	999	999	999	999	110	999	110	999	110	110
	999	999	999	999	999	110	110	110	999	110
	999	999	999	999	999	120	999	110	110	999

В результаті застосування програми, що написана по алгоритму динамічного програмування і модифікована для роботи з двома критеріями якості, було отримано наступний маршрут:

```

result =
    CurrentTime: 2
    CurrentVertex: 1
    NextVertex: 4
    Cost: 289
result =
    CurrentTime: 3
    CurrentVertex: 4
    NextVertex: 6
    Cost: 289
result =
    CurrentTime: 4
    CurrentVertex: 6
    NextVertex: 10
    Cost: 299
    
```

Отже, по результатам моделювання в середовищі MATLAB 7.0 маршрут 0 – 3 – 5 – 9 було обрано найоптимальнішим.

Щоб перевірити ці дані необхідно модифікувати і модель в середовищі OPNet як показано на рисунку 10.

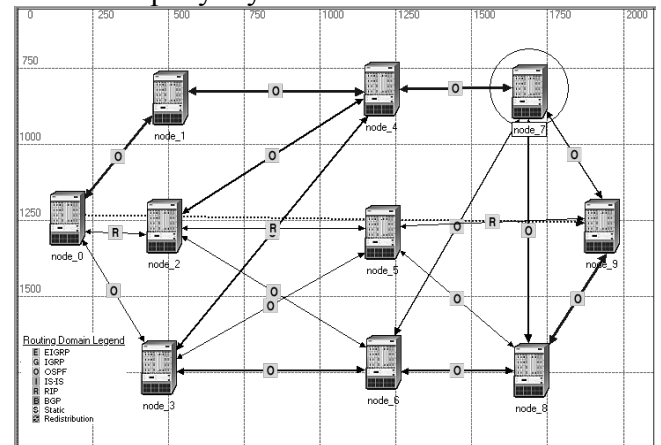


Рис. 10. Модель модифікованої мережі в OPNet

В результаті виконаних налаштувань найбільш оптимальним був обраний шлях:

List of traversed IP interfaces:

IP Address	Hop Delay	Node Name
192.0.14.2	0,00000	Enterprise Network.node_0
192.0.13.1	0,00304	Enterprise Network.node_1
192.0.2.1	0,00403	Enterprise Network.node_4
192.0.12.2	0,00281	Enterprise Network.node_7
192.0.18.1	0,00221	Enterprise Network.node_9

Моделювання процесу маршрутизації на основі двох критеріїв якості, принесло не зовсім очікувані результати, оскільки оптимальний маршруту 0-1-4-7-9 вирахований засобами середовища OPNet відрізнявся від отриманого в середовищі MATLAB, так як був отриманий на основі роботи алгоритму динамічного програмування, що не повторює собою застосовувані на даний момент алгоритми маршрутизації мережевого рівня.

Висновок: Отримана таким чином інформація, що описує стан мережі вказаними вище методами, надає можливість отримати альтернативне оптимальне рішення задачі з двома критеріями якості, завдяки чому можна знаходити множину оптимальних маршрутів по першому з показників якості (кількість проміжних вузлів) і уже з цієї множини обирати оптимальне рішення по другому показнику якості (пропускна здатність сполучення). За допомогою методу динамічного програмування, застосованому для моделювання процесу маршрутизації, можна знизити ймовірність утворення черг в телекомунікаційних мережах, а отже і зменшити ймові-

рність втрати пакетів даних. Даний метод надає можливість організації оптимального розподілення мережевих ресурсів, за рахунок перерозподілу навантаження на найбільш оптимальних маршрутах, за рахунок використання сполучень, що зазвичай не використовуються для передачі даних в стандартних алгоритмах.

Окрім того застосування даного методу зменшує час, що витрачається на процес вибору маршруту за інших рівних умов, оскільки часткові рішення задачі пошуку оптимального шляху зберігаються у буфері, то при визначенні повної задачі є можливість використовувати вже існуючі, розраховані часткові рішення, що в поєднанні дають рі-

шення початкової задачі. Підтвердження даної гіпотези стає можливим завдяки процесу емуляції програмної реалізації математичних методів на спеціалізованому технічному обладнанні, що розроблене для моделювання та тестування мережевих рішень рівня Network моделі OSI.

В нашому випадку основною складністю та недоліком у процесі моделювання на основі методу динамічного програмування являється процес складання матриці з'єднань, оскільки матриці з'єднань складаються вручну. В подальшому планується написання спеціального блоку підпрограм, що вирішує дану проблему.

Список посилань

1. Чураков Е.П. Оптимальные и адаптивные системы : учеб. пособие для вузов / Чураков Е.П. – М. : Энергоатомиздат, 1987. – 256 с.
2. J. Postel. RFC792 - Internet Control Message Protocol. 1981.
3. Репнікова Н.Б., Дорошенко К.С., Жуковський О.О. Метод моделювання процесу маршрутизації в IP-мережі за допомогою мови програмування MatLab /«Штучний інтелект» 2'2009. – С.63-68.
4. T. Svensson, A. Popescu –Development of laboratory exercises based on OPNET Modeler / Blekinge Institute of Technology, 2003.
5. Карачун В.Я., Бех П.О., Гульчук Г.Г., Карачун О.О., Коряченко В.Г., Прохур Ю.З., Черненко І.А, Чиж С.М. Російсько-українсько-англійський науково-технічний словник – Київ «Техніка»,1997

Поступила в редакцію 8.12.2009