

4. Шенк Х. Теория инженерного эксперимента /Х. Шенк/. – М.: Мир, 1972. – 384 с.
5. Вентцель Е.С. Теория вероятностей. /Е.С. Вентцель/ – М.: Наука, 1969. – 576 с.
6. Справочник по теории вероятностей и математической статистики / [Королюк В.С., Портенко Н.И., Скороход А.В., Турбин А.Ф.]. – М.: Наука, 1985. – 640 с.
7. Гмурман В.Е. Руководство к решению задач по теории вероятностей и математической статистике. / Гмурман В.Е. – М.: Высшая школа, 1975. – 146 с.
8. Статистика: підручник /А.В. Головач, А.М. Єріна, О.В. Козирєв та інш; за ред. А.В. А.В. Головача, А.М. Єриної, О.В. Козирєва. – К.: Вища школа, 1993. – 623 с.

Анотації:

Наведені методика та результати експериментального дослідження напруги відпускання аварійних реле залізничної автоматики.

Ключові слова: реле, напруга відпускання, інтервал, допуск.

Представлены методика и результаты экспериментального исследования напряжения отпускания аварийных реле железнодорожной автоматики.

Ключевые слова: реле, напряжение отпускания, интервал, допуск.

Presented methods and results of the experimental study of the voltage of the unhooking emergency relay railway automation.

Keywords: relay, voltage of the unhooking, interval, tolerance.

УДК 656.212.5:681.3

ПАХОМОВА В. М., к.т.н., доцент (ДНУЗТ).

Дослідження запобігання нестабільностям роботи протоколу RIP на програмній моделі

Вступ

Дистанційно-векторний протокол RIP (Routing Information Protocol) - один з найбільш розповсюджених протоколів внутрішньої маршрутизації в комп'ютерних мережах, які дозволяють маршрутизаторам динамічно оновлювати маршрутну інформацію, отримуючи її від сусідніх маршрутизаторів. На жаль, поведінка дистанційно-векторних протоколів (і зокрема, протоколу RIP) при зміні топології системи не завжди коректна та передбачувана. Основна проблема – зациклювання, тобто данні, адресовані в певну мережу будуть пересилатися між двома

вузлами до тих пір, поки не закінчиться час життя дейтаграм і вони не будуть знищені [1]. Для того, щоб уникнути зациклювання, в алгоритм розсилки векторів відстаней вносяться наступні доповнення: 1) якщо дейтаграми, адресовані в мережу X , надсилаються через маршрутизатор G , що знаходиться в мережі N , то в векторі відстаней, що розсилаються в мережі N , відстань до мережі X не вказується; 2) якщо маршрутизатор G оголошує нову відстань до мережі X , то ця відстань вноситься в таблиці маршрутів вузлів, відправляються дейтаграми в мережу X через G , незалежно від того, більше вона чи менше вже внесеної в таблиці відстані.

Очевидно, що при виконанні вищевказаних умов, зациклювання не утворюються та будуються коректні маршрути. Однак, таким чином, усуваються далеко не всі випадки зациклювання.

Аналіз досліджень та публікацій

Дистанційно-векторні алгоритми DV (Distance Vector) засновані на обміні дуже незначним обсягом інформації. Передбачається, що кожен об'єкт (маршрутизатор або хост), який бере участь в обміні інформацією, зберігає відомості про всіх одержувачів в системі. У загальному випадку інформація про всіх підключених до однієї мережі об'єктів збирається в єдиний запис, який описує маршрут до всіх одержувачів у даній мережі [1]. Потрібно зберігати для кожного одержувача такі відомості: IP-адреса хоста або мережі; перший маршрутизатор на шляху до одержувача; фізичний інтерфейс, який використовується для зв'язку з першим маршрутизатором; метрика (число, що показує «віддаленість» одержувача); таймер (час, який минув з моменту останнього оновлення запису). Ця база даних ініціалізується з описом об'єктів, які безпосередньо підключені до системи, а потім оновлюється згідно з інформацією, що одержується в повідомленнях від найближчих сусідів. Найбільш важлива інформація між маршрутизаторами передається в оновленнях (update message). Кожен об'єкт, бере участь у схемі маршрутизації, передає оновлення, що описують поточний стан своєї маршрутної бази даних.

У простих мережах в якості метрики природно використовувати лічильник інтервалів (хопів) до одержувача. У більш складних мережах метрика може враховувати величину затримки, витрати на доставку та інші параметри, які можуть впливати на вибір маршруту.

Нехай $D(i,j)$ являє метрику кращого маршруту від об'єкта i до об'єкта j (ця метрика повинна бути визначена для кожної пари об'єктів); $d(i,j)$ представляє вартість окремих кроків. Припустимо, що $d(i,j)$ являє вартість прямої доставки від i до j . Ця вартість буде нескінченною, якщо i та j не є прямими сусідами.

Краща метрика описується виразом $D(i,i)=0, D(i,j)=\min[d(i,k)+D(k,j)]$ і найкращий маршрут від i слідує до сусіда k , для якого $d(i,k)+D(k,j)$ має мінімальне значення. Відзначимо, що друге рівняння відноситься до вузла k , що є безпосереднім сусідом i . Для інших випадків значення $d(i,k)$ нескінченне і ніколи не може давати мінімуму. З наведеного доказу очевидно, що можна побудувати простий алгоритм розрахунку вартості для будь-якого маршруту. Об'єкт i запитує у свого сусіда k його оцінку відстаней до адресата і після цього до кожного з отриманих значень просто додається $d(i,k)$. Після цього i може порівняти значення, отримані від усіх сусідів і визначити найбільш «дешевий» шлях [1].

Робота даного алгоритму заснована на припущенні, що кожен об'єкт зберігає копії оцінок, отриманих від всіх сусідів, і визначає мінімальне значення з використанням цих копій. При отриманні відомостей про більш привабливу метрику до запису вносяться відповідні корективи. Такий підхід дозволяє істотно скоротити обсяг обчислень і розміри збережених таблиць [2].

Протокол RIP містить низку доповнень до базового алгоритму DV. Кожен об'єкт, який бере участь в роботі протоколу, повинен виконувати наступні процедури.

- Підтримка таблиці з записом для кожного можливого одержувача в системі. Запис містить відомості про «відстані» до адресата D і перші маршрутизатори (G) на шляху до адресата [3].

- Періодична розсилка кожному сусідові повідомлень про зміну маршрутів. Таке оновлення являє собою набір повідомлень, що містять всю інформацію з маршрутної таблиці. Оновлення містить запис для кожного одержувача з зазначенням дистанції до нього.

- Коли приходить оновлення від сусіда G' , до нього додається вартість для мережі, пов'язаної з G' (це має бути мережа, через яку прийнято оновлення). Після цього обчислюється результат дистанції D і проводиться порівняння із записами поточної таблиці маршрутизації. Якщо нова дистанція D' для адресата N менше існуючого значення D , приймається новий маршрут. Таким чином, в запис для адресата N включатиме дистанцію D' і маршрутизатор G' . Якщо маршрутизатор G' є тим, з якого починається існуючий маршрут (тобто, $G'=G$), нова метрика D' має включатися в таблицю навіть у тих випадках, коли вона перевищує стару.

Передбачається, що кожний маршрутизатор, який підтримує RIP, має таблицю маршрутизації. Ця таблиця має один запис для кожного адресата, досяжного через систему, яка використовує RIP. У кожного запису повинні міститися наступні відомості: адреса IPv4 для одержувача; метрика, яка представляє загальну вартість доставки даних від маршрутизатора до адресата; адреса IPv4 для наступного маршрутизатора на шляху до адресата (next hop); прапор, який показує, що інформація про маршрут була нещодавно змінена (route change flag); різні таймери, пов'язані з маршрутом. Кожен запис має містити також маску підмережі. Маски разом з адресою IPv4 для одержувача дозволяють маршрутизатору ідентифікувати різні підмережі в одній мережі, а також враховувати маски віддалених підмереж [4]. Можливо введення додаткових маршрутів (це маршрути до хостів чи мереж за межами видимості системи маршрутизації), такі

маршрути називають статичними (static route).

В протоколі RIP є два типи повідомлень, якими обмінюються маршрутизатори: відповідь (response): розсилка вектору відстані; запит (request): маршрутизатор запрошує у сусідів їх маршрутні таблиці чи данні про певний маршрут. Формат повідомлень обох типів однаковий, загальний вигляд яких приведений в [4].

Створення таблиць маршрутизації. Таблицю маршрутів називають таблицею, яка є результатом діяльності протоколу RIP, що складається з рядків з полями «Мережа», «Відстань», «Наступний маршрутизатор». Вектором відстаней називається набір пар («Мережа», «Відстань до цієї мережі»), витягнутий із таблиці маршрутів; кожен таку пару називають елементом вектора відстаней [3].

Кожен маршрутизатор, на якому запущено RIP, періодично широкомовно поширює свій вектор відстаней. Вектор поширюється через всі інтерфейси маршрутизатора, підключених до мереж, що входять до RIP-системи. Кожен маршрутизатор також періодично отримує вектори відстаней від інших маршрутизаторів. Відстань в цих векторах збільшуються на 1, після чого порівнюються з даними в таблиці маршрутів, і, якщо відстань до якоїсь з мереж в отриманому векторі виявляється менше відстані, вказаної в таблиці, значення з таблиці заміщується новим (меншим) значенням, а адреса маршрутизатора, що надіслали вектор з цим значенням, записується в полі «Наступний маршрутизатор» в цьому рядку таблиці. Після цього вектор відстаней, що розсилаються даними маршрутизатором, відповідно зміниться.

Цілі статті

Ознайомитися з можливими способами передачі маршрутних оновлень

та запропонувати відповідну формальну модель. Розробити програмну модель RIP, в основі якої запропонована формальна модель. На програмній моделі провести дослідження часу перебудови таблиць маршрутизації за різними методами.

Основний матеріал

Робота зі змінами топології. На практиці маршрутизатори та канали мають властивість «падати» і відновлювати свою працездатність. Для того, щоб враховувати такі особливості, потрібно внести в алгоритм деякі зміни. Теоретичний варіант алгоритму використовує інформацію від усіх найближчих сусідів. При зміні топології змінюється і набір сусідів. У результаті такої зміни при подальшому розрахунку ці зміни будуть враховані. Однак, як було зазначено вище, в реальних програмах використовується варіант мінімізації з урахуванням зростання (incremental version of the minimization), при якому запам'ятовується тільки кращий маршрут. Якщо маршрутизатор, включений у цей маршрут «падає», або розривається мережне з'єднання, розрахунок може не відобразити таких змін. У реальних системах алгоритм залежить від способу, використовуваного маршрутизатором для передачі повідомлень про зміну топології. Якщо маршрутизатор припиняє працювати, у нього вже немає можливості повідомити своїх сусідів про зміну топології. Для вирішення цієї проблеми протоколи на основі алгоритмів DV повинні використовувати поняття тайм-ауту для маршрутизаторів. Деталі реалізації залежать від конкретного протоколу.

Так, зокрема, в протоколі RIP кожен маршрутизатор передає оновлення всім своїм сусідам кожні 30 с. Відзначимо, що для оголошення маршруту некоректним необхідно чекати протягом 180 с. Це зроблено для того, щоб позбутися від помилкових тайм-аутів в результаті втрати пакетів при констатації тайм-ауту в результаті відсутності одного повідомлення.

Для індикації недоступності мережі використовується спеціальне значення метрики (16 в протоколі RIP). Це значення зазвичай трактується як «нескінченність», оскільки воно перевищує всі інші значення метрики.

Запобігання нестабільностям. Алгоритм DV буде завжди забезпечувати маршрутизаторам можливість розрахунку коректної таблиці маршрутизації, однак на практиці такої можливості виявляється недостатньо. Алгоритм DV лише гарантує сходження розрахунків для таблиць маршрутизації за кінцевий час, однак немає жодних гарантій, що цей час буде досить малим і незрозуміло, що може відбутися з метрикою, коли мережа стане недоступною. Математичну сторону алгоритму досить просто розширити для обробки маршрутів, що стали недоступними. Запропоновані вище угоди дозволяють це зробити, просто вказавши досить велике значення метрики для недоступних мереж. Це значення повинне перевищувати будь-яку реально використовувану метрику (значення 16 в якості метрики для недоступних мереж).

Припустимо, що мережа перестала бути доступною. Всі сусідні з нею маршрутизатори перестануть бачити мережу і після тайм-ауту встановлять для неї метрику 16. Для аналізу можемо припустити, що всі сусідні маршрутизатори отримали пряме з'єднання з мережею, що зникла, ці з'єднання характеризуються метрикою 16. Оскільки зі зниклою мережею є тільки одне з'єднання, всі інші маршрутизатори в системі будуть сходиться до нових маршрутів, що проходять через один із сусідніх з зниклою мережею маршрутизаторів. Легко побачити, що після сходження розрахунку всі сусідні маршрутизатори отримають для зниклої мережі метрику не менше 16. Маршрутизатори, що слідуєть за найближчими сусідами (1 хоп), отримають метрику не менше 17, що слідуєть за ними - не менше 18 і т. д. Оскільки всі ці значення перевищують максимум, метрика буде встановленою 16. Зазвичай системи будуть сходиться на значенні 16

в якості метрики для зниклої мережі. На жаль, питання про час сходження розрахунків не настільки простий.

Найгірша ситуація виникає в тих випадках, коли мережа стає повністю недоступною з якоїсь частини системи. У таких випадках метрика повільно збільшується, поки не буде обчислено недоступність, такі ситуації називаються «обчислення нескінченності» (counting to infinity). Зі сказаного можна зрозуміти, чому для «нескінченної» метрики вибрано настільки мале значення. Якщо мережа стає повністю недоступною, необхідно виявити недоступність якомога скоріше. Значення метрики для недоступної мережі повинно бути більше метрики будь-якого реального маршруту, але ніщо не вимагає збільшувати його додатково. Таким чином, вибір метрики для недоступної мережі визначається компромісом між розміром мережі та швидкістю «обчислення нескінченності». Для запобігання проблем, подібних до описаної, існує кілька способів.

Розщеплення горизонту (Split horizon) являє собою схему запобігання проблем, викликаних включенням маршруту в оновлення, що передаються маршрутизатору, від якого була отримана інформація про даний маршрут. Проста схема розщеплення горизонту (Simple split horizon) опускає маршрути, отримані від сусіда при передачі оновлень цьому сусідові. Схема «Split horizon with poisoned reverse» включає такі маршрути в оновлення, але встановлює для них нескінченну метрику (анонсування недоступності зворотного маршруту).

Оновлення подією (Triggered update) є спробою прискорити процес сходження метрики. Для використання Triggered update просто додається правило, за яким маршрутизатор, змінюючи метрику на шляху, повинен передати повідомлення про це, не чекаючи часу штатної передачі оновлення. Проміжок часу, по закінченню якого повинно передаватися таке повідомлення, залежить від протоколу.

Припустимо, що шлях від маршрутизатора в мережу N проходить через ма-

шрутизатор G . Якщо оновлення приходить безпосередньо від G , який прийняв його, маршрутизатор повинен довіряти отриманої інформації незалежно від знаку зміни метрики. Якщо в результаті метрика змінюється, прийняв оновлення маршрутизатор буде передавати в зв'язку з цією подією оновлення (Triggered updates) всім маршрутизаторам, безпосередньо підключеним до нього. Кожен з вузлів, що одержали оновлення, може передати його своїм сусідам. У результаті виникає каскад оновлень. Легко побачити, які маршрутизатори будуть залучені в цей каскад. Припустимо, що маршрутизатор G фіксує тайм-аут для доступу в мережу N . У результаті цієї події G передаватиме оновлення всім своїм сусідам. Однак довіряться цьому оновленню тільки ті маршрутизатори, чий маршрут в мережу N проходить через G . Решта маршрутизаторів побачать цю інформацію про новий маршрут, який вони використовують, і просто проігнорують оновлення. Сусіди, чий маршрут проходить через G , оновлять свою метрику та відправлять Triggered update всім своїм сусідам. І знову ці оновлення будуть сприйняті лише тими маршрутизаторами, для яких шлях в мережу N проходить через G . Таким чином, оновлення події назад по всіх шляхах, отриманими від маршрутизатора G , встановлюють для цього шляху нескінченну метрику. Поширення оновлень припиниться, як тільки вони попадуть в ту частину мережі, звідки шлях у N йде через інші маршрутизатори. Якщо система зберігає «спокійний» стан, поки поширюється каскад оновлень, «обчислення нескінченності» ніколи не буде потрібно. «Погані» маршрути завжди будуть негайно видалятися і петель в маршрутизації виникати не може. На жаль, в реальності все складніше. Доки передаються оновлення події, можуть бути згенеровані та передані штатні оновлення. Маршрутизатори, ще не отримали оновлення події, будуть продовжувати розсилку інформації, яка містить недоступний маршрут. Можливо також після прийому оновлення

події отримання штатного оновлення від маршрутизатора, який ще не отримав оновлення події. В результаті цього статус недоступного маршруту може бути відновлений некоректно (як доступного). Якщо оновлення по подіях розсилаються

часто, вірогідність такої події стає досить великою. Однак можливість «обчислення нескінченності» зберігається. Порівняльна характеристика способів передачі маршрутних оновлень наведена в табл.1.

Таблиця 1. Порівняльна характеристика способів передачі

	Split horizon	Split horizon with poisoned reverse	Triggered update
призначення	розрив петель між парою маршрутизаторів	розрив петель між парою маршрутизаторів	розрив петель між трьома маршрутизаторами
спосіб	вилучення маршрутів отриманих від сусіда при передачі йому оновлень	анонсування маршрутів, отриманих від сусіда з метрикою 16	додається правило, маршрутизатор змінюючи метрику, повинен це передати
недоліки	не відображає всіх змін	збільшення об'єму повідомлення	зростання об'єму мережного трафіку
коментарі	реалізують в якості опцій в залежності від потреб, оскільки обидва способи не є універсальними та доповнюють один одного		часто використовується протягом деякого часу після змін, а потім відключається

Формальна модель за протоколом RIP. Розроблена діаграма станів за протоколом RIP, що представлена на рис. 1. Робота протоколу починається з ініціалізації (стан «Механізм маршрутних таблиць»); в кожному маршрутизаторі створюється мінімальна таблиця маршрутизації, в якій враховуються тільки безпосередньо приєднані мережі, а також встановлюються таймер тайм-ауту $T_{тайм}=180$ с та таймер розсилки $T_{розс}=30$ с.

Якщо мережа, інтерфейс та маршрутизатор працездатні, то відбувається перехід в стан «Розсилка векторів», де всім своїм сусідам відсилаються повідомлення RIP, в якому

містяться таблиці маршрутизації. (На першому етапі це обмін мінімальними таблицями, в подальшому обмін оновленими таблицями). Цей етап контролюється таймером розсилки, при цьому працює механізм часу життя маршруту, кожен запис таблиці маршрутизації, отриманий по протоколу RIP, має час життя (TTL). При надходженні чергового RIP-повідомлення, яке підтверджує справедливність цього запису, таймер TTL встановлюється в початковий стан, а потім з нього кожен секунду віднімається одиниця. По закінченню $T_{розс}$ повернення в стан «Механізм маршрутних таблиць».

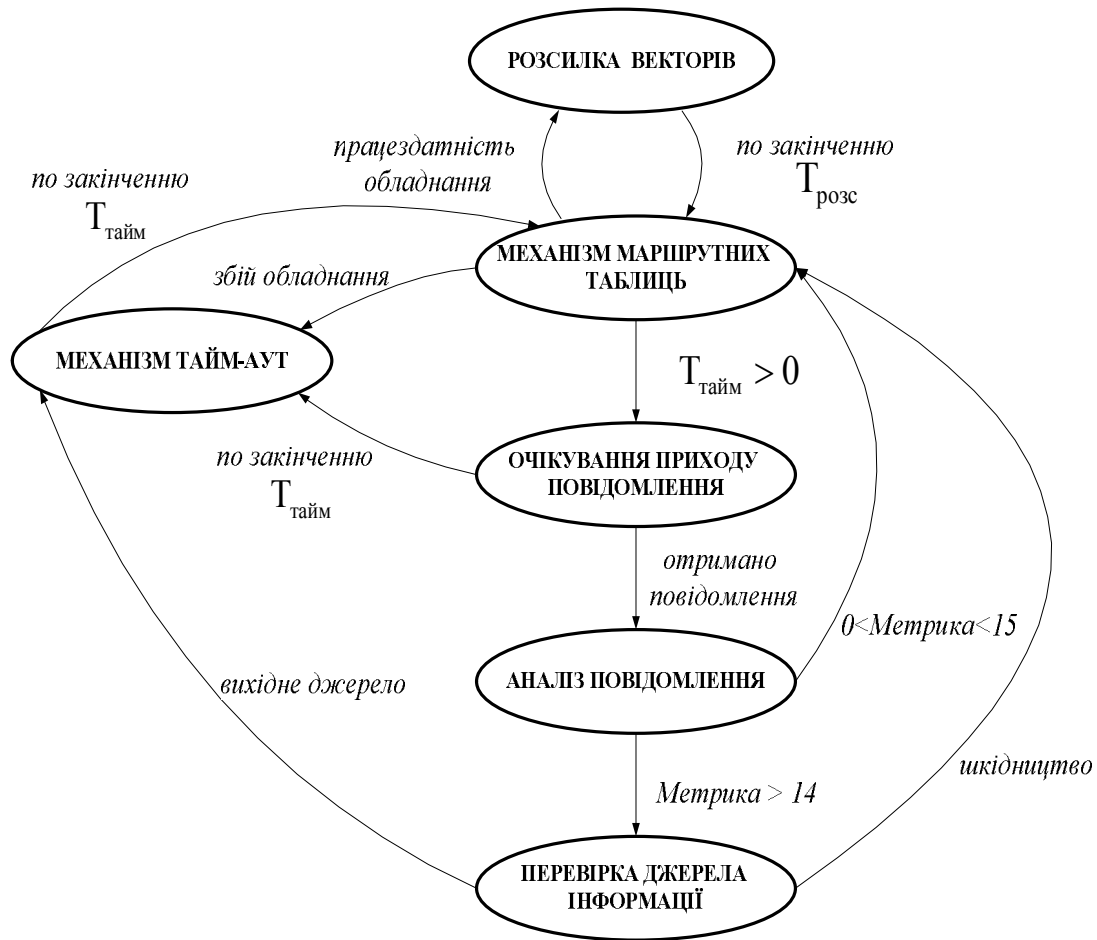


Рис. 1. Діаграма станів за протоколом RIP.

Якщо має місце збій обладнання, то перехід в стан «Механізм тайм-ауту»; якщо тайм-ауту незавершений ($T_{\text{тайм}} > 0$), то відбувається перехід в стан «Очікування приходу повідомлення», при отриманні якого перехід в стан «Аналіз повідомлення».

Якщо $1 < \text{метрика} < 15$, то перехід в стан «Механізм маршрутних таблиць», якщо $\text{метрика} > 14$, то перехід відбувається в стан «Перевірка джерела інформації», в якому маршрутизатор повинен перевірити, виходить ця «погана» інформація про мережу від того ж маршрутизатора, повідомлення якого послужило в свій час підставою для запису про дану мережу в таблицю маршрутизації. Якщо це той же маршрутизатор, то інформація вважається достовірною, здійснюється перехід в стан «Механізм тайм-ауту»; якщо ж

інформація прийшла не від того маршрутизатора, від якого вона була отримана раніше, вважається що збій в мережі (шкідництво) і таке повідомлення ігнорується, відбувається перехід в стан «Механізм маршрутних таблиць» і процес повторюється.

Програмна модель «RIP». Розроблена програмна модель «RIP» має певні обмеження: мережа складається з чотирьох маршрутизаторів; маршрутизатори мають чітко фіксоване положення на екрані; мережа, що вийде з строю, задається перед запуском моделі; затримка між кроками алгоритму залежить від апаратного забезпечення. Програмна модель реалізує роботу протоколу RIP, що дозволяє відобразити динамічне оновлювання маршрутної інформації, отримуючи її від сусідніх маршрутизаторів, при цьому вихід з не-

стабільних ситуацій реалізується у двох режимах: тайм-аут та Split horizon.

Мережа має дві характеристики: IP-адресу та номер маршрутизатора (номер порту, через який зв'язаний з мережею). Маршрутизатор описаний наступними складовими: мережа, номери портів відносно маршрутизатора та відносно мережі. При запуску програмної моделі

«RIP» на екрані з'являється головна форма. Необхідно обрати: мережу, яка вийде з строю в процесі моделювання; метод перебудови таблиць маршрутизації: тайм-аут та Split horizon; режим запуску моделі: «Результат» (дозволяє отримати кінцеві дані), «Демонстрація» (для поетапного відображення роботи протоколу).

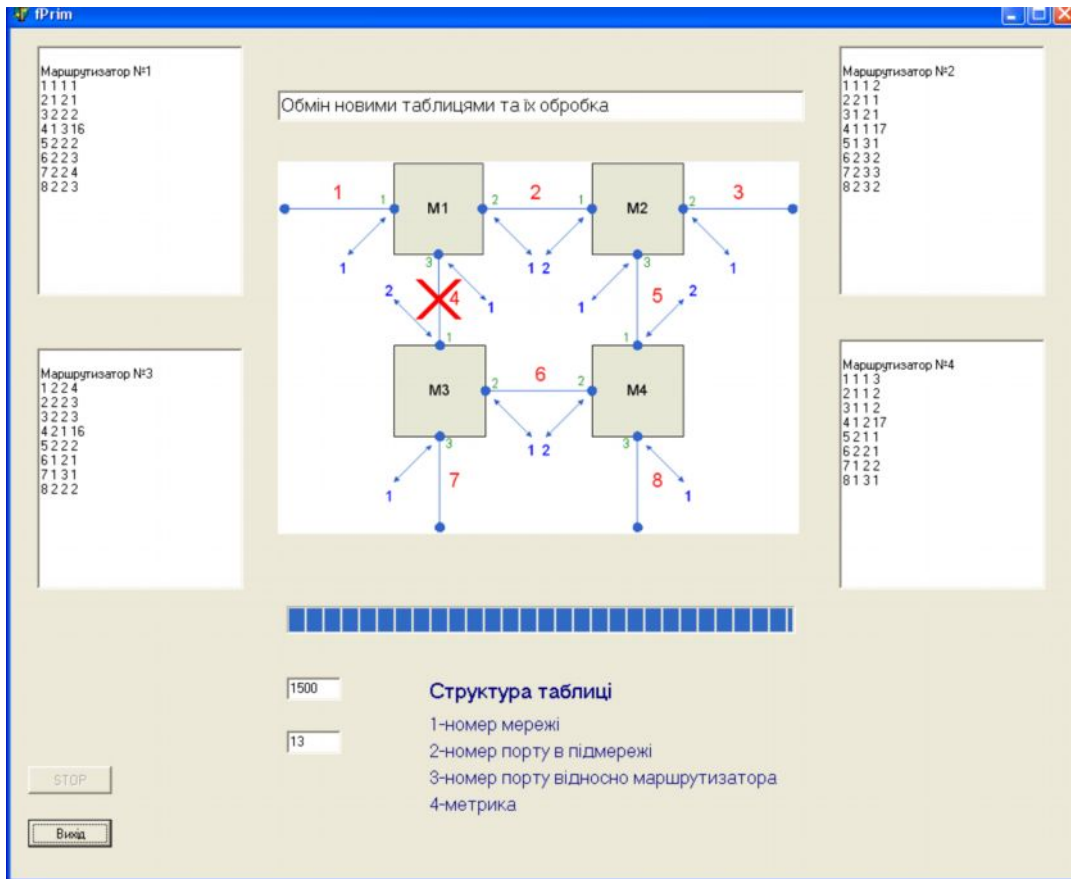


Рис. 2. Результат роботи програмної моделі «RIP» за механізмом тайм-ауту.

Результуючі дані. Проведена серія запусків програмної моделі «RIP» в різних режимах та з різними початковими даними. Кожний варіант таких даних задавався для пошуку в двох режимах. Результати, що видавались програмною моделлю, проаналізовані на правильність роботи алгоритму та на кількість операцій, що необхідні для побудови таблиць маршрутизації. Одною з найважливіших характеристик протоколів маршрутизації є динамічна модифікація маршрутизації при зміні топології мережі,

яку можна проаналізувати на основі часу моделювання, кроків та складності реалізації.

Час моделювання за механізмом тайм-ауту складає 1500 одиниць модельного часу, кількість кроків варіюється від 10 до 13, а за методом Split horizon час становить 500 одиниць модельного часу, кількість кроків від 5 до 7. Зазначимо, що кількість кроків першого варіанту значно перевищує кількість кроків в другому. Кількість кроків залежить від того яка мережа вийшла з

строю: якщо суміжна, наприклад, 2, 4, 5, 6, то оновлення таблиць та встановлення стабільної роботи відбувається за меншу кількість кроків у порівнянні з крайніми мережами 1, 3, 7, 8.

По складності реалізації механізм тайм-ауту потребує більшої кількості

обчислювальних операцій, при цьому не вирішує всі проблеми; метод Split horizon є дещо простіше в реалізації та значно скорочує об'єм трафіку, що передається, за рахунок технології розбиття горизонту.

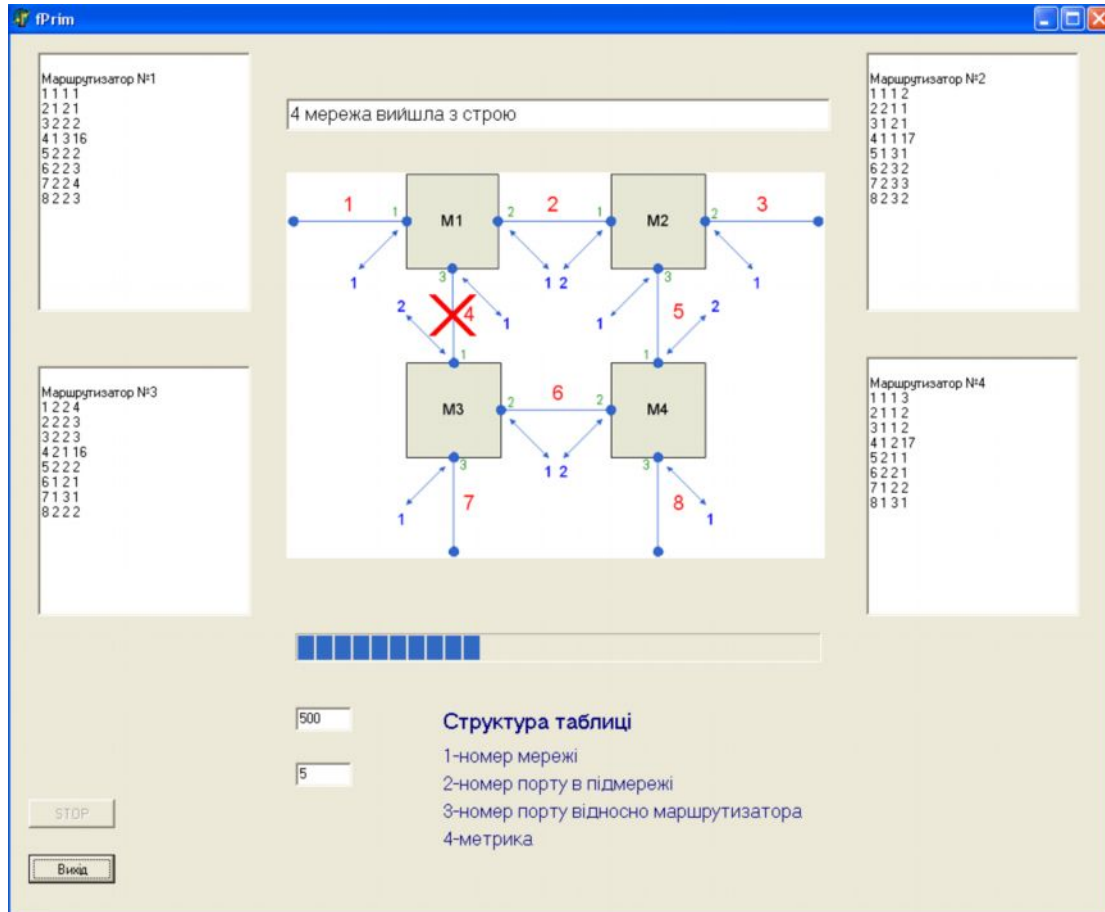


Рис. 3. Результат роботи програмної моделі «RIP» за методом Split horizon.

На рис. 2-3 представлені результати запуску програмної моделі «RIP» при однакових початкових даних за різними методами перебудови таблиць маршрутизації: тайм-аут і Split horizon. При цьому отримані результати однакові, що доводить правильність роботи створеної моделі. Сходження розрахунків для таблиць маршрутизації відбулося за кінцевий час моделювання.

Висновки

1. За протоколом RIP на основі стандарту RFC 2453 розроблено форма-

льну модель, що містить наступні стани: механізм маршрутних таблиць, розсилка векторів, механізм тайм-ауту, очікування повідомлення, аналіз повідомлення, перевірка джерела інформації. При стабільній роботі мережі застосування механізму тайм-ауту дає досить добрий результат.

2. Серед способів передачі маршрутних оновлень розрізняють: Split horizon, Split horizon with poisoned reverse, Triggered update, основне призначення яких це розрив петель. На програмній моделі «RIP» визначено, що раціонально використання методу Split horizon, що дає

зможу будувати коректні таблиці маршрутизації та значно скоротити час їх побудови в комп'ютерних мережах.

3. Доцільно надавати можливість вибору режиму розщеплення горизонту (простий або з анонсування недоступності зворотного шляху) в якості опції. Допустима реалізація гібридних схем, коли анонсується недоступність (метрика 16) лише для частини зворотних маршрутів. Прикладом такої схеми буде використання метрики 16 для зворотних маршрутів протягом деякого часу після зміни маршрутизації та відмова від таких анонсів по закінченні часу.

Список літератури

1. Малкин Ж. И. Стандарт протоколу RIP RFC 2453 [Текст] / Ж. И. Малкин. – СПб: Питер, 1998.
2. [Семенов Ю. А. Телекоммукационные технологии](http://book.itep.ru) [Електроний ресурс] / Ю. А. Семенов. – <http://book.itep.ru>
3. Берлин А. Н. [Основные протоколы интернет](http://www.intuit.ru) [Електроний ресурс] / А. Н. Берлин. – <http://www.intuit.ru>
4. Олифер В. Г. Компьютерные сети. Принципы, технологии, протоколы [Текст] / В. Г. Олифер, Н. А. Олифер. – СПб.: Питер, 2006. – 958 с.

Анотації:

Складена формальна модель роботи протоколу RIP, яка покладена в основу розробленої програмної моделі, що дозволяє відобразити динамічне оновлювання маршрутної інформації, отримуючи її від сусідніх маршрутизаторів, при цьому вихід з нестабільних ситуацій реалізується за механізмом тайм-ауту та

методу Split horizon. Встановлено, що використання методу Split horizon дає змогу будувати коректні таблиці маршрутизації та значно скоротити час їх побудови в комп'ютерних мережах. Визначено, що доцільно надавати можливість вибору режиму розщеплення горизонту: простий або з анонсування недоступності зворотного шляху.

Ключові слова: метрика, таблиця маршрутизації, зациклювання, тайм-аут, розщеплення горизонту, оновлення подією.

Составлена формальная модель работы протокола RIP, которая положена в основу разработанной программной модели, позволяющей отобразить динамическое обновление маршрутной информации, получая ее от соседних маршрутизаторов, при этом выход из нестабильных ситуаций реализуется по механизму тайм-аута и метода Split horizon. Установлено, что использование метода Split horizon позволяет строить корректные таблицы маршрутизации и значительно сократить время их построения в компьютерных сетях. Определено, что целесообразно предоставлять возможность выбора режима расщепления горизонта: простой или анонсирования недоступности обратного пути.

Ключевые слова: метрика, таблица маршрутизации, зацикливания, тайм-аут, расщепление горизонта, обновление событием.

Compiled formal model RIP protocol, which form the basis of software developed model allows you to display dynamic update of routing information, getting it from the neighboring routers, where the output of volatile situations is realized by the mechanism time-out method and Split horizon. Found that the use of the Split horizon allows you to build the correct routing table and significantly reduce the time of their construction in computer networks. Determined that it is advisable to provide the ability to select the Split horizon: Simple split horizon or Split horizon with poisoned reverse.

Keywords: metric, routing table, looping, time-out, Split horizon, update event.

