Висновки

1. Запропоновані модифікації шифрування призначені для шифрування зображень у градаціях сірого і грунтуються на використанні ідей базового алгоритму RSA.

2. Запропоновані модифікації можна використати стосовно будь-якого типу зображень, але найбільших переваг досягають у випадку використання зображень, які дають змогу чітко виділяти контури.

3. Обидва типи модифікацій без жодних застережень можна використати і стосовно кольорових зображень. Однак, незалежно від типу зображення пропорційно до розмірності вхідного зображення може зрости розмір шифрованого зображення.

4. Стійкість до несанкціонованого дешифрування запропонованою потоковою модифікацією забезпечується стійкістю алгоритмому RSA.

5. При шифруванні зберігається ізоморфність вхідного і дешифрованого набору, що можна використати для визначення інших характеристик зображення: розпізнавання, сегментації тощо.

6. Швидкість роботи запропонованих модифікованих методів шифрування є достатньою для того, щоб бути інтегрованими у комплексні системи реального часу, які вирішують завдання не лише шифрування об'єктів, а й розпізнавання, визначення відбракованих об'єктів, фасування тощо.

1. Павлидис Т. Алгоритмы машиной графики и обработки изображений. – М.: Радио и связь, 1986.-399с. 2. Б.Яне. Цифровая обработка изображений. – М., Техносфера, 2007. – 583с. 3. Брюс Шнайер. Прикладная криптография. – М.: Триумф, 2003. – 815с. 4. Рашкевич Ю.М., Пелешко Д.Д. Ковальчук А.М., Пелешко М.З. Модифікація алгоритму RSA для деяких класів зображень. Технічні вісті 2008/1(27), 2(28). С. 59 – 62. 5. Ковальчук А., Пелешко Д., Хомин М., Борзов Ю. Поєднання алгоритму RSA і побітових операцій при шифруванні – дешифруванні зображень // Вісник Нац. ун-ту «Львівська політехніка» «Комп'ютерні науки та інформаційні технології». – 2011. – №694. – С.309–313. 6. Вельшенбах М. Криптография на Си и С++ в действии: Учеб. пособие. – М.: Издательство Триумф, 2004. – 464 с. 7. Вербіцький О.В. Вступ до криптології. – Львів: Видавництво науково-технічної літератури, 1998. – 247 с.

УДК 681.513

С. Кухарєв
Національний технічний університет України "Київський політехнічний інститут",
Навчально-науковий комплекс "Інститут прикладного системного аналізу",
кафедра математичних методів системного аналізу

КОНЦЕПТУАЛЬНА МОДЕЛЬ ІМІТАЦІЙНОГО АЛГОРИТМУ МОДЕЛЮВАННЯ РОБОТИ МЕРЕЖІ З ТЕХНОЛОГІЄЮ MPLS

Описано концептуальну модель розробленого імітаційного алгоритму, призначеного для моделювання роботи мереж з технологією MPLS, програмна реалізація якого дає змогу досліджувати завантаженість мережевих елементів, аналізувати та оптимізувати характеристики мереж, а також досліджувати поведінку трафіка різних класів.

Ключові слова: імітаційний алгоритм, мережа, трафік, оптимізація характеристик

We describe a conceptual model developed simulation algorithm designed to simulate the networks with technology MPLS, software implementation which allows to study the workload of network elements, analysis and optimization characteristics of networks and study the behavior of different classes of traffic.

Keywords: imitation algorithm, network, traffic, optimizing performance

Вступ

При проектуванні комп'ютерних мереж з технологією MPLS (Multiprotocol Label Switching – багатопротокольна комутація за мітками) [1, 2] виникають такі проблеми: визначення типів та

параметрів мережевого обладнання, яке забезпечило б потреби користувачів при передаванні трафіка, вибір архітектури мережі за критерієм вартості обладнання. Важливим є визначення завантаженості окремих вузлів чи каналів зв'язку мережі при заданих її параметрах, а також дослідження поведінки трафіка різних класів.

У сучасній комп'ютерній мережі з технологією MPLS одночасно передається інформація різних типів (відео- та аудіоінформація, тиснута відео- та аудіоінформація, а також дані менш чутливі до затримок) з різними показниками якості обслуговування, на її роботу істотно впливають методи управління трафіком. Побудова аналітичної моделі для дослідження впливу різних факторів на показники якості функціонування мереж у цій ситуації неможлива, обладнання для побудови мереж з технологією MPLS є доволі дорогим, тому для моделювання роботи мережі, а також аналізу та оптимізації її характеристик пропонується один з найпотужніших інструментів дослідження складних систем – імітаційне моделювання [3–5].

1. Постановка задачі

Нехай задано мережу з технологією MPLS. $N=\{N_i\}, i=1..n$ – множина вузлів: LSR (Label Switching Router – маршрутизатор комутації за мітками) та LER (Label Edge Router – прикордонний маршрутизатор) у мережі, $K=\{K_j\}, j=1..k$ – множина каналів, що з'єднують ці вузли. Нехай $t_d, d=1..D$ – задана d-та одиниця модельного часу, де D – задана тривалість процесу імітації (в одиницях модельного часу) Задано обсяги вхідного $bi_i, i=1..n$ та вихідного $bo_i, i=1..n$ буферів комутаторів, також задано пропускні спроможності $c_j, j=1..k$ усіх каналів зв'язку, продуктивність обробки комутаторами пакетів $pr_i, i=1..n$. Позначимо $P=\{P_r\}, r=1..R$ – множина пакетів у мережі, їх розмір $-S_r, r=1..R$. Позначимо затримку r-го пакета на d-й одиниці модельного часу $delay_{r,d}, r=1..R, d=1..D$. Задано класи пріоритетів обслуговування потоків – $priority_m, m=0..7$, де 0 – найвищий пріоритет, 7 – найнижчий. Позначимо кількість модельного часу, що минув для кожного вузла та каналу, відповідно $delayN_{i,d}, i=1..N, d=1..D$ та $delayK_{j,d}, j=1..K, d=1..D$. Задано кількість прогонів моделі $H=\{h_v\}, v=1..V$.

Необхідно, за заданими параметрами, описати концептуальну модель імітаційного алгоритму, призначеного для моделювання роботи комп'ютерної мережі з технологією MPLS.

2. Концептуальна модель імітаційного алгоритму

2.1. Рекурсивний алгоритм побудови повної топології мережі з усіма шляхами

Крок 1. Обираємо довільний вузол N_{i_1}, вважаємо його кореневим вузлом.

Крок 2. Розглядаємо множину каналів $\{K_{j_w}\}, w=\overline{1,W_1}$, що пов'язані з цим вузлом (входять до нього чи виходять з нього) та ще не розглядалися. Обираємо з цієї множини довільний канал K_{j_1}. Цей канал K_{j_1} таким чином з'єднує вузол N_{i_1} та N_{i_2}. Отримуємо ланцюг $L_{i_1,j_1,i_2} =< N_{i_1}, K_{j_1}, N_{i_2} >$. Ланцюг завжди починається кореневим вузлом та закінчується також вузлом. Вводимо цей ланцюг до загальної множини шляхів мережі.

Крок 3. Розглядаємо множину каналів $\{K_{j_w}\}, w=\overline{1,W_2}$, що пов'язані з останнім у ланцюгу вузлом N_{i_2} (входять до нього чи виходять з нього) та ще не розглядалися. Обираємо з цієї множини довільний канал K_{j_2}. Цей канал K_{j_2}, таким чином, з'єднує вузол N_{i_2} та N_{i_3}. Отримуємо ланцюг $L_{i_1,j_1,i_2,j_2,i_3} =< N_{i_1}, K_{j_1}, N_{i_2}, K_{j_2}, N_{i_3} >$. Вводимо цей ланцюг до загальної множини шляхів мережі.

Повторюємо операції кроку 2 (або кроку 3) для вузла N_{i_3} і т.д., доки не знайдемо вузол чи канал, що вже раніше був присутній у ланцюгу, або не отримаємо ситуацію, за якої інші канали для цього вузла відсутні. Якщо у цьому ланцюгу знайшовся вузол чи канал, що вже раніше був присутній у ланцюгу, або вузол, для якого немає інших каналів, крім того, який з'єднує його з попереднім вузлом, то вважаємо цей вузол чи вузол, що знаходився безпосередньо перед каналом, що вже зустрічався у ланцюгу, вузлом цього ланцюга. Повертаємося на крок назад (до попереднього вузла відносно кінцевого).

При додаванні ланцюга до загальної множини шляхів також фіксується вартість, максимальна та мінімальна пропускні спроможності шляху зокрема, що далі використовуватимуться для скорішого прокладання обхідних LSP (Label Switched Path – шляхів комутації за мітками).

Алгоритм запускаємо для усіх вузлів мережі почергово.

Отже, кожний LSR у мережі має повну інформацію щодо її топології, усіх шляхів та їх вартостей.

2.2. Алгоритм імітації роботи мережі

Попередній етап. Нехай здійснюємо прогін h_v моделі. Перед початком процесу імітації програма у випадку наявності випадкових величин, заданих для подій мережі, генерує їх. Отже, маємо повністю статичний розклад подій.

Встановлюємо для майбутніх пакетів затримку за замовчуванням, що дорівнює нулю: $delay_{r,d} = 0, r = 1..R, d = 1..D$. Аналогічно кількість модельного часу, що минув для кожного вузла та каналу на початок моделювання, встановлюємо нульовою: $delayN_{i,d} = 0, i = 1..N, d = 1..D$, $delayK_{j,d} = 0, j = 1..K, d = 1..D$.

Опишемо довільну d -ту ітерацію (d -та одиниця модельного часу за порядком).

Обираємо серед ще не оброблених подій ті, що на цей момент потрібно запустити в обробку (моментом запуску яких є цей момент або момент, що знаходиться у минулому відносно поточного моменту часу d, для якого подія не була оброблена, оскільки не було можливості прокласти LSP). Для цих подій намагаємося побудувати LSP та зарезервувати ресурси за допомогою протоколу RSVP. Якщо це вдається, то подія стає "активною".

Побудова LSP та резервування ресурсів відбуваються так. За допомогою алгоритму SPF (shortest path first) [6] шукаємо найкоротший за критерієм вартості шлях між вузлом-відправником та вузлом-отримувачем (для зменшення часу обчислень використовуємо заздалегідь побудовану перед попереднім етапом) топологію мережі. За допомогою протоколу RSVP (Resource Reservation Protocol – протокол резервування ресурсів) намагаємося зарезервувати ресурси уздовж знайденого шляху та "розвісити" мітки, сформувати в LSR таблиці LIB (Label Information Base – інформаційна база міток). Якщо резервування відбулося успішно, то LSP вважаємо таким, що прокладено.

Для множини активних подій генеруємо множину пакетів (у кількості r_d штук), які повинні з’явитися у мережі в момент часу, що дорівнює поточній одиниці модельного часу (d), відповідно до інтенсивності генерування, заданої для цієї події. Для кожного зі згенерованих пакетів встановлюємо нульову затримку: $P_{r_d} = 0$. Додаємо ці пакети до множини пакетів, що вже є у мережі на цей момент. Намагаємося помістити ці пакети до вхідних буферів пограничних LSR згідно з розкладом подій.

Серед множини вузлів $N = \{N_i\}, i = 1..n$ та каналів $K = \{K_j\}, j = 1..k$ обираємо елемент з найменшим значенням модельного часу, що минув для цього мережевого елемента: $\min\{delayN_{i,d}, delayK_{j,d}\}, i = 1..N, j = 1..K, d = 1..D$. Це може бути або вузол, або канал.

У випадку (а) обрання вузла переміщуємо пакет з вхідного буфера даного вузла у вихідний буфер цього вузла.

У випадку (б) обрання каналу виконуємо переміщення пакета з вихідного буфера одного вузла на вхідний буфер іншого вузла (через канал, що з’єднує цю пару вузлів).

Після вибору вузла чи канала вибираємо пакет. Пакет всередині відповідних елементів обирають з послідовним врахуванням двох критеріїв: пріоритету обслуговування (першими обираються пакети з найменшим його числовим значенням, що відповідає найвищому пріоритету), а далі – з найменшою затримкою (серед однакових пріоритетів). Отже, у вузлах мережі неявно отримуються черги пакетів з однаковими пріоритетами обслуговування.

Після обрання пакета рахується зміна затримки (приріст затримки) пакета як:

$$d\,delay(P_{r_d}) = \frac{S_{r_d}}{pr_i(P_{r_d})},\ d\,delay(P_{r_d}) = \frac{S_{r_d}}{c_j(P_{r_d})},$$

де $pr_i(P_{r_d})$ – продуктивність обробки пакета комутатором, $c_j(P_{r_d})$ – пропускна спроможність каналу, через який передано пакет P_{r_d} (або продуктивність вузла у випадку обрання вузла).

У випадку (а) виконуємо такі послідовні обчислення.

Нова повна затримка для пакета на цій одиниці модельного часу:

$$delay^{new}(P_{r_d}) = \max\{delay(P_{r_d}), delayN_i(d)\} + d\,delay(P_{r_d}),$$

де $delay(P_{r_d})$ – поточна затримка пакета, $delayN_i(d)$ – поточна затримка для цього вузла, $d\,delay(P_{r_d})$ – приріст затримки пакета.

Нове значення модельного часу, що минув кожного вузла:

$$delayN_i^{new}(d) = \max\{delay(P_{r_d}), delayN_i(d)\}$$

У випадку (б) виконуємо такі послідовні обчислення.

Нова повна затримка для пакета на цій одиниці модельного часу:

$$delay^{new}(P_{r_d}) = \max\{delay(P_{r_d}), delayK_j(d)\} + d\,delay(P_{r_d}),$$

де $delayK_j(d)$ – поточна затримка для цього каналу.

Нове значення модельного часу, що минув для кожного каналу:

$$delayK_j^{new}(d) = \max\{delay(P_{r_d}), delayK_j(d)\}$$

Отже, переміщають елементи, які є найбільш "відсталими" з погляду величини затримки у модельному часі.

Зберігаємо поточні значення затримок пакетів для статистики.

Зберігаємо коефіцієнт завантаження буферів (вхідного та вихідного), коефіцієнт використання каналу, вузла.

Вибір елементів всередині циклу відбувається доти, доки існує хоча б один елемент мережі (вузол або канал), значення модельного часу, що минув, для якого не перевищує одиницю модельного часу та в вузлі чи каналі відповідно маються пакети, затримка яких не перевищує одиницю модельного часу.

Коли більше відповідних елементів знайти неможливо, переходимо на наступну одиницю модельного часу; для цього від значення модельного часу для всіх елементів мережі (вузлів, каналів) і пакетів, у яких затримка більша за одиницю модельного часу, віднімається одиниця модельного часу. Для них отримуємо позитивне число, що є меншим за одиницю модельного часу, воно відповідає затримці, яка перевищила одиницю модельного часу на попередній ітерації. Залишок затримки переносимо на наступну ітерацію.

Елементи мережі (вузли, канали), значення модельного часу яких дорівнює нулю, та пакети, затримка яких дорівнює нулю, переводимо на наступну ітерацію з такою самою затримкою.

Пакети, затримка яких не перевищувала одиниці модельного часу, але була більша за нуль, при переході на наступну ітерацію з наступною одиницею модельного часу мають затримку, встановлену у нуль.

Ітерації закінчуються при досягненні кінцевого значення тривалості імітації.

Обчислюємо повну затримку пакета: якщо пакет був доставлений, то беремо його поточну затримку на цій одиниці модельного часу, додаємо до цього числа кількість одиниць модельного часу, що пройшли від моменту створення пакета до моменту часу, коли він був доставлений.

Слід зазначити, що коли обираються пакети всередині вузлів та каналів, то не обираються пакети, що були доставлені чи відкинуті.

Якщо пакет відкинуто, то це може трапитися через:

–неможливість прийому вхідним буфером комутатора (немає місця у буфері),

–неможливість прийому вихідним буфером комутатора (немає місця у буфері).

Якщо відкидання відбулося, пакет позначають як відкинутий, створюють новий пакет у вузлі-відправнику, ідентичний цьому, з позначкою “пакет повторно згенерований”), та всім елементам шляху (зворотного) від місця його відкидання до відправника збільшують затримку (значення модельного часу) на розмір службового пакета (який сигналізує відправника про повторну відправку), розділений на пропускну спроможність елемента.

Знову створеному пакету встановлюємо затримку на цій ітерації, що дорівнює сумі затримки відкинутого пакета та зміни затримки службового пакета на зворотному шляху.

Слід зазначити, що у процедурі вибору беруть участь лише ті елементи (вузли та канали), в яких наявні пакети, затримка яких не перевищує одиницю модельного часу (тобто пакети, "активні" впродовж цієї одиниці модельного часу).

Висновки

Отже, у роботі наведено рекурсивний алгоритм побудови повної топології мережі з усіма шляхами, що використовується потім при побудові концептуальної моделі імітаційного алгоритму, призначеного для моделювання роботи мереж з технологією MPLS, програмна реалізація якого дасть змогу досліджувати завантаженість мережевих елементів (виявлення "вузьких" місць у мережі), аналізувати та оптимізувати характеристики мереж (середньої затримки, варіації затримки, відсотка відкинутих пакетів), а також досліджувати поведінку трафіка різних класів. Програмний комплекс може стати зручним засобом для проектувальників комп'ютерних мереж з технологією MPLS.

1. Гольдштейн А.Б., Гольдшейн Б.С. Технология и протоколы MPLS. – СПб.: БХВ – Санкт – Петербург, 2005. – 304 с. 2. Олвейн В. Структура и реализация современной технологии MPLS. : Пер. с англ. – М.: Издательский дом "Вильямс", 2004. – 480 с. 3.Томашевський В.М. Моделювання систем. – К.: Видавнича група BHV, 2005. – 352 с.: іл. 4.Шеннон Р. Имитационное моделирование систем – искусство и наука. М.: Мир, 1978.-422 с. 5. Кельтон В., Лоу А. Имитационное моделирование. Классика CS. 3-е изд. – СПб.: Питер, К.: Мздательская группа BHV, 2004. – 847 с. 6. Томас М. Томас II. Структура и реализация сетей на основе протокола OSPF, 2-е изд.: Пер. с англ. – М.: Издательский дом "Вильямс", 2004. – 816 с.

УДК 004.93'14

Р. Мельник, Ю. Каличак
Національний університет "Львівська політехніка",
кафедра програмного забезпечення

ЕКСТРАКЦІЯ ОЗНАК ЗОБРАЖЕНЬ ПРИ ФРАГМЕНТАЦІЇ ІНТЕНСИВНОСТІ

Розроблено метод отримання розподілених ознак зображення на основі фрагментації його інтенсивності. За допомогою методу обчислюються ознаки інтенсивності, координати пікселів зображення та змішані ознаки. Продемонстровано експериментальні результати залежностей вибраних ознак для тестових зображень.

Ключові слова: візуальний образ, зображення, інтенсивність, об'єм інтенсивності, фрагментація.

The method for distributed features of visual patterns extraction is considered. Based on presented methods the features of intensity, pixels coordinates and mixed features are calculated. Some experimental results of using selected method for image features extraction are presented.

Keywords: visual pattern, image, intensity, intensity volume, fragmentation.

Вступ

Системи знаходження зображень за їх вмістом (Content-based image retrieval − CBIR) [1] працюють у два етапи: індексування та пошук. На етапі індексування кожний образ у базі даних