

УДК 044.724.4(045)

Павленкова Є.О,
Логвінова І.В.

КОНСТРУЮВАННЯ ТРАФІКУ В КОМП'ЮТЕРНІЙ МЕРЕЖІ НА ОСНОВІ ПРОТОКОЛУ БАГАТОШЛЯХОВОЇ МАРШРУТИЗАЦІЇ

Національний технічний університет України «КПІ ім. І.Сікорського»

janepavlenkova@gmail.comirene.lohvinova@gmail.com

Зроблено аналіз наявних протоколів багатошляхової маршрутизації, які є наближеними до оптимальних. Було визначено області використання кожного з них та обрано протокол для внутрішньодоменної маршрутизації

Ключові слова: мережа, маршрут, вузол, трафік, управління трафіком, багатошляхова маршрутизація.

Вступ

Комп'ютерні мережі одна з найбільш актуальних галузей у сфері ІТ. Основна задача будь-якої мережі це передача різного роду інформації від джерела до адресата по встановленим маршрутам. З передачею даних пов'язана ціла наука – управління трафіком, яка включає такі поняття як: контроль перенавантажень, протоколи маршрутизації і розробка трафіку Traffic Engineering (TE). Головна ціль TE полягає у передачі трафіку через дану мережу настільки ефективно, надійно і швидко, наскільки це можливо.

У традиційних схемах маршрутизації весь трафік спрямовується через єдиний «оптимальний» маршрут і це призводить до заторів, сповільнення передачі даних, невдоволення користувачів якістю передачі даних, отже, і до збитків для власників мережі. Тому, зараз багато досліджень спрямовано на розробку стратегій багатошляхової маршрутизації, які є більш ефективними ніж традиційні, тому що вони розподіляють трафік серед декількох “хороших” маршрутів, замість того, щоб посилати весь трафік через єдиний “найкращий” маршрут, і це в свою чергу зменшує затори.

Тож основною задачею є: вирішити яким чином розподіляти трафік між різними шляхами. На сьогодні існує досить багато протоколів для розподілу трафіку, проте через те, що потрібно враховувати

прості, але жорсткі функціональні можливості маршрутизації та переадресації, оптимальні рішення для управління ресурсами та контролю динаміки трафіку, як і раніше, ще не отримані, тому протоколи постійно оптимізуються, а також шукаються нові алгоритми розподілу.

У роботі проведений аналіз відомих протоколів багатошляхової маршрутизації, які є наближеними до оптимальних: OSPF з ECMP, DEFT, PEFT, MATE. Мета огляду полягає у визначенні області використання кожного з розглянутих протоколів та виборі протоколу для внутрішньодоменної маршрутизації.

Аналіз досліджень та публікацій

Проведений аналіз досліджень та публікацій показав, що протоколи MATE, PEFT, DEFT, OSPF з ECMP є найбільш наближеними до оптимальних.

MATE: MPLS Adaptive Traffic Engineering

MATE це багатошляховий адаптивний механізм управління трафіком, призначений для комутованих мереж, наприклад для MPLS мереж. MATE вимірює та аналізує навантаження на шляхах, використовуючи ці вимірювання проводить адаптивне балансування навантаження між кількома шляхами, що в свою чергу призводить до виконання його основної мети - мінімізації або запобігання перевантаження мережі.

MATE працює таким чином, що проміжні вузли не виконують управління трафіком, лише вхідні та вихідні вузли приймають у цьому участь. Принцип роботи MATE наступний (на вхідному вузлі): 1)трафік, що надходить до вузла, обробляється функцією фільтрування та розподілу, яка забезпечує такий розподіл трафіку, при якому зменшується ймовірність прибуття даних у неправильній послідовності; 2)дані переходять до функції управління трафіком, де вони розподіляються між шляхами так, щоб урівняти навантаження на них. Ця функція бере дані про навантаження на шляхах із модуля вимірювання та аналізу; 3)розділені між шляхами дані направляються до кінцевого вузла [1].

Отже, основні властивості цього протоколу такі:

- End-to-end управління між вхідним та вихідним вузлами;
- Не потрібна інформація про вимоги трафіку;
- Рішення щодо оптимізації приймаються на основі вимірювань перевантажень шляхів;
- Мінімальне перенаправлення пакетів;
- Це частково розподілений алгоритм. Вважається, що вхідний вузол миттєво отримує інформацію про стан всієї мережі.
- Мінімізує суму затримок мережі. Якщо мережа не перевантажена, то затримки рівні затримкам розповсюдження трафіку[5].

OSPF: Open Shortest Path First

Open Shortest Path First (OSPF) - це протокол динамічної маршрутизації на основі станів каналів, він знаходить найкоротший маршрут між парою вузлів використовуючи алгоритм Дейкстри. Він є одним із широко використовуваних протоколів одношляхової маршрутизації. Але він може працювати як протокол багатошляхової маршрутизації, якщо використовується разом із схемою балансування навантажень - Equal Cost Multipath (ECMP). У цьому випадку, якщо до кін-

цевого вузла існує декілька маршрутів однакової вартості, то вони всі використовуються для передачі даних, а трафік розподіляється між ними порівну. Можна також використовувати шляхи не однакової вартості, тоді потрібно налаштувати OSPF вручну [7]. OSPF є досить популярним при TE, тому що він досить легко реалізується і швидко працює, проте через те, що згідно ECMP трафік розділяється порівну, не у багатьох випадках він надає розподіл близький до оптимального. Адже оптимальне TE, як правило, розподіляє трафік на різні частини. Таким чином оптимальне TE може бути отримано використовуючи OSPF, але за умови, що ECMP буде замінена більш складною схемою балансування навантажень [8].

У статті [3] запропоновано використовувати віртуальні канали поруч з реальними фізичними каналами, що дає можливість розподіляти трафік між однаковими маршрутами нерівномірно, і тим самим отримати розподіл більш близький до оптимального.

OSPF широко використовується у внутрішньодоменній маршрутизації або у мережах малої розмірності. Але для великих мереж проблема отримання ваг посилянь для даного трафіку потребує великих обчислень і його використання стає недоцільним[2].

DEFT: Distributed Exponentially-weighted Flow Splitting

У роботі [6] пропонується розширена версія існуючих протоколів OSPF та ISIS - Distributed Exponentially-weighted Flow Splitting (DEFT). Його відмінною властивістю є те, що він направляє трафік по не-найкоротшим маршрутам, накладаючи при цьому експоненціальні штрафи на довші шляхи. Через це він має кращі показники, зберігаючи при цьому простоту і стабільність.

Однією з причин, які ускладнюють використання OSPF, є те, що ваги посилянь та матриця трафіку не об'єднані у одну оптимізаційну формулу. DEFT вирішує цю проблему введенням такої формули, де потоки і ваги посилянь є її змін-

ними, це призводить до ефективного ітеративного двох-етапного методу.

У результаті того, що використовуються не-найкоротші шляхи і нова формула оптимізації є такі переваги DEFT [1]:

- знаходить унікальний потік трафіку для даних праматерів ваг посилянь за поліноміальний час;

- легко реалізується у якості доповнення до існуючого IGP (наприклад для OSPF);

- надає майже оптимальний потік трафіку навіть для великомасштабних мереж;

- сходиться значно швидше у процедурі оптимізації.

Отже розширення DEFT є перспективним для досягнення необхідних показників ефективності трафіку за прийнятний час, навіть у великомасштабних мережах. Реалізує майже оптимальний підхід до конструювання трафіку.

PEFT: Penalizing Exponential Flow Splitting

Покращеною версією DEFT є Penalizing Exponential Flowsplitting (PEFT), що описується у статті [4]. Як і свій попередник PEFT також направляє трафік не по найкоротшим, а по всім шляхам, накладаючи експоненційний штраф на довші шляхи. Він досягає оптимального конструювання трафіку, зберігаючи при цьому простоту hop-by-hop передачі. Його ключові відмінності від DEFT: алгоритм PEFT працює набагато швидше ніж двох-етапний метод DEFT; доведено, що PEFT досягає оптимального

TE, у той час як DEFT досягає майже оптимального TE; PEFT має path-based розділення трафіку, а DEFT - link-based. [4]. PEFT реалізує гнучке розділення трафіку між кількома шляхами, і хоча для цього потребується додаткова пам'ять і час для обчислення, вони не настільки великі щоб бути критичними.

PEFT дуже близький до оптимального конструювання трафіку в плані мінімізації максимального використання шляхів та збільшенні пропускної здатності, він добре працює і у великих мережах.

В результаті проведеного аналізу багатошляхових протоколів маршрутизації для конструювання трафіку можна зробити висновок, що у мережі, розбитій на домени, для внутрішньодоменної маршрутизації доцільно використовувати OSPF з віртуальними каналами.

Постановка задачі

Для досягнення максимальної ефективності управління трафіком необхідно здійснити розподіл даних між маршрутами в межах домену, а також забезпечувати надійність передачі даних. Згідно обраного протоколу, треба знайти декілька найкоротших шляхів і серед них розподілити трафік використовуючи віртуальні канали для здійснення нерівномірного розподілу (як у статті [3]).

Виклад основного матеріалу

Нехай дана мережа розміром 50 вузлів, її було розбито на три домени як на рис. 1.

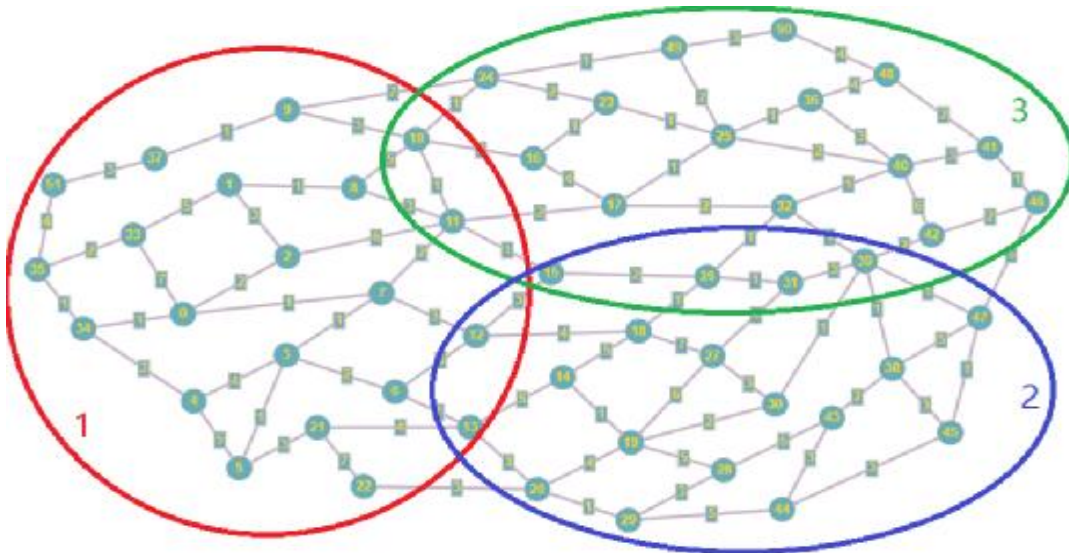


Рис. 1. Задана мережа, розбита на домени

Покажемо використання OSPF з віртуальними каналами на прикладі домену номер № 3 (рисунок 2).

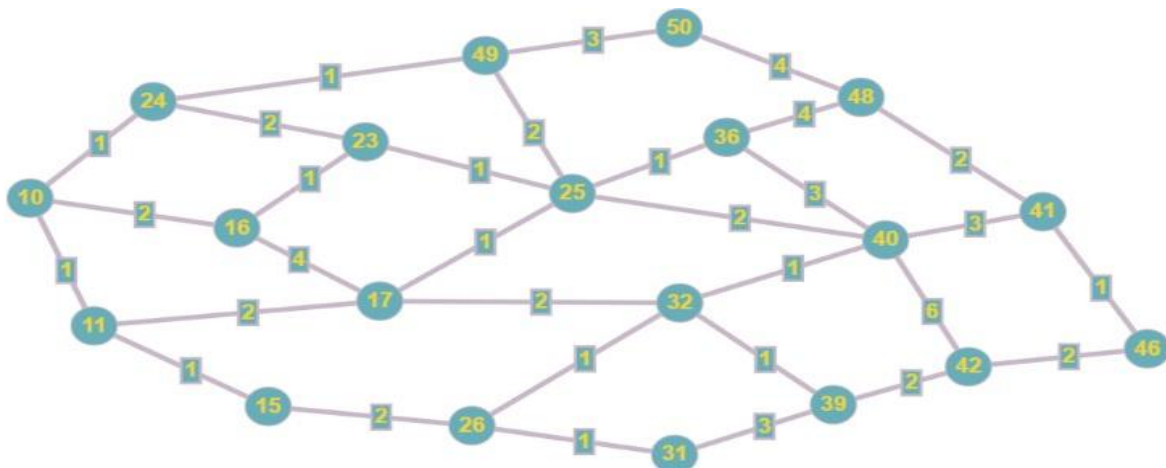


Рис. 2. Домен №3

У межах цього домену треба надіслати повідомлення (трафік) від вузла 10 до вузла 41. Трафік між 10 і 41 існує багато можливих шляхів, тобто трафік можна направити через кілька шляхів. Згідно обраного протоколу, треба знайти декілька найкоротших шляхів і серед них розподілити трафік використовуючи віртуальні канали для здійснення нерівномірного розподілу (як у статті [3]).

Крок 1. Знайти найкоротші шляхи.

Множина найкоротших маршрутів визначається так:

1) знаходяться усі дійсно найкоротші маршрути;

2) знаходяться усі маршрути, які є більшими за найкоротші на прийнятну кількість одиниць (у даному прикладі встановимо її у розмірі 2 од.).

У таблиці 1 наведена множина знайдених шляхів.

Таблиця 1. Знайдені маршрути

№	Шлях	Вартість
1	10-24-49-50-48-41	11

2	10-24-49-25-36-40-41	11
3	10-24-23-25-40-41	9
4	10-11-17-32-40-41	9
5	10-11-17-32-39-42-46-41	11
6	10-11-15-26-32-39-42-46-41	11
7	10-16-23-25-36-40-41	11

Серед знайдених шляхів обираються ті, між якими буде розділятися трафік. Для підвищення надійності пересилання даних краще використовувати маршрути, що не перетинаються, тоді при виході з ладу одного з шляхів дані через інші шляхи надійдуть до вузла-призначення і буде можливість відновити ціле повідомлення.

Максимальним числом шляхів, що не перетинаються, є $\min[a;b]$, де a – кількість ребер, що виходять з вершини-джерела, b – кількість ребер, що входять у вершину-призначення. З вузла 10 виходить три ребра, у вершину 41 входить три ребра, отже, потрібно обрати три шляхи, які не перетинаються.

Так як через вузол 16 проходить лише один шлях (№), то він автоматично обирається, і відкидаються шляхи, що перетинаються з ним: № 2, 3, 4.

Після цього лишився один шлях, що проходить через вузол 24 - №1 (тепер він також автоматично обирається), і два шляхи, що проходять через вузол 11: № 5 і №6. Отже трафік можна розділити між маршрутами № 1, 7, 5 або між № 1, 7, 6. Щоб обрати між маршрутами № 5 і № 6 порахуємо їх надійність.

Крок 2. Порахувати надійність.

Для шляхів № 5 і № 6 порахуємо надійність. Нехай надійність вузлів у цих маршрутах задана і наведена у таблиці 2.

Таблиця 2. Надійність вузлів

Вузол	Надійність	Вузол	Надійність
10	0,98	32	0,96
11	0,99	39	0,99
15	0,99	42	0,97
17	0,94	46	0,97
26	0,99	41	0,99

Для обчислення імовірності втрати даних $P(A_i)$ для маршрутів №5 і №6 будемо застосовувати формулу добутку незалежних

$$\text{подій: } P_k = 1 - \prod_{i=1}^n P(p_i), i = \overline{1, n}$$

де P_k – ймовірність втрати пакету при передачі по k -му маршруту, p_i – надійність i -го вузла.

Імовірність втрати даних для шляху № 5 (10-11-17-32-39-42-46-41):

$$P_5 = 1 - (0.95 * 0.99 * 0.94 * 0.96 * 0.99 * 0.97 * 0.97 * 0.99) = 1 - 0.807 = 0.193$$

Імовірність втрати даних для шляху №6 (10-11-15-26-32-39-42-46-41):

$$P_7 = 1 - (0.98 * 0.99 * 0.99 * 0.99 * 0.96 * 0.99 * 0.97 * 0.97 * 0.99) = 1 - 0.842 = 0.158$$

Отже шлях № 6 є надійнішим, так як ймовірність втрати пакету на ньому менша.

Обрані шляхи, для розподілу трафіку:

№7: 10-16-23-25-36-40-41

№6: 10-11-15-26-32-39-42-46-41

№1: 10-24-49-50-48-41

Крок 3. Розподіл трафіку

Отже, тепер потрібно передати G кількість трафіка по n маршрутах, таким чином, щоб отриманий розподіл трафіка $\tilde{g}_1, \tilde{g}_2, \dots, \tilde{g}_n$ був якомога ближче до бажаного розподілу g_1, g_2, \dots, g_n . Зрозуміло, що при використанні схеми ЕСМР трафік буде поділено порівну між усіма шляхами, щоб цього уникнути у статті [3] пропонується на кожен реальний шлях накласти $e_i - 1$ віртуальних шляхів, тобто в сумі буде e_i паралельних шляхів замість кожного реального, при цьому кількість трафіка, що буде передаватись по i -му маршруту ($i = \overline{1, n}$) розраховується по формулі:

$$\tilde{g}_i = \frac{G e_i}{E + n}, \text{ де } E = \sum_{i=1}^n e_i$$

Задача полягає у знаходженні таких цілих чисел e_i ($i = \overline{1, n}$), які мінімізують

$$\text{похибку: } U = \max_{i=1..n} \left\{ \begin{matrix} \tilde{g}_i - g_i \\ g_i \end{matrix} \right\}$$

Очевидно, що якщо кількість віртуальних каналів, які можна накласти на реальні, є необмеженою, то при $E + n = G$ задача стає тривіальною. Але на практиці таке буває вкрай рідко, зазвичай є обмеження: $E + n \leq Q$, де межа Q залежить від технічних характеристик пристроїв на яких розгорнута мережа, наприклад, для деяких Cisco NX-OS Softwares її максимально допустиме значення 12. Окреслену вище задачу можна представити у вигляді задачі цілочисельного лінійного програмування (таблиця 3) і розв'язати відповідними методами (додатково вводяться параметри r -число порядку 10^{-5} і M - число порядку 10^5).

Таблиця 3. Постановка ЗЦЛП

Змінні:	$e_i, i = \overline{1, n}$
	$y_j, j = \overline{1, Q}$
	a
Параметри:	1) $g_i, i = \overline{1, n}$
	2) $G = \sum_{i=1}^n g_i$
	3) r
	4) M
Цільова функція:	$a + r \sum_{i=1}^n e_i \rightarrow \min$
Обмеження:	1) $\sum_{j=1}^Q y_j = 1$
	2) $\sum_{i=1}^n e_i = \sum_{j=1}^Q j y_j$
	3) $\frac{G e_i}{g_i} \leq (a + 1)j + M(1 - y_j),$ $i = \overline{1, n}, j = \overline{1, Q}$
	4) $\frac{G e_i}{(E + n)g_i} - 1 \leq a, i = \overline{1, n}$
	5) $e_i \geq 0, e_i \in Z, i = \overline{1, n}$
	6) $y_j \in \{0; 1\}, j = \overline{1, Q}$

Саме змінні y_j допомагають знайти оптимальне число $E + n$. Перейдемо від загального вигляду задачі до конкретного нашого прикладу. Нехай нам потрібно передати $G = 100$ одиниць трафіку: по маршруту №7 $g_1 = 20$ од., по маршруту №8 $g_3 = 50$ од., по маршруту №1 $g_3 = 30$. І нехай задана верхня межа сумарної кількості шляхів $Q = 9, E + n \leq 9$. Підставимо у задачу відомі значення (таблиця 4).

Таблиця 4. ЗЦЛП з відомими значеннями

Змінні:	e_1, e_2, e_3
	$y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8,$ y_9
	a
Параметри:	1) $g_1 = 20, g_2 = 30, g_3 = 50$
	2) $G = 100$
	3) $r = 10^{-5}$
	4) $M = 10^5$
Цільова функція:	$a + 10^{-5}(e_1 + e_2 + e_3) \rightarrow \min$
Обмеження:	1) $\sum_{j=1}^9 y_j = 1$
	2) $\sum_{i=1}^3 e_i = \sum_{j=1}^9 j y_j$
	3) $\frac{100 e_i}{g_i} \leq (a + 1)j + 10^5(1 - y_j),$ $i = \overline{1, 3}, j = \overline{1, 9}$
	4) $\frac{100 e_i}{(E + 3)g_i} - 1 \leq a, i = \overline{1, 3}$
	5) $e_i \geq 0, e_i \in Z, i = \overline{1, 3}$
	6) $y_j \in \{0; 1\}, j = \overline{1, 9}$

Розв'язавши задачу було знайдено такі значення змінних:

$$e_1 = 2, e_2 = 3, e_3 = 4, a = 0.11, y_9 = 1, y_2 = 0,$$

$$y_3 = 0, y_4 = 0, y_5 = 0, y_6 = 0, y_7 = 0, y_8 = 0.$$

Тобто оптимальна кількість $E + n = 9$, на шляху №6 потрібно накласти 1 віртуальний шлях, на шляху №7 – 2, на шляху №1 – 3. Таким чином, через шляху №7 пройде $\tilde{g}_1 = 22.22$ одиниць трафіка, через шляху №8 – $\tilde{g}_2 = 33.33$, через шляху №1 – $\tilde{g}_3 = 44.44$. Як видно отриманий розподіл є досить близьким до бажаного і все, що потрібно для цього зробити – це налаштувати 6 віртуальних шляхів над існуючими реальними.

Висновки

В даній статті, на основі аналізу існуючих відомих протоколів багатошляхової маршрутизації, було запропоновано рішення для конструювання трафіку всередині домену, а саме протокол OSPF з віртуальними каналами. Такий підхід дозволить збільшити пропускну здатність каналу передачі даних та підвищить надійність конструювання трафіку.

Список літератури

1. *Ditixa Vyas, Ritesh Patel, Amit Ganatra*. Survey of Distributed Multipath Routing Protocols for Traffic Management// International Journal of Computer Applications. – 2013. – №14(17). – P.43-47.
2. *Ke Xu, Meng Shen, Fan Li, Tong Li*. Achieving Optimal Traffic Engineering Using a Generalized Routing Framework// Transactions on Parallel and Distributed Systems. – 2015.–№27(1). – Pp.51-65.
3. *Kriszti'an N'emeth , Attila K'or'osi , G'abor R'etv'ari*. Optimal OSPF Traffic Engineering using Legacy Equal Cost Multipath Load Balancing//IFIP Networking Conference. – 2013. – P.2-6.
4. *Dahai Xu, Mung Chiang, Jennifer Rexford*. Link-State Routing With Hop-by-Hop Forwarding Can Achieve Optimal Traffic Engineering//IEEE/ACM transactions on networking. – 2011. – №19(6) – P.1717-1730.

5. *Jiayue He, Ma'ayan Bresler, Mung Chiang, Jennifer Rexford*. Towards Robust Multi-Layer Traffic Engineering: Optimization of Congestion Control and Routing//IEEE Journal on selected areas in communications. – 2007. - №25(5) –P.869-875.

6. *Dahai Xu, Mung Chiang, Jennifer Rexford*. DEFT:Distributed Exponentially-weighted Flow Splitting// INFOCOM 2007, IEEE International Conference on Computer Communications. 2007. – P.25-34.

7. *Гликман Ю. К.* Разработка метода и алгоритмов многопутевой маршрутизации для повышения отказоустойчивости IP сетей: Дис. ... канд. техн. наук:05.13.01 – Спб., 2005– 122с.

8. Cisco Systems [Интернет-ресурс]/Web-сайт:www.cisco.com; Режим доступу: <http://www.cisco.com/en/US/docs/switches/datacenter/sw/6x/nxos/unicast/command/reference/13cmd.pdf>, вільний.

Статтю подано до редакції 18.04.2017