

ДОДАТОК А

Графічний матеріал кваліфікаційної роботи

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки
Факультет комп'ютерної інженерії та управління
Кафедра ЕОМ

Магістерська кваліфікаційна робота

Методи балансування інформаційними потоками комп'ютерних мереж

Виконав: ст. гр. КСМм-23-1 Соколовський С.О.

Керівник: доц. каф. ЕОМ Янковський О.А.

ВСТУП

Інтернет спочатку був розроблений як «мережа з двома з'єднаннями», щоб гарантувати, що жоден збій не спричинить втрату зв'язку будь-якої частини мережі, яка не має збою. По суті, будь-яка пара джерело-приймач повинна підтримувати більше одного шляху, щоб забезпечити надійність і відмовостійкість мережі.

Зростанням Інтернет-трафіку і поява нових його типів, а також необхідність задоволення запитів на надання все більш різноманітного діапазону мережеских послуг, сприяє збільшенню основної мережевої інфраструктури з впровадженням нових хостів і, як наслідок, нових комутаторів та маршрутизаторів, необхідних для розширення мережі. Загалом із розширенням мережі створюються нові зв'язки, які створюють надлишковість шляхів. Таким чином, маючи надлишкові канали зв'язку, їх можна використовувати не лише в ситуаціях недоступності, але й у повсякденному балансуванні навантаження між ними. Такий тип використання надлишковості дає змогу збільшити пропускну здатність трафіку даних шляхом поєднання пропускну здатності двох або більше мережеских шляхів.

МЕТА КВАЛІФІКАЦІЙНОЇ РОБОТИ

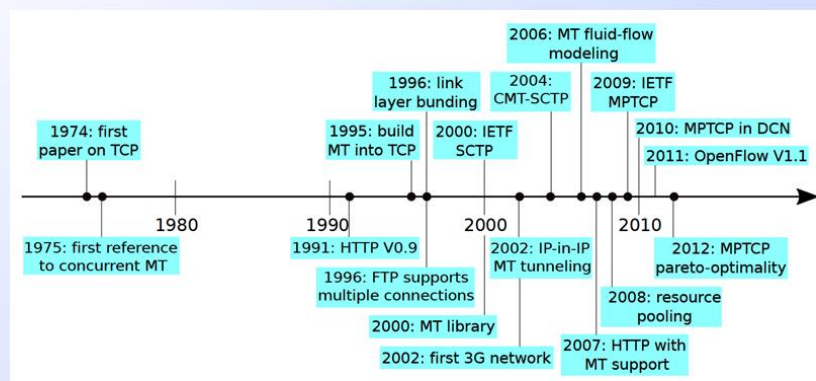
Незважаючи на те, що найчастіше в комп'ютерній мережі існує багато ресурсів, вони не використовуються повністю. Причина полягає в тому, що за замовчуванням звичайний стек TCP/IP використовує лише один «найкращий» шлях відповідно до певних показників маршрутизації, інші доступні шляхи залишаються в режимі очікування лише для цілей резервного копіювання та відновлення.

У невеликих мережах зазвичай використовуються пристрої Ethernet, які застосовують такі рішення, як протокол Spanning Tree Protocol (STP), щоб пересилати пакети за одним шляхом без петель комутації. Однак це запобігає використанню неактивних каналів, які можуть зменшити перевантаження та збільшити загальну пропускну здатність мережі.

Таким чином, метою магістерської кваліфікаційної роботи є поліпшення продуктивності комп'ютерних мереж завдяки застосуванню різноманітних методів балансування навантаження на мережевих каналах.

3

ВІХИ В ЕВОЛЮЦІЇ БАГАТОШЛЯХОВОЇ ПЕРЕДАЧІ



4

ТИПИ БАГАТОШЛЯХОВОЇ ПЕРЕДАЧІ

- на канальному рівні
- на мережевому рівні
- на транспортному рівні
- на прикладному рівні

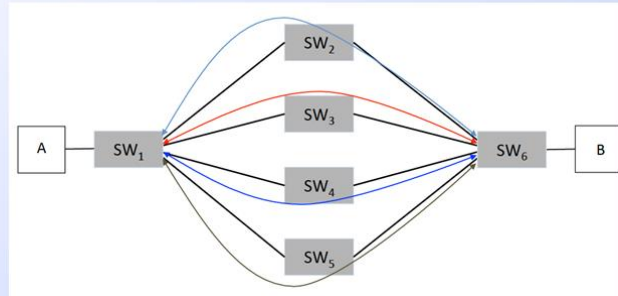
5

ОСНОВНІ ПЕРЕВАГИ БАГАТОШЛЯХОВОЇ ПЕРЕДАЧІ

- надійність
- агрегація смуги пропускання
- справедливість і оптимальність за Парето
- безпека

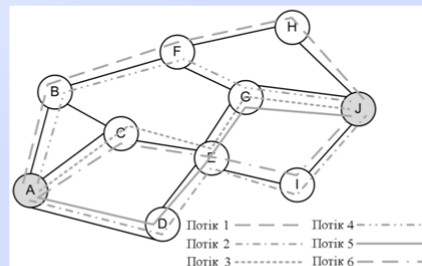
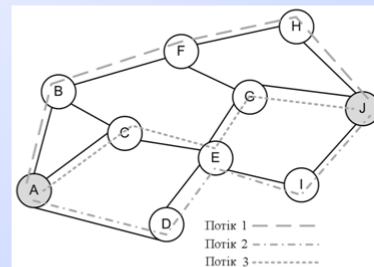
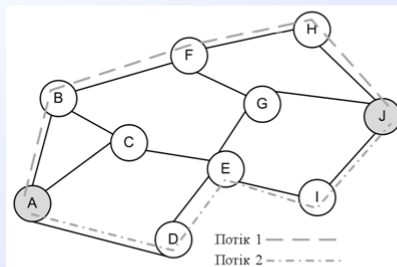
6

ОСНОВНА ІДЕЯ БАЛАНСУВАННЯ НАВАНТАЖЕННЯМ



7

СЦЕНАРІЇ ОБЧИСЛЕННЯ ШЛЯХУ



8

АЛГОРИТМ СЕТА

Псевдокод алгоритму Сета

```

begin
  src = source host
  dst = destination host
  while true do
    connected_nodes = proc_topology()
    G = networkx.graph(connected_nodes)
    paths = networkx.all_shortest(G, src, dst, dijkstra)
    for i from 1 to number(paths) do
      t1 = returns_data_tx_rx(OpenDaylight_restconf, paths[i])
      wait 2s
      t2 = returns_data_tx_rx(OpenDaylight_restconf, paths[i])
      path_cost[i] = t2 - t1
    end
    best_path = returns_index(min(path_cost))
    configure_flow(paths[best_path], source, destination)
  end
end

```

9

АЛГОРИТМ БАЛАНСУВАННЯ НАВАНТАЖЕННЯМ

Псевдокод модифікованого алгоритму

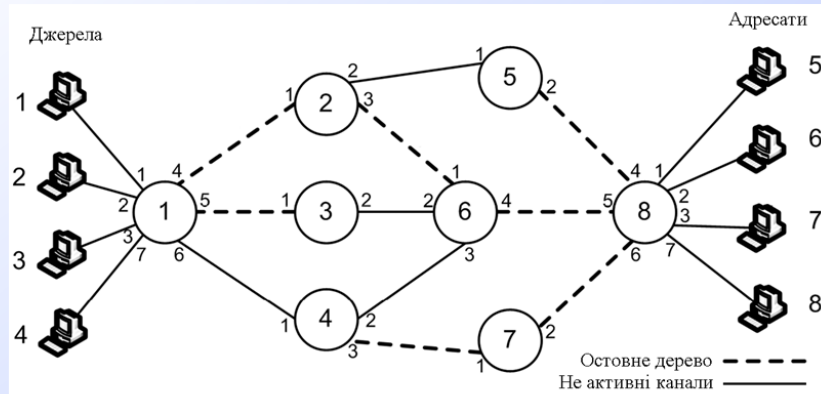
```

begin
  src = source switch
  dst = destination switch
  no_flows = number of flows
  path_capacity = 10,000,000
  current[flow] = 0
  while true do
    connected_nodes = proc_topology()
    G = networkx.graph(connected_nodes)
    for flow from 1 to no_flows do
      paths = networkx.edge_disjoint(G, src, dst, edmonds_karp)
      for i from 1 to number(paths) do
        paths_cost[i] =
          calculate_occupation(OpenDaylight_restconf, paths[i])
      end
      best_path = returns_index(min(paths_cost))
      if current[flow] = 0 then
        configure_flow(paths[best_path], src, dst, flow)
        current[flow] = best_path
      end
      else if paths_cost[current[flow]] ≥ condition1 ×
        path_capacity & path_cost[best_path] ≤ condition2 ×
        path_cost[current[flow]] then
        configure_flow(paths[best_path], src, dst, flow)
        current[flow] = best_path
      end
    end
  end
end

```

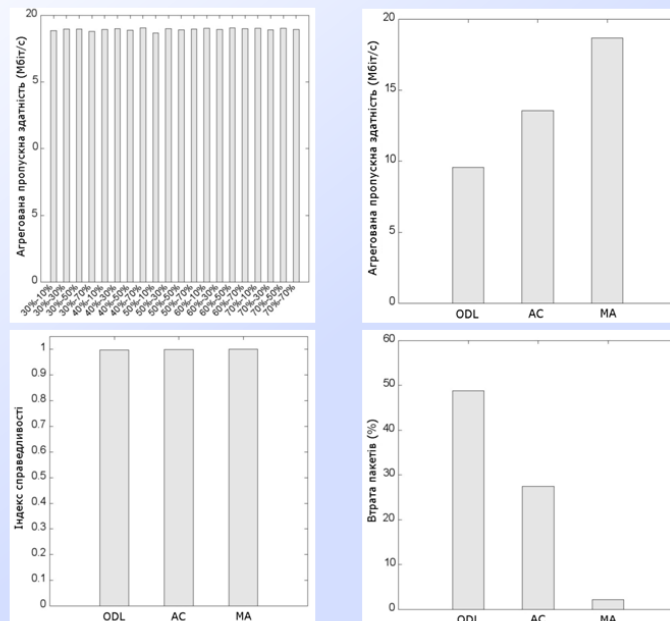
10

ТОПОЛОГІЯ ЕКСПЕРИМЕНТАЛЬНОЇ МЕРЕЖІ



11

ЕКСПЕРИМЕНТ З ДВОМА ПОТОКАМИ



12

ПУБЛІКАЦІЯ ЗА ТЕМОЮ КВАЛІФІКАЦІЙНОЇ РОБОТИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ОБОРОНИ
АЗЕРБАЙДЖАНСЬКОЇ РЕСПУБЛІКИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
"ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ"
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ
УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНИКИ
НАЦІОНАЛЬНИЙ АЕРОКОСМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ М. С. ЖУКОВСЬКОГО
"ХАРКІВСЬКИЙ АВІАЦІЙНИЙ ІНСТИТУТ"
УНІВЕРСИТЕТ МІСТА ЖИЛІНА

СУЧАСНІ НАПРЯМИ РОЗВИТКУ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЗАСОБІВ УПРАВЛІННЯ

Тези доповідей чотирнадцятої міжнародної
науково-технічної конференції
25 – 26 квітня 2024 року
Том 1: секції 1, 2

Баку – Харків – Жиліна – 2024

Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління

АНАЛІЗ А.ПОРТМІВ БАГАТОШЛЯХОВОЇ МАРШРУТІЗАЦІЇ ДЛЯ ПРОГРАМНО-ВИЗНАЧЕНИХ МЕРЕЖ

Бойршинов Є.В., Соколовський С.О., Яковський О.А.
Харківський національний університет радіоелектроніки, Харків, Україна

На сьогодні, із швидким розвитком Інтернету масштаби мережі та додатків різко збільшуються, що призводить до дедалі складнішої структури та функцій. Мережа, заснована на традиційній архітектурі TCP/IP, стикається з багатьма проблемами, особливо такі її компоненти, як маршрутизатори, які беруть на себе занадто багато функцій [1]. Достовірність і ефективність пересилання даних стикається з серйозними проблемами.

Зараз для вирішення цих проблем широко використовуються програмно-визначені мережі (SDN). Це дозволяє підвищити ефективність мережевого обладнання, підвищити зερваність і безпеку мережі. За допомогою SDN можна керувати мережею на програмному рівні і тим самим поліпшити контроль над передачею даних у мережі [2].

Враховуючи зростаючу популярність технології SDN, постає проблема розробки методів управління трафіком відповідно до принципів архітектури SDN. Одним із найбільш перспективних способів управління трафіком у великих комп'ютерних мережах є використання багатошляхової маршрутизації. Багатошляхова маршрутизація – це метод пошуку шляху, який використовує для передачі даних більш ніж один доступний шлях в мережі [3].

Метою доповіді є аналіз існуючих алгоритмів багатошляхової маршрутизації для програмно-визначених мереж (SDN) з акцентом на QoS, балансуванні навантаження та уникненні перевантаження.

Доповідь спрямована на визначення переваг та недоліків існуючих алгоритмів, що надає змогу визначити найбільш перспективні напрями їх розвитку.

В доповіді наведено результати аналізу різноманітних алгоритмів багатошляхової маршрутизації, що дозволяє вважати їх ефективними у порівнянні з класичною одношляховою маршрутизацією.

Список літератури

1. Yilan Liu, Yun Pan, Minx Yang, Wensong Wang, Chi Fang and Ruijun Jiang, "The multi-path routing problem in the Software Defined Network", 2015 11th International Conference on Natural Computation (ICNC), Zhangjiajie, 2015, pp. 250-254. DOI: <https://doi.org/10.1199/ICNC.2015.2377999>
2. Akka, M. (2016). A Preliminary Survey on the Security of Software-Defined Networks. Proceedings of International Conference on Advanced Technology & Sciences (ICATS'16), 468-473. DOI: <https://dergipark.org.tr/en/pub/iatas/issue/215619/210588>
3. Syarifuddin, S., Azzu, M. F., & Sunardi, F. D. S. (2021). Comparison Analysis of Multipath Routing Implementation in Software Defined Network. Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control, 6(2). DOI: <https://doi.org/10.22719/kinetik.v6i2.1238>

111

15

ПУБЛІКАЦІЯ ЗА ТЕМОЮ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Інститут систем управління
МНО Азербайджанської республіки
Національний технічний університет
"Харківський політехнічний інститут"
Харківський національний
університет радіоелектроніки
Національний аерокосмічний університет
імені М. С. Жуковського
"Харківський авіаційний інститут"
Університет технологій і гуманітарних наук
(м. Бельсько-Бяла, Польща)

ПРОБЛЕМИ ІНФОРМАТИЗАЦІЇ

Тези доповідей дванадцятої міжнародної
науково-технічної конференції
21 – 22 листопада 2024 року
Том 1: секції 1, 2, 3

Баку – Харків – Бельсько-Бяла – 2024

Problems of informatization: the twelfth international scientific and technical conference

КОМПЛЕКСНА БОРОТЬБА З ПЕРЕВАНТАЖЕННЯМ В МЕРЕЖАХ

Соколовський С.О., Афанасьєв М.В., Ключа М.І., Яковський О.А.
Харківський національний університет радіоелектроніки, Харків, Україна

Одним із ключових механізмів запобігання мережевим перевантаженням є TCP (Transmission Control Protocol). TCP використовує алгоритм AIMD (Additive Increase, Multiplicative Decrease), який знижує швидкість передачі даних у разі виведення втрат пакетів.

Серед новітніх підходів виділяють TCP Cubic та BBR (Bottleneck Bandwidth and Round-trip propagation time), що адаптовані до сучасних мереж. TCP Cubic завдяки неспійному приросту швидкості передачі швидше відповідає зростаючій після зниження пропускової здатності, тоді як BBR орієнтується на пропуску здатність каналу, що дозволяє оптимально використовувати доступні ресурси [1, 2].

Для подальшого зниження затримок та уникнення переповернення черг у маршрутизаторах широко використовуються алгоритми AQM, що активно керують чергами пакетів. Одним з них є популярний алгоритм Random Early Detection (RED), випадково відкидає пакети при високому навантаженні черги, сприяючи уникненню її переповернення. Інноваційні алгоритми такі, як CoDel (Controlled Delay) і PIE (Proportional Integral Controller Enhanced), забезпечують динамічне регулювання, враховуючи рівень затримки в черзі. CoDel забезпечує контроль затримки без необхідності налаштування порогів довжини черги, а PIE пропонує контроль відкидання пакетів, що знижує вплив затримок та сприяє стабільності трафіку [3, 4].

Метою доповіді є аналіз сучасних методів для зниження перевантаження в комп'ютерних мережах, зокрема різновидів протоколу TCP та алгоритмів активного управління чергами (Active Queue Management, AQM), які сприяють стабільності передачі даних та зменшують затримки.

Таким чином, комплексне поєднання технологій, таких як сучасні версії TCP та алгоритми AQM, забезпечує ефективну боротьбу з перевантаженнями, покращуючи продуктивність і надійність комп'ютерних мереж.

Список літератури

1. Ha, S., Rhee, I., & Xu, L. (2008). CUBIC: A new TCP-friendly high-speed TCP variant. *ACM SIGOPS Operating Systems Review*, 42(5), 64-74.
2. Cardwell, N., Chang, Y., Gan, C. S., Yegorov, S. H., & Jacobson, V. (2016). BBR: Congestion-based congestion control. *Communications of the ACM*, 60(2), 58-66.
3. Nichols, K., & Jacobson, V. (2012). Controlling Queue Delay. *Communications of the ACM*, 55(7), 42-50.
4. Pan, R., Natarajan, P., Baker, F., & Prabh, G. (2013). PIE: A lightweight control scheme to address the bufferbloat problem. *IEEE 14th International Conference on High Performance Switching and Routing (HPSR)*, 148-155.

78

16

ВИСНОВКИ

Мета кваліфікаційної роботи полягала в поліпшенні продуктивності комп'ютерних мереж завдяки балансуванню навантажень на мережевих каналах зв'язку.

В кваліфікаційній роботі запропоновано метод балансування навантажень між шляхами, який включає функцію управління перемиканням потоків на недовантажені канали.

Запропонований алгоритм обчислює кілька шляхів із непересічними зв'язками, які мають найменшу кількість переходів між джерелом і одержувачем. Крім того, алгоритм має функцію «контролю перемикання», яка перевіряє, чи перевищує поточне завантаження шляху відсоток його пропускної здатності, і чи потенційний новий шлях, має завантаженість принаймні на порогове значення менше, ніж значення поточного шляху.

Алгоритм перевершує за показниками стандартний режим роботи ODL та алгоритм Seta. Можна зробити висновок, що модифікований алгоритм балансування навантажень є хорошим рішенням для уникнення перевантажень і таким чином, сприяє збільшенню сукупної пропускної здатності мережі.