

МОДЕЛЮВАННЯ МАРШРУТИЗАЦІЇ ГРЯНЯМИ У БЕЗДРОТОВИХ ДИНАМІЧНИХ МЕРЕЖАХ ЗА ДОПОМОГОЮ ВІРТУАЛЬНИХ ГРАФІВ

Національний авіаційний університет

mchrissten@gmail.com

Запропоновано загальну постановку задачі маршрутизації пакетів гроями у неплоских мережевих графах. Проаналізовано загальний процес маршрутизації гроями у віртуальному плоскому графі. Запропоновано алгоритм обчислення віртуального шляху у плоских віртуальних графах

Ключові слова: географічна маршрутизація, бездротові динамічні мережі, віртуальний граф, алгоритм маршрутизації

Вступ

Якщо два вузли у мережі передачі даних безпосередньо не з'єднані лінією зв'язку, їхні повідомлення один до одного повинні пересилатися проміжними вузлами. Пошук шляху між двома вузлами, між якими йде обмін повідомленнями у мережі передачі даних є фундаментальним завданням, що називається маршрутизацією. У традиційній комп'ютерній мережі є вузли під назвою маршрутизатори, які виконують пересилання інформації у комп'ютерних мережах. Додатки на комп'ютерах з'єднуються з серверами і повідомлення перенаправляються маршрутизаторами до місць призначення. На відміну від традиційних комп'ютерних мереж, бездротові динамічні мережі не розрізняють вузли, сервери і маршрутизатори. Також, дані мережі відрізняються і від бездротових мереж з базовими станціями, таких як мобільні телефонні системи, в яких повідомлення ретранслюються на базові станції. У бездротових динамічних мережах вузли є не тільки комп'ютерами з додатками, але і функціонують як маршрутизатори для пересилання повідомлень для інших вузлів, які не входять до радіусу дії бездротової передачі один одному. Вузли у таких мережах утворюють самоорганізовану мережу без централізованого адміністрування або підтримки [6]. Таким чином бездротові

динамічні мережі є розподіленими системами.

Було запропоновано велику кількість динамічних протоколів маршрутизації, починаючи від модифікації та оптимізації традиційних підходів до маршрутизації для статичних мереж до інноваційних методів для динамічних мереж, які використовують інформацію про географічне місцезнаходження вузлів.

Маршрутизація, що базується на місцезнаходженні, також відома як геометрична маршрутизація або географічна, була запропонована для вирішення проблеми масштабування. Замість використання топологічної інформації, протоколи даного виду маршрутизації використовують географічну інформацію про розташування вузлів для маршрутизації пакетів. Дані протоколи припускають, що вузли знають своє власне географічне положення (наприклад, від системи глобального позиціонування [7]) і вихідному вузлу відоме розташування вузла призначення.

Одним із видів географічної маршрутизації є маршрутизація гроями, яка гарантує доставку пакетів у статичних зв'язаних плоских графах [1-3]. Вона застосовується на плоскому графі, відповідно пакет передається по границях граней, які перетинаються відрізком, що з'єднує вихідний вузол і вузол призначення.

Існують дві проблеми у застосуванні маршрутизації гранями на неплоскому графі мережі [7-9]: мережевий граф може не мати з'єднаного загального плоского підграфу, і, навіть якщо такий підграф існує, його отримання може бути складним завданням.

Аналіз попередніх досліджень

Маршрутизація гранями, яка запропонована у [1], була першим геометричним алгоритмом маршрутизації, що гарантував доставку пакетів без лавинного розповсюдження повідомлень. Згодом було запропоновано декілька варіантів протоколів даного виду маршрутизації [2, 3, 7, 10]. Маршрутизація гранями застосовується на плоскому підграфі мережевого графа. Плоский граф ділить площину на грані. Відрізок, що з'єднує вихідний вузол і вузол призначення, перетинає деякі грані. Згідно даної маршрутизації, пакет пересилається по межах цих граней. Деякий протокол геометричної маршрутизації гранями надає набір правил для кожного вузла, за допомогою яких приймається рішення, куди відправити пакет, використовуючи тільки локальну інформацію про своїх сусідів і інформацію у заголовку пакета.

Типовий протокол маршрутизації гранями функціонує наступним чином [2]. Починаючи, пакет пересилається по межі першої грані, що перетинається відрізком від початкової точки до місця призначення. Першою дугою обходу грані є перша дуга за годинниковою стрілкою навколо початкової точки, після з'єднуючого відрізка. Після обходу дуги (u, v) , наступною дугою обходу грані є перша дуга після (v, u) за годинниковою стрілкою навколо v . Таким чином, пакет проходить дугам межі грані у напрямку проти годинникової стрілки. Даний обхід використовує правило правої руки. Коли обхід досягає дуги, яка перетинається відрізком, що з'єднує початкову і цільову точки, у точці, яка є ближчою до місця призначення, ніж початкова, то дана точка стає новою точкою відліку і обхід перемикається на наступну грань. Дана процедура повто-

рюється, доки не буде досягнуто місце призначення.

Згідно іншого варіанту [1], пакет проходить всю межу грані і запам'ятовує точки перетину з відрізком між початковою точкою і місцем призначення, які знаходяться ближче до місця призначення. Після того, як пакет повертається до вихідної точки цієї грані, він пересилається по її межі до збереженої точки перетину.

Відповідно, дана точка стає новою відправною точкою і обхід перемикається на наступну грань. Дана процедура продовжується, поки цільова точка не буде досягнута. Рис.1 показує приклад обходу, який обрахований за допомогою протоколу маршрутизації гранями [2].

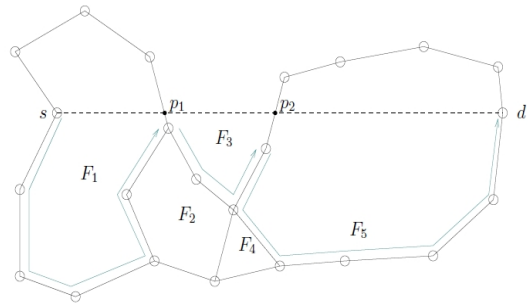


Рис. 1. Приклад шляху, який проходить пакет згідно маршрутизації гранями

У даному прикладі пакет передається від вузла s до вузла d . Спочатку обхід відбувається вздовж межі F_1 , поки він не досягне p_1 , точки перетину межі грані F_1 і відрізка sd , у якій обхід перемикається до наступної грані F_3 . Далі пакет рухається по межах граней F_3 і F_5 .

Хоча основна ідея маршрутизації гранями проста, декілька варіантів маршрутизації гранями у деяких плоских графах [4] можуть не доставити пакет. Наприклад, якщо алгоритм обходу використовує вузли, які ближче до точки призначення, ніж початковий вузол, в якості нових відправних точок, тоді він не визначить нову початкову точку на деякій грані, як у прикладі: грань F_1 на рис. 2 (а). У даному прикладі, вузол s є початковим вузлом і вузол d - вузол призначення. Інші два вузли v_1 і v_2 на межі грані F_1 знаходяться далі від вузла d , ніж від s . Алго-

ритм обходу зазнає невдачі у випадку використання поточного вузлу в якості нової точки відліку, замість точки перетину дуги грані з відрізком, що з'єднує вихідний вузол з кінцевим.

В іншому випадку пакет може переміщатися вздовж послідовності граней не досягаючи пункту призначення. Це показано на прикладі на рис. 2 (б), де даний алгоритм буде пересилати пакет по $(s1; s2)$, $(s2; s3)$ і $(s3; s1)$.

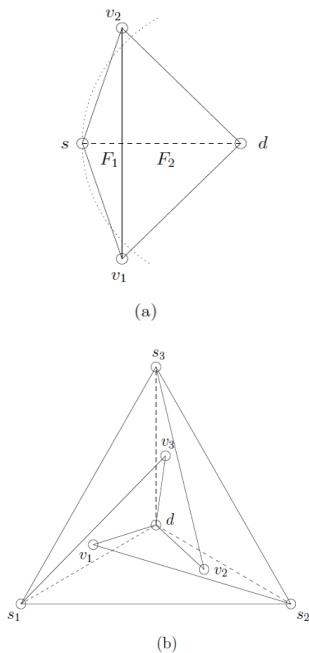


Рис. 2. Приклади невдалої роботи алгоритмів обходу

Маршрутизація гранями гарантує доставку пакета в статичних з'єднаних плоских графах [2, 5]. Її було об'єднано з "жадібним" підходом маршрутизації, щоб зробити протокол більш ефективним. Одним з таких протоколів є *GPSR (Greedy Perimeter Stateless Routing)* [3]. Основна стратегія маршрутизації в *GPSR* – «жадібна» маршрутизація. Вузол посилає пакет до одного з сусідів у мережі, що ближче до місця призначення, за умови, що сусід ближче до місця призначення. Однак, коли вузол ближче до місця призначення, ніж будь-який з його сусідів, маршрутизація гранями використовується як механізм відновлення.

Як і протокол *GPSR*, протокол *GOAFR+* [10] також є поєднанням жадібної маршрутизації та маршрутизації гра-

нями. *GOAFR+* використовує більш складні методи, щоб об'єднати маршрутизацію гранями з жадібною маршрутизацією.

Постановка задачі

Протоколи маршрутизації гранями мають два обмеження: вони потребують окремо побудованого плоского підграфу мережі для маршрутизації, і припускається, що даний плоский підграф залишається незмінним протягом процесу маршрутизації [7-9]. Отримання зв'язаного загального плоского підграфу розподіленим чином може бути складно у реальних мережах. Експериментальні результати [8-9] показують, що більшість існуючих алгоритмів мають проблеми у реальних бездротових мережах через перешкоди у зоні дії і неточну інформацію про місцезнаходження. Проблеми з отриманням графа маршрутизації можуть привести до порушення роботи протоколів маршрутизації гранями. Більш того, експерименти показали, що зв'язки у деяких бездротових динамічних мережах часто нестабільні, навіть якщо вузли є стаціонарними [9, 1]. Як результат, дані протоколи не можуть бути практично застосовані у реальних бездротових динамічних мережах.

У даній статті описано загальний підхід до застосування маршрутизації гранями до неплоского графу без вилучення плоского підграфу. Даний метод полягає у отриманні (віртуального) плоского графу з будь-якого графа, шляхом заміни кожного перетину дуг на віртуальний вузол.

Неплоский мережевий граф може розглядатися як віртуальний плоский граф. Тобто додається віртуальний вузол у кожній точці, де перетинаються два або більше ребер, і відповідно у цих віртуальних точках розділяються ребра. Таким чином, отримується віртуальний плоский граф, який складається з оригінальних вузлів мережі і віртуальних вузлів. Якщо вихідний граф з'єднаний так, що він є віртуальним плоским графом, і якщо застосувати маршрутизацію гранями у даному віртуальному графі, то можливо визначити шлях до місця призначення. Даний

шлях має назву віртуальний шлях, оскільки він може містити віртуальні вузли. Віртуальний вузол не може отримати або відправити пакет. Кун та ін. [4] і Лілліс та ін. [5] зберігали таблиці маршрутизації у реальних вузлах для того, щоб зробити можливим відправку пакетів до та від віртуальних вузлів. У нашому підході додаткова інформація маршрутизації не повинна зберігатися у реальних вузлах. Тобто відбувається обчислення реального шляху у мережевому графіку, який слідує по віртуальному шляху. Такий протокол моделює маршрутизацію гранями у віртуальному плоскому графі.

Щоб формально визначити, що означає для реального шляху слідувати віртуальному, введемо деякі позначення та визначення. Для позначення вузлів у віртуальному шляху, які можуть бути як віртуальними так і реальними, використаємо грецькі літери α , β , і т.п., а для позначення реальних вузлів використаємо малі літери. Кожне ребро у віртуальному графі може бути усією дугою мережевого графіку або її частиною. Дуга у віртуальному графі є віртуальною дугою. Початковий вузол або точка і кінцевий вузол або точка використовуються для позначення двох кінцевих точок реальної чи віртуальної дуги.

Враховуючи віртуальний шлях, у якому останній вузол є реальним вузлом, визначимо функцію, яка відображає даний шлях на послідовність реальних вузлів.

Визначення. Нехай $\pi = v_0, v_1, \dots, v_k = u$, при $k \geq 1$ є віртуальним шляхом, де останнім вузлом v_k є реальний вузол. Нехай $F(\pi) = u_0, u_1, \dots, u_{k-1}, u$, де u_i є початковим вузлом реальному ребра, що містить віртуальне ребро (v_i, v_{i+1}) при $i = 0, \dots, k-1$. Вважаємо, що реальний шлях P слідує віртуальному шляху π , якщо $F(\pi)$ є підпослідовність P .

Протокол, який моделює маршрутизацію гранями у всіх віртуальних плоских графах, гарантує доставку повідомлень у статичних зв'язаних графах.

Віртуальний плоский граф зв'язного графа також зв'язаний, тому що додавання віртуальних вузлів і поділ ребер у віртуальному вузлі не роз'єднує будь-який шлях в оригінальному графі. Оскільки маршрутизація гранями гарантує доставку пакетів у статичному зв'язному плоскому графі, віртуальний шлях отримується, застосовуючи маршрутизацію гранями у віртуальному плоскому графі статичного зв'язного графа. За визначенням, протокол, який моделює маршрутизацію гранями обчислює реальний шлях, який слідує за віртуальним шляхом і, отже, приводить до вузла призначення.

Алгоритм обчислення віртуального шляху

Для моделювання маршрутизації гранями у віртуальному графі потрібно розглянути два питання. Перше полягає в обчисленні віртуального шляху, а друге у знаходженні реального шляху у мережі, який слідує віртуальному. Всі обчислення мають бути виконані розподілено і з використанням тільки локальної інформації, що доступна у кожному вузлі, і інформацію, яку протокол ставить у заголовок пакету.

Віртуальний шлях буде обчислюватися одне ребро за один раз. Одним із способів зробити це – нехай початковий вузол реальному ребра, що містить поточне віртуальне ребро визначає наступне віртуальне ребро. Нехай поточне віртуальне ребро (α, β) , наступне віртуальне ребро (β, γ) є першим віртуальним ребром після (β, α) за годинниковою стрілкою навколо β (Рис. 3).

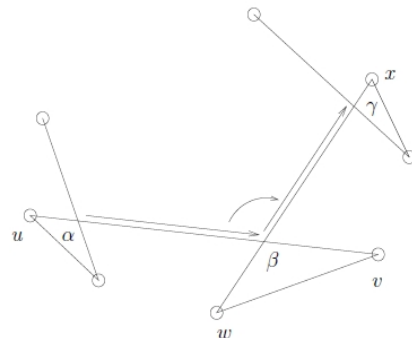


Рис. 3. Визначення наступного віртуального ребра

Припустимо, що (α, β) міститься у реальному ребрі (u, v) , і (β, γ) міститься у реальному ребрі (w, x) , яке перетинає (u, v) у β . Якщо вузол u знає своїх сусідів, (реальні) ребра яких перетинаються із (реальними) ребрами вузла u , і які (реальні) ребра перетинають їх, то можливо для u обчислити (β, γ) . Однак, якщо вузол u знає лише своїх сусідів і реальні реальні ребра, які перетинаються із його власними ребрами, то вузол u не може обов'язково визначити кінцеву точку γ наступного віртуального ребра.

З цією обмеженою інформацією віртуальний шлях може бути визначений, якщо розподілити обчислення іншим способом. На кожному кроці обходу вважаємо α початковою точкою поточного віртуального ребра і (u, v) реальне ребро, яке її містить; вузол u визначає кінцеву точку β поточного віртуального ребра, яка також є початковою точкою наступного віртуального ребра, а (w, x) реальне ребро, яке містить її. Щоб знайти реальний шлях, яким слідує віртуальне ребро, вузол u обчислює реальний шлях до w . У даному розподіленому обчисленні ребер віртуального шляху, два кінця віртуального ребра можуть бути визначені при різних реальних вузлах.

Далі наводиться алгоритм, який виконує обчислення згідно даної процедури. Вхідними параметрами даного алгоритму є (u, v) і α , де (u, v) є реальним ребром, яке містить поточне віртуальне ребро і α є початковою точкою поточного віртуального ребра. α може бути або вузлом u або внутрішньою точкою на ребрі (u, v) . Вихідними даними алгоритму є β , кінцева точка поточного віртуального ребра, яка також є початковою точкою наступного віртуального ребра; (w, x) - ребро, яке містить наступне віртуальне ребро; і π - шлях від вузла u до вузла w , по якому пакет може бути направлений до w для продовження обходу. На рис. 4 зображено геометричні залежності між вхідними та вихідними параметрами даного алгоритму.

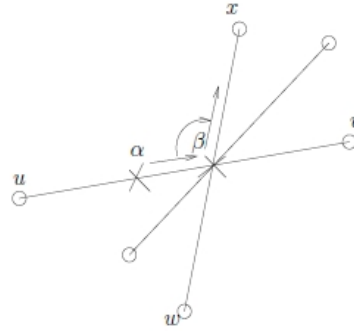


Рис. 4. Вхідні та вихідні параметри алгоритму

Алгоритм($(u, v), \alpha, \beta, (w, x), \pi$)

* Вхідні дані: ребро (u, v) і точка α на (u, v) ;
* Вихідні дані: β , перший вузол на (u, v) після α у віртуальному графі; (w, x) – дуга, що містить перше віртуальне ребро після $(\beta; u)$ у годинниковому напрямку навколо β ; і π , шлях від u до w у мережевому графі.

begin

if there is an edge that crosses (α, v) then

* наступне віртуальне ребро є найближчим ребром перетину

let β be the crossing point on $(\alpha; v)$ that is closest to α

let (w, x) be the edge that contains the next virtual edge after $(\beta; \alpha)$ in clockwise order around β

let π be a path from u to w in network graph

else

* ребра, що перетинають (α, v) відсутні, наступне ребро починається з v

$\beta \leftarrow 0076$

$(w, x) \leftarrow$ the next edge after (v, u) in clockwise order around v

* $w = v$

$\pi \leftarrow u, w$

end

Якщо вхідні параметри (u, v) містять поточне віртуальне ребро вздовж границі віртуальної грані і α – точка на віртуальному ребрі, то вихідний параметр β алгоритму є кінцевою точкою поточного віртуального ребра і (w, x) – ребро що містить наступне віртуальне ребро вздовж границі цієї ж віртуальної грані.

Важливо звернути увагу на те, що частина умовного оператора алгоритму визначає які вихідні параметри мають бути без явного визначення як обчислювати їх, так як це залежатиме від доступної інформації на вузлі u і яка модель графу мережі.

Після того як кінцева точка β поточного віртуального ребра визначена, вузол u перевіряє чи містить (α, β) будь-яку точку, що знаходиться ближче ніж початкова точка поточної віртуальної грані. Якщо така точка знаходиться, процес обходу перемикається до наступної віртуальної грані; в іншому випадку, вузол u пересилає пакет до вузла w , і процес обходу продовжується.

Висновок

Було розроблено метод, який розширює і узагальнює підхід до маршрутизації гранями в результаті чого, вона може бути застосована безпосередньо на загальних неплоских мережевих графах, без окремої побудови плоского підграфу маршрутизації. Відповідно до даного підходу, було проаналізовано загальний процес маршрутизації гранями у віртуальному плоскому графі і запропоновано алгоритм обчислення віртуального шляху у плоских віртуальних графах.

Список літератури

1. E. Kranakis, H. Singh, J. Urrutia. Compass routing on geometric networks // In Proc. 11th Canadian Conference on Computational Geometry. – Vancouver, August 1999. – P. 51-54.
2. P. Bose, P. Morin, I. Stojmenovic, J. Urrutia. Routing with guaranteed delivery in ad hoc wireless networks // Wireless Networks, №7 (6). – 2001. – P. 609-616.
3. B. Karp, H. T. Kung. GPSR: greedy perimeter stateless routing for wireless networks // In MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking. – 2000. – P. 243-254.
4. F. Kuhn, R. Wattenhofer, A. Zollinger. Ad-hoc networks beyond unit disk graphs // In DIALM-POMC '03: Proceedings of the 2003 joint workshop on Foundations of mobile computing. – ACM Press, 2003. – P. 69-78.
5. K. M. Lillis, S. V. Pemmaraju, and I. Pirwani. Topology control and geographic routing in realistic wireless networks // In Evangelos Kranakis and Jaroslav Opatrny, editors, ADHOC-NOW, volume 4686 of Lecture Notes in Computer Science. – Springer, 2007. – P. 15-31.
6. J. P. Hubaux, Th. Gross, J. Y. Le Boudec, and M. Vetterli. Towards self-organized mobile ad hoc networks: the Terminodes project // IEEE Communications Magazine, №31(1). – 2001. – P. 118-124.
7. F. Kuhn, R. Wattenhofer, A. Zollinger. Asymptotically optimal geometric mobile ad-hoc routing // In DIALM '02: Proceedings of the 6th international workshop on Discrete algorithms and methods for mobile computing and communications. – New York, ACM Press, 2002. – P. 24-33.
8. Y. Kim, R. Govindan, Brad Karp, and Scott Shenker. On the pitfalls of geographic face routing // In DIALM-POMC '05: Proceedings of the 2005 joint workshop on Foundations of mobile computing. – New York, ACM Press, 2005. – P. 34-43.
9. K. Seada, A. Helmy, R. Govindan. On the effect of localization errors on geographic face routing in sensor networks // In IPSN '04: Proceedings of the 3rd international symposium on Information processing in sensor networks. – New York, ACM, 2004. – P. 71-80.
10. B. Leong, S. Mitra, B. Liskov. Path vector face routing: Geographic routing with local face information // In ICNP '05: Proceedings of the 13TH IEEE International Conference on Network Protocols (ICNP'05). – Washington, IEEE Computer Society, 2005. – P. 147-158.

Статтю подано до редакції 15.04.2014