

Разработка процедуры двунаправленного поиска для решения задачи маршрутизации в транспортных программно-конфигурируемых сетях

Волков А.С.*, Баскаков А.Е.**

Национальный исследовательский университет «Московский институт электронной техники», площадь Шокина, 1, Москва, Зеленоград, 124498, Россия

**e-mail: leshvol@mail.ru*

***e-mail: 9999924816@ya.ru*

Статья поступила 12.04.2021

Аннотация

В работе рассмотрена модернизация алгоритма поиска множества путей передачи данных с построением вспомогательного графа, возможная к применению в транспортных программно-конфигурируемых сетях связи, путем добавления возможности двунаправленного поиска при работе алгоритма.

Программно-конфигурируемые сети связи, как правило, функционируют на базе установленных протоколов, описывающих процедуры взаимодействия ПКС-коммутаторов и контроллера, например, протокол OpenFlow. Так, контроллер имеет актуальную информацию о сети связи, ее топологии и отдельных устройства, а значит, сеть связи может быть представлена в виде неориентированного взвешенного графа, что сводит задачу маршрутизации к задаче поиска множества путей на графе.

Существующий алгоритм поиска множества путей на основе построения вспомогательного графа имеет высокое значение временной сложности при наличии в

исходном графе большого количества вершин, то есть не обеспечивает должной степени масштабируемости системы в целом. Предложенное решение основано на принципе работы вышеуказанного алгоритма с разбиением исходного графа на подграфы, с дальнейшим построением вспомогательного графа для каждого из них, что теоретически снижает временную сложность до двух раз, ввиду отсутствия необходимости работы с графом большой размерности.

Ключевые слова: программно-конфигурируемые сети, управление ресурсами транспортной сети, двунаправленный поиск путей передачи.

Введение

Современные сети связи представляют собой сложное архитектурное решение, выполняющее ряд телекоммуникационных задач, решение которых позволяет наиболее эффективно передавать большие массивы данных через сеть. Ежегодный рост объемов передаваемого трафика делает невозможным использование сетевых технологий на протяжении продолжительного промежутка времени без изменений, поскольку вместе с ним растут и требования его обработки, к сетевым решениям и оборудованию предъявляются все более жесткие ограничения [1-2]. Одним из наиболее важных параметров передачи данных является задержка из конца в конец, представляющая собой совокупность временных величин, необходимых для обработки входящего

сетевого пакета, его передачи через физические каналы связи, проводные или беспроводные, а также время работы используемых сетевых протоколов и технологий, в частности, время, затраченное на построение различных таблиц и работу программных реализаций алгоритмов.

Для обеспечения возможности приоритезации трафика существуют известные решения, например, механизмы обеспечения качества обслуживания, но при росте интенсивности входного потока, в случае недостаточной вычислительной мощности устройств или ограниченной пропускной способности их интерфейсов, могут возникать очереди, вносящие дополнительные задержки при передаче данных, что может быть критично в ряде случаев [3-4].

Для решения вышеуказанной проблемы существуют известные сетевых технологии и методы передачи данных, например, применение принципов программно-определяемых сетей связи позволяет частично сократить задержки передачи сетевых пакетов [5]. Основой таких сетей является идея переноса вычислительных процессов и протоколов управления в отдельное устройство или массив управляющих устройств – контроллеров программно-конфигурируемой сети. В качестве основных устройств плоскости передачи данных при этом применяются программно-конфигурируемые коммутаторы, функционал которых, по сравнению с классическими коммутаторами, частично сокращен, но добавлена возможность передачи актуальных параметров на центральное устройство. Так, в ПКС применимы методы динамического отслеживания состояний отдельных сетевых устройств в подконтрольном сегменте сети связи, что

позволяет сократить общее время на решение проблемы маршрутизации пакетов данных, поскольку центральное устройство всегда имеет информацию о всех устройствах сети, доступных соединениях и их параметрах, получаемую через специализированные протоколы связи ПКС-коммутатор – ПКС-контроллер. Одним из наиболее известных примеров таких протоколов является OpenFlow, реализованный в большинстве ПКС устройств в качестве стандартно используемого. Сам протокол активно развивается на протяжении десяти лет и имеет более пяти версий реализации, не считая подверсий, отличающихся форматами передаваемых сообщений между устройствами и контроллером, а также общим количеством служебных сообщений и реализованным функционалом управления [6-7].

Типовым решением для ПКС сетей является организация на устройстве контроллера функций маршрутизации, в том числе поиска кратчайших путей передачи в подконтрольной сети связи, на основе входящего служебного сообщения. Такое решение позволяет частично решить проблему сокращения времени передачи пакета, но зачастую контроллер использует типовые алгоритмы поиска путей передачи данных, как правило без реализации принципов многопутевой маршрутизации. Таким образом, актуальной задачей, особенно в области транспортных программно-конфигурируемых сетей, является разработка методов и алгоритмов многопутевой маршрутизации, а также ее составляющих, например, алгоритмов построения путей передачи данных на центральном устройстве управления [8].

Алгоритмы поиска путей передачи данных

Для решения задачи поиска путей, сеть связи можно представить в виде взвешенного неориентированного графа. Вершинами графа при этом считаются активные сетевые устройства, а ребрами – их межсоединения, тогда граф имеет вид $G(V, E)$, где V – множество вершин, E – множество ребер. При этом каждое ребро имеет вид $E(b, d)$, где b – пропускная способность соединения, d – задержка передачи данных соединения, что в совокупности является весом этого ребра.

Существующие решения задачи поиска кратчайшего пути, как правило, не предоставляют возможности объективной оценки проблемы, поскольку их свойства заключаются в поиске одного пути за одну итерацию выполнения алгоритма. Тем не менее, существует возможность повторного вычисления кратчайшего пути, без учета уже найденного, что в совокупности позволяет найти множество путей передачи данных, но, как правило, пути при этом непересекающиеся.

Рассмотрим случай, когда в качестве исходного графа в алгоритм передается топология сети связи, имеющая в составе пять вершин (ПКС-коммутаторов) и шесть ребер, имеющих различные параметры пропускной способности и задержки передачи данных, приведенная на рис. 1., а для поиска путей передачи данных применяется алгоритм Дейкстры.

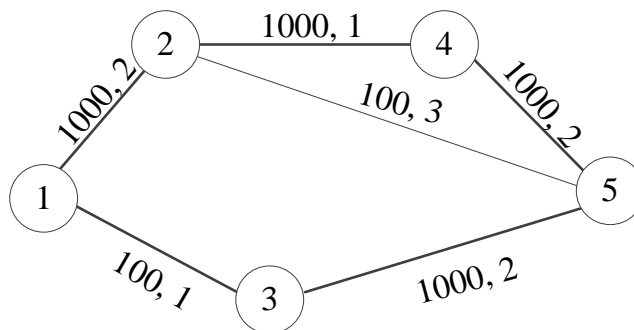


Рис.1 – Граф сети связи

Алгоритм при выполнении создаст дерево кратчайших путей от начальной вершины, до всех конечных точек на графе. При поиске кратчайшего пути из вершины 1 в вершину 5, при использовании задержки в качестве метрики кратчайший путь всегда будет построен как 1-3-5, что не является оптимальным решением задачи при высокой интенсивности входящего трафика на вершине 1. Для обхода подобной проблемы можно использовать мультиметрику, тогда оптимальным по алгоритму Дейкстры будет считаться путь 1-2-4-5, который наоборот, не удовлетворяет условиям передачи данных с минимальной задержкой [9-10].

Алгоритм поиска путей на основе вспомогательного графа

Кроме рассмотренного выше алгоритма Дейкстры, существует множество различных решений задачи поиска путей передачи данных на графах, наиболее известными из которых являются поиск в глубину и в ширину. Такие алгоритмы частично решают изначальную проблему, но, как правило, поиск осуществляется исключительно непересекающихся маршрутов, что для указанного примера оставит нерассмотренным возможность передачи данных по пути 1-5.

Решением является использование алгоритма поиска путей передачи данных с построением вспомогательного графа [11-13]. В простейшем случае, учитывается только один параметр ребра графа в качестве метрики – задержка передачи данных, для реализации более точного поиска следует определить новую метрику с использованием обоих параметров ребра. Так, для нахождения искомого параметра метрики необходимо ввести новую переменную входного подпотока данных, такую, что сумма всех входных подпотоков равна исходному входному потоку на стартовой вершине. Тогда метрика может быть представлена в виде $D_i = d_i + \frac{\lambda_i}{b_i}$, где переменные d и b описывают задержку передачи данных и пропускную способность ребра соответственно, а λ_i – входной подпоток [14-16]. Пример построения вспомогательного графа и путей передачи данных для исходной задачи, с установленными ограничениями по задержке передачи данных, не превышающей 5мс и требуемой пропускной способности 100 Мбит/с представлен на рис.2 и описывает все три возможные пути передачи данных, а именно: 1-2-5, 1-2-4-5 и 1-3-5. При этом алгоритм имеет псевдо полиномиальную сложность и может быть оценен через временную сложность, составляющую $O(qS_m D \log(S_m D))$, где q – количество путей, S_m – средняя степень вершин, D – длина максимального пути.

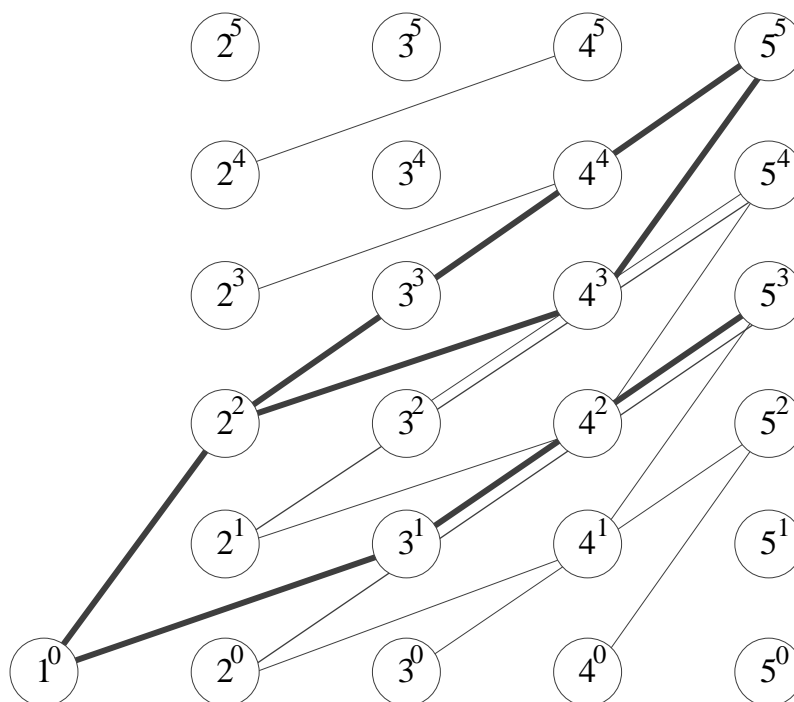


Рис.2 – Вспомогательный граф для решения задачи поиска множества путей

Поиск путей передачи указанным способом действительно имеет преимущества перед типовыми алгоритмами поиска пути, поскольку имеет меньшую временную сложность, особенно при поиске множества путей, а также может быть модернизирован для поиска путей не только между двумя вершинами, но и от одной вершины к каждой другой, в таком случае для каждой пары вершин будет найдено свое подмножество путей. Но сложность такого способа прямо зависит от длины найденного маршрута и средней степени вершин в графе, а значит, имеет низкую степень масштабируемости, при графах с большим количеством вершин и ребер между ними алгоритм может выполняться продолжительное время. Для решения этой проблемы можно применить известные способы разделения графа на подграфы, например, алгоритм разделения графа на компоненты минимальной связности, или использовать метод

двунаправленного поиска. Указанный метод описывает возможность поиска пути одновременно с двух вершин, являющихся начальной и конечной для поставленной задачи, при этом общая сложность алгоритма понизится, поскольку операции будут производиться над графами меньшей размерности. Для демонстрации принципа работы метода двунаправленного поиска в качестве исходных данных примем взвешенный неориентированный граф $G(V, E)$, состоящий из 8 вершин и ребер между ними (рис.3), необходимо найти набор путей из вершины 1 в вершину 8.

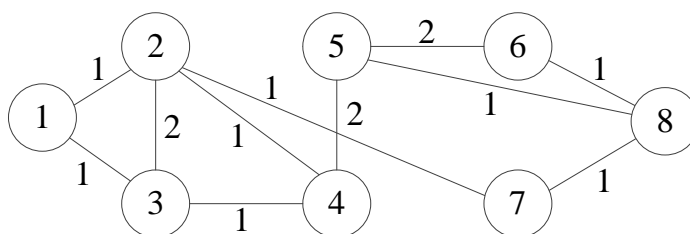


Рис.3 – Граф сети связи

Исходя из возможности начала работы с двух вершин одновременно, построим два вспомогательных графа на основе подграфов $G_1(V_1, E_1)$ и $G_2(V_2, E_2)$, где $G_1 \in G$, $G_2 \in G$, P_{ai} - порожденные пути между G_1 и G_2 , то есть пути от вершины 4-5 и 2-7. Тогда достаточным условием решения поставленной задачи является нахождение путей передачи от вершины 1 к вершинам 2 и 4 для G_1 , и путей от вершины 8 к 5 и 7 для G_2 , на рис.4 продемонстрированы построенные вспомогательные графы для G_1 , G_2 и найденные пути передачи данных.

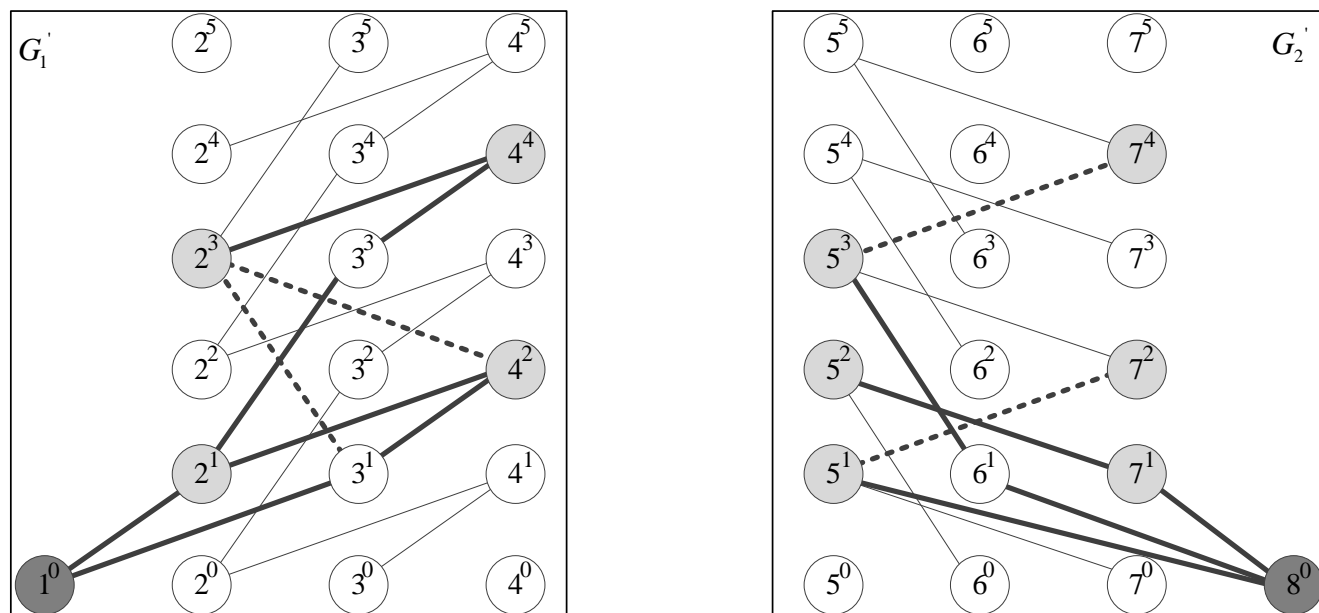


Рис.4 – Двухнаправленный поиск путей с построением вспомогательных графов

Так, в результате работы алгоритма, изначальный подграф был разбит на два подграфа, на основе которых были построены вспомогательные графы, аналогично алгоритму без применения метода двухнаправленного поиска, после чего в каждом вспомогательном графе были найдены пути передачи данных с использованием задержки передачи в качестве веса ребра. Соответственно для G_1' были найдены семь возможных путей передачи. Три из них являются путями до вершины 2: 1-2 (вес равен 1); 1-3-2 (вес равен 3); 1-3-4-2 (вес равен 3); и четыре до вершины 4: 1-3-4 (вес равен 2); 1-2-4 (вес равен 2); 1-2-3-4 (вес равен 4); 1-3-2-4 (вес равен 4). Для G_2' аналогично найдены шесть путей: 8-7, 8-5, 8-6-5, 8-7-5, 8-5-7 и 8-6-5-7. Таким образом, зная что между подграфами существуют порожденные пути 2-7 и 4-5 решением исходной задачи

является комбинации путей из множеств $P_{V_2i}^{G_1}$ (пути от вершины 1 в 2), P_{2-7} и $P_{V_7i}^{G_2}$ (пути от вершины 7 в 8), а также $P_{V_4i}^{G_1}$, P_{4-5} и $P_{V_3i}^{G_2}$.

Исходя из теории графов, а именно леммы о рукопожатиях имеем, что сумма степеней вершин произвольного графа, равна удвоенному количеству его ребер, то есть

$$\sum_{v \in V} S(v) = 2|E|, \text{ для } G(V, E) \text{ [17-19]}$$

Тогда искомое значение средней степени вершины в графе, для определения временной сложности алгоритма может быть представлено в

$$\text{виде } S_m = \frac{2|E|}{V}, \text{ а значение максимальной длины пути в общем случае выражено через}$$

$$D = V - 1. [20]$$

Представим временную сложность алгоритма поиска путей на основе вспомогательного графа с использованием вышеуказанных формул:

$$O(q \frac{2EV - EV}{V} \log(\frac{2EV - EV}{V})), \text{ тогда, для частного случая, представленного на рис.3,}$$

расчетная временная сложность поиска множества путей передачи данных составит

значение 519,23, аналогичный расчет сложности поиска путей с использованием

алгоритма Дейкстры составит 1344, что более чем в два раза превышает аналогичный

результат при использовании поиска с вспомогательным графом. В случае

использования метода двунаправленного поиска вместе с построением

вспомогательного графа временная сложность может быть выражена через

$$O(q_1 S_{m1} D_1 \log(S_{m1} D_1) + q_2 S_{m2} D_2 \log(S_{m2} D_2)), \text{ где } q_i \text{ – количество путей, } S_{mi} \text{ – средняя степень}$$

вершин, D_i – длина максимального пути для подграфов G_1 , G_2 соответственно, что эквивалентно

$$O(q_{G_1} \frac{2E_{G_1}V_{G_1} - E_{G_1}V_{G_1}}{V_{G_1}} \log(\frac{2E_{G_1}V_{G_1} - E_{G_1}V_{G_1}}{V_{G_1}}) + q_{G_2} \frac{2E_{G_2}V_{G_2} - E_{G_2}V_{G_2}}{V_{G_2}} \log(\frac{2E_{G_2}V_{G_2} - E_{G_2}V_{G_2}}{V_{G_2}})),$$

расчетная временная сложность выполнения алгоритма для исходной задачи составит 105,95, такое значение может быть получено при идеальных условиях реализации алгоритма, без учета дополнительных итерационных действий по формированию вспомогательных графов.

Реальное время работы алгоритма при этом будет зависеть от множества факторов, определяемых конкретной программной реализацией, поскольку процедуры записи, модификации, хранения информации, ровно как и добавления вершин и ребер в граф имеют свою временную и алгоритмическую сложности, разную для используемых способов хранения данных: списка смежности, списка инцидентности, матрицы смежности и т.д. Общей же тенденцией остается увеличение сложности при повышении размерности графа, аналогично типовым алгоритмам поиска путей на графе.

Заключение

В работе рассмотрена проблема, существующая в программно-конфигурируемых транспортных сетях связи, выраженная в необходимости поиска набора путей передачи данных от одного устройства к другому на установленной сетевой топологии, а также поиска набора путей передачи данных от устройства к множеству устройств с

применением различных графокомбинаторных алгоритмов поиска путей: алгоритма Дейкстры, алгоритма поиска множества путей на основе вспомогательного графа, алгоритма двунаправленного поиска множества путей на основе вспомогательного графа.

Предложено решение исходной задачи, заключающееся в модернизации алгоритма поиска множества путей на основе вспомогательного графа, а именно, организации двунаправленного поиска в указанном алгоритме. Типовые решения, как правило, основаны на принципе поиска множества непересекающихся путей, что не всегда отвечает предъявляемым требованиям к организации процесса передачи данных через сегмент сети связи. Разработанное решение осуществляет поиск множества всех возможных путей передачи данных в исходном графе от вершины к вершине или от вершины к множеству вершин, при этом временная сложность алгоритма в разы меньше аналогичного параметра у алгоритма без двунаправленного поиска.

Разработанное решение позволяет осуществлять поиск путей передачи при известной сетевой топологии и применимо к программно-конфигурируемым сетям связи, в том числе транспортным, но при наличии корректных входных данных может быть также использовано и в других областях, например для решения задач маршрутизации беспроводных сенсорных сетей или самоорганизующихся сетей, в том числе FANET, основанном на устройствах БПЛА.

Библиографический список

1. Hou A. et al. Bandwidth scheduling for big data transfer using multiple fixed node-disjoint paths // Journal of Network and Computer Applications, 2017, vol. 85, pp. 47 - 55. DOI: [10.1016/j.jnca.2016.12.011](https://doi.org/10.1016/j.jnca.2016.12.011)
2. Srinivasan S.M., Truong-Huu T., Gurusamy M. Flexible bandwidth allocation for big data transfer with deadline constraints // 2017 IEEE Symposium on Computers and Communications (ISCC), IEEE, 2017, pp. 347 - 352. DOI: [10.1109/ISCC.2017.8024554](https://doi.org/10.1109/ISCC.2017.8024554)
3. Aleksieva V., Haka A. Modified scheduler for traffic prioritization in lte networks // International Conference on Intelligent Information Technologies for Industry, Springer, Cham, 2017, pp. 228 - 238. DOI: [10.1007/978-3-319-68324-9_25](https://doi.org/10.1007/978-3-319-68324-9_25)
4. Yamada Y. et al. Feature-selection based data prioritization in mobile traffic prediction using machine learning // 2018 IEEE Global Communications Conference (GLOBECOM), IEEE, 2018. DOI: [10.1109/GLOCOM.2018.8647627](https://doi.org/10.1109/GLOCOM.2018.8647627)
5. Barakabitze A.A. et al. 5G network slicing using SDN and NFV: A survey of taxonomy, architectures and future challenges // Computer Networks, 2020, vol. 167, pp. 106984. DOI: [10.1016/j.comnet.2019.106984](https://doi.org/10.1016/j.comnet.2019.106984)
6. Alsaeedi M., Mohamad M.M., Al-Roubaiey A.A. Toward adaptive and scalable OpenFlow-SDN flow control: A survey // IEEE Access, 2019, vol. 7, pp. 107346 - 107379. DOI: [10.1109/ACCESS.2019.2932422](https://doi.org/10.1109/ACCESS.2019.2932422)

7. Priya A.V., Radhika N. Performance comparison of SDN OpenFlow controllers // International Journal of Computer Aided Engineering and Technology, 2019, vol. 11, no. 4-5, pp. 467 - 479. DOI: [10.1504/IJCAET.2019.100444](https://doi.org/10.1504/IJCAET.2019.100444)
8. Gopi D., Cheng S., Huck R. Comparative analysis of SDN and conventional networks using routing protocols // 2017 International Conference on Computer, Information and Telecommunication Systems (CITS), IEEE, 2017, pp. 108 - 112. DOI: [10.1109/CITS.2017.8035305](https://doi.org/10.1109/CITS.2017.8035305)
9. Waleed S. et al. Demonstration of single link failure recovery using Bellman Ford and Dijkstra algorithm in SDN // 2017 International Conference on Innovations in Electrical Engineering and Computational Technologies (ICIEECT), IEEE, 2017, DOI: [10.1109/ICIEECT.2017.7916533](https://doi.org/10.1109/ICIEECT.2017.7916533)
10. Wang X. Y. L. SDN load balancing method based on K-Dijkstra // International Journal of Performability Engineering, 2018, vol. 14, no. 4, pp. 709. DOI: [10.23940/ijpe.18.04.p14.709716](https://doi.org/10.23940/ijpe.18.04.p14.709716)
11. Dong K. et al. Auxiliary graph based routing, wavelength, and time-slot assignment in metro quantum optical networks with a novel node structure // Optics express, 2020, vol. 28, no. 5, pp. 5936 - 5952. DOI: [10.23919/PS.2019.8817743](https://doi.org/10.23919/PS.2019.8817743)
12. Бородин В.В., Петраков А.М., Шевцов В.А. Анализ алгоритмов маршрутизации в сети связи группировки беспилотных летательных аппаратов // Труды МАИ. 2016. № 87. URL: <http://trudymai.ru/published.php?ID=69735>

13. Бородин В.В., Петраков А.М., Шевцов В.А. Имитационная модель для исследования адаптивных сенсорных сетей // Труды МАИ. 2018. № 100. URL: <http://trudymai.ru/published.php?ID=93398>
14. Tripp-Barba C. et al. Survey on routing protocols for vehicular ad hoc networks based on multimetrics // Electronics, 2019, vol. 8, no. 10, pp. 1177. DOI:[10.3390/electronics8101177](https://doi.org/10.3390/electronics8101177)
15. Borodin V., Shevtsov V., Petrakov A., Shevgunov T. Characteristics of Updating Route Information of Networks with Variable Topology // Proceedings of the Computational Methods in Systems and Software. Springer, Cham, 2019, pp. 376 – 384. DOI:[10.1007/978-3-030-30329-7_33](https://doi.org/10.1007/978-3-030-30329-7_33)
16. Шевцов В.А., Бородин В.В., Крылов М.А. Построение совмещенной сети сотовой связи и самоорганизующейся сети с динамической структурой // Труды МАИ. 2016. № 85. URL: <http://trudymai.ru/published.php?ID=66417>
17. Моршинин А.В. Об одной задаче кластеризации графа // Вестник Омского университета. 2018. Т. 23. № 1. С. 4 - 9. DOI: [10.25513/1812-3996.2018.23\(1\).4-9](https://doi.org/10.25513/1812-3996.2018.23(1).4-9)
18. Курносоев А.Д. О нижней оценке числа связного доминирования в графах с фиксированной степенной последовательностью // Труды МФТИ. 2020. Т. 12. № 2. С. 40 - 54.
19. Баранский В.А., Сеньчонок Т.А. О максимальных графических разбиениях, ближайших к заданному графическому разбиению // Сибирские электронные математические известия. 2020. Т. 17. С. 338 - 363.

20. Dragan F.F., Köhler E., Leitert A. Line-distortion, bandwidth and path-length of a graph

// Algorithmica, 2017, vol. 77, no. 3, pp. 686 – 713. DOI:[10.1007/978-3-319-08404-6_14](https://doi.org/10.1007/978-3-319-08404-6_14)

Bidirectional search procedure development for solving the the transport software-defined network routing problem

Volkov A.S.*, Baskakov A.E.**

*National Research University of Electronic Technology,
1, sq. Shokina, Moscow, Zelenograd, 124498, Russia*

**e-mail: leshvol@mail.ru*

***e-mail: 9999924816@ya.ru*

Abstract

Modern communication networks represent a complex architectural solution that performs a number of telecommunication tasks, which solution allows the most efficient transmission of large amounts of data through the network. The annual growth in the transmitted traffic volume makes the network technologies application for a long period of time without changes impossible, since the requirements for its processing grow, and increasingly stringent restrictions are being imposed on the network solutions and equipment along with it. One of the most important parameters of data transmission is the end-to-end delay. It represents a set of time values required for an incoming network packet processing and transmitting it through the physical communication channels, as well as the operating time of the network protocols and technologies being employed, in particular, the time spent on of various tables' construction and operation of algorithms software implementations.

Thus, the solution to the above-said problem can be reduced in the first approximation to the problem of finding a set of routes on a graph describing the a communication network topology.

The existing standard solutions to the problem of searching for the paths on a graph describe, as a rule, the solution to the single route searching per one algorithm iteration, or provide the ability to search for exclusively disjoint data transmission paths, which in some cases is not sufficient for solving the original problem. Application of the routes searching algorithm with the auxiliary graph constructing, allowing find a set of paths, including partially intersecting ones, on the graph from one vertex to another, or from one vertex to set of vertices, may alter the situation. This solution disadvantage consists in critical increase of the algorithm operating time while employing graphs of large dimensionality.

This problem can be solved by applying the bidirectional search method for the above-said algorithm with the auxiliary graph constructing. Thus, the original graph can be conditionally divided into two parts, being the subgraphs from the original graph, connected by the generated paths. A mandatory condition herewith is starting and ending points finding in the constructed subgraphs, then, the auxiliary graphs construction based on them will allow reducing the time complexity by up to two times, depending on the specific software implementation of the algorithm and the methods of the graph writing: adjacency list, incidence list, adjacency matrix etc. The calculated time complexity herewith for the tested network topology containing eight nodes and the connections between them will be 519.23 and 105.95 for the algorithm based on the auxiliary graph and its bidirectional variation, respectively.

Keywords: software-defined network, transport network resource management, bidirectional transmission paths search.

References

1. Hou A. et al. Bandwidth scheduling for big data transfer using multiple fixed node-disjoint paths, *Journal of Network and Computer Applications*, 2017, vol. 85, pp. 47 - 55. DOI: [10.1016/j.jnca.2016.12.011](https://doi.org/10.1016/j.jnca.2016.12.011)
2. Srinivasan S.M., Truong-Huu T., Gurusamy M. Flexible bandwidth allocation for big data transfer with deadline constraints, *2017 IEEE Symposium on Computers and Communications (ISCC)*, IEEE, 2017, pp. 347 - 352. DOI: [10.1109/ISCC.2017.8024554](https://doi.org/10.1109/ISCC.2017.8024554)
3. Aleksieva V., Haka A. Modified scheduler for traffic prioritization in LTE networks, *International Conference on Intelligent Information Technologies for Industry*, Springer, Cham, 2017, pp. 228 - 238. DOI: [10.1007/978-3-319-68324-9_25](https://doi.org/10.1007/978-3-319-68324-9_25)
4. Yamada Y. et al. Feature-selection based data prioritization in mobile traffic prediction using machine learning, *2018 IEEE Global Communications Conference (GLOBECOM)*, IEEE, 2018. DOI: [10.1109/GLOCOM.2018.8647627](https://doi.org/10.1109/GLOCOM.2018.8647627)
5. Barakabitze A.A. et al. 5G network slicing using SDN and NFV: A survey of taxonomy, architectures and future challenges, *Computer Networks*, 2020, vol. 167, pp. 106984. DOI: [10.1016/j.comnet.2019.106984](https://doi.org/10.1016/j.comnet.2019.106984)
6. Alsaeedi M., Mohamad M.M., Al-Roubaiey A.A. Toward adaptive and scalable OpenFlow-SDN flow control: A survey, *IEEE Access*, 2019, vol. 7, pp. 107346 - 107379. DOI: [10.1109/ACCESS.2019.2932422](https://doi.org/10.1109/ACCESS.2019.2932422)

7. Priya A.V., Radhika N. Performance comparison of SDN OpenFlow controllers, *International Journal of Computer Aided Engineering and Technology*, 2019, vol. 11, no. 4-5, pp. 467 - 479. DOI: [10.1504/IJCAET.2019.100444](https://doi.org/10.1504/IJCAET.2019.100444)
8. Gopi D., Cheng S., Huck R. Comparative analysis of SDN and conventional networks using routing protocols, *2017 International Conference on Computer, Information and Telecommunication Systems (CITS)*, IEEE, 2017, pp. 108 - 112. DOI: [10.1109/CITS.2017.8035305](https://doi.org/10.1109/CITS.2017.8035305)
9. Waleed S. et al. Demonstration of single link failure recovery using Bellman Ford and Dijkstra algorithm in SDN, *2017 International Conference on Innovations in Electrical Engineering and Computational Technologies (ICIEECT)*, IEEE, 2017, DOI: [10.1109/ICIEECT.2017.7916533](https://doi.org/10.1109/ICIEECT.2017.7916533)
10. Wang X. Y. L. SDN load balancing method based on K-Dijkstra, *International Journal of Performability Engineering*, 2018, vol. 14, no. 4, pp. 709. DOI: [10.23940/ijpe.18.04.p14.709716](https://doi.org/10.23940/ijpe.18.04.p14.709716)
11. Dong K. et al. Auxiliary graph based routing, wavelength, and time-slot assignment in metro quantum optical networks with a novel node structure, *Optics express*, 2020, vol. 28, no. 5, pp. 5936 - 5952. DOI: [10.23919/PS.2019.8817743](https://doi.org/10.23919/PS.2019.8817743)
12. Borodin V.V., Petrakov A.M., Shevtsov V.A. *Trudy MAI*, 2016, no. 87. URL: <http://trudymai.ru/eng/published.php?ID=69735>
13. Borodin V.V., Petrakov A.M., Shevtsov V.A. *Trudy MAI*, 2018, no. 100. URL: <http://trudymai.ru/eng/published.php?ID=93398>

14. Tripp-Barba C. et al. Survey on routing protocols for vehicular ad hoc networks based on multimetrics, *Electronics*, 2019, vol. 8, no. 10, pp. 1177. DOI:[10.3390/electronics8101177](https://doi.org/10.3390/electronics8101177)
15. Borodin V., Shevtsov V., Petrakov A., Shevgunov T. Characteristics of Updating Route Information of Networks with Variable Topology, *Proceedings of the Computational Methods in Systems and Software*. Springer, Cham, 2019, pp. 376 – 384. DOI:[10.1007/978-3-030-30329-7_33](https://doi.org/10.1007/978-3-030-30329-7_33)
16. Shevtsov V.A., Borodin V.V., Krylov M.A. *Trudy MAI*, 2016, no. 85. URL: <http://trudymai.ru/eng/published.php?ID=66417>
17. Morshinin A.V. *Vestnik Omskogo universiteta*, 2018, vol. 23, no. 1, pp. 4 - 9. DOI: [10.25513/1812-3996.2018.23\(1\).4-9](https://doi.org/10.25513/1812-3996.2018.23(1).4-9)
18. Kurnosov A.D. *Trudy MFTI*, 2020, vol. 12, no. 2, pp. 40 - 54.
19. Baranskii V.A., Sen'chonok T.A. *Sibirskie elektronnye matematicheskie izvestiya*, 2020, vol. 17, pp. 338 - 363.
20. Dragan F.F., Köhler E., Leitert A. Line-distortion, bandwidth and path-length of a graph, *Algorithmica*, 2017, vol. 77, no. 3, pp. 686 – 713. DOI:[10.1007/978-3-319-08404-6_14](https://doi.org/10.1007/978-3-319-08404-6_14)