



МЕТОД ЖАДНОГО ПРОДВИЖЕНИЯ В ИЕРАРХИЧЕСКИХ СЕТЯХ

(Самарский университет)

Введение

Интернет вещей (Internet of Things) является частью будущей интернет-парадигмы, которая быстро меняет развитие технологий и способна перестроить различные экономические и общественные процессы, исключив из них необходимость участия человека. Возможность физических и виртуальных объектов идентифицировать и связываться друг с другом порождает огромное количество услуг в разных областях жизни. Проще говоря, это концепция подключения любых устройств к интернету и между собой. В качестве примера можно привести «Умный дом» и «Умный транспорт».

«Умный дом» позволяет пользователю избавиться от необходимости помнить, закрыл ли он дверь в квартиру, выключил ли электроприборы, система автоматически записывает все показания, поддерживает оптимальную температуру и влажность. «Умный транспорт» измеряет загруженность транспортных каналов и строит лучший маршрут. Также сейчас очень популярны автономные электромобили, которые способны без участия водителя доехать до конечной точки маршрута, найти ближайшие заправки и оптимизировать потребление электроэнергии.

Все эти устройства и объекты должны быть объединены в единую сеть. Поскольку устройства могут постоянно перемещаться, а отсутствие связи может стать критическим фактором, то наиболее оптимальным является выбор сети MANET.

Сеть MANET (Mobile Ad hoc Network) – беспроводная самоорганизующаяся сеть, состоящая из мобильных устройств (узлов), реализующих полностью децентрализованное управление при отсутствии базовых станций или опорных узлов. Каждый узел знает информацию только о ближайших соседях и не знает сеть целиком.

Узлы этих сетей являются не только оконечными пользовательскими терминалами, которые могут перемещаться в пространстве и разрывать/устанавливать соединения с соседями, но и являются ретрансляторами-маршрутизаторами, ретранслируя пакеты других абонентов и участвуя в нахождении маршрутов к ним. Такие сети могут состоять их десятков, сотен и тысяч узлов. Пример такой сети можно посмотреть на рисунке 1.

Сети MANET имеют следующие преимущества:

- Возможность передачи информации на длинные дистанции без увеличения мощности передатчика;
- Устойчивость к изменениям в инфраструктуре сети;
- Возможность быстрой реконфигурации и высокая скорость развертывания.



Когда в сети присутствует такое огромное количество узлов, очень важно настроить маршрутизацию между ними. Передача сигнала будет происходить быстрее, если правильно построить маршрут для его него. С этим может помочь метод жадного продвижения (Greedy Forwarding – жадный алгоритм).

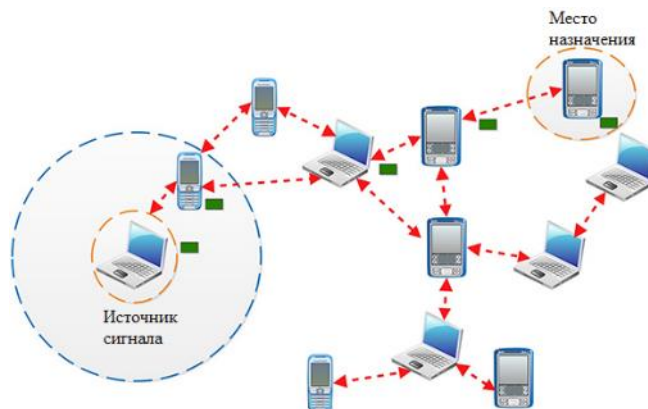


Рис. 1. Сеть MANET

Жадное продвижение в иерархических сетях

Под иерархической системой будем понимать некий набор узлов с возможностью организации связи между ними. Для этой системы можно построить иерархический граф связности, выделяя центральный узел и дочерние. Все остальные связи между узлами при таком построении игнорируются.

Обычно маршрутизация в таких сетях выстраивается с помощью таблиц маршрутизации. Основная проблема в том, что каждый отдельно взятый узел сети должен полностью знать всю сеть, которая находится под ним. При огромном количестве узлов, необходимо будет помнить информацию о миллионе узлов, что требует памяти огромных размеров.

В работе Клейнберга [1], рассмотрен масштабируемый и надежный алгоритм маршрутизации точечной топологии для специальных беспроводных сетей и сенсорных антенн (гиперболическое преобразование). Он присваивает каждому узлу сети виртуальную координату в гиперболической плоскости и выполняет жадную географическую маршрутизацию относительно этих виртуальных координат. Это гарантирует, что жадный алгоритм всегда успешно находит маршрут к месту назначения, если такой маршрут существует.

Жадная маршрутизация не гарантирует двухточечного соединения в сети, поскольку сообщения могут попасть в локальные минимумы функции расстояния до места назначения (такие локальные минимумы иногда называют «пустотами», когда в области, занятой узлами, имеется большое пустое пространство).

Теорема 2 в статье Клейнберга гласит: каждый связный конечный граф имеет жадное вложение в гиперболическую плоскость (рис 2).

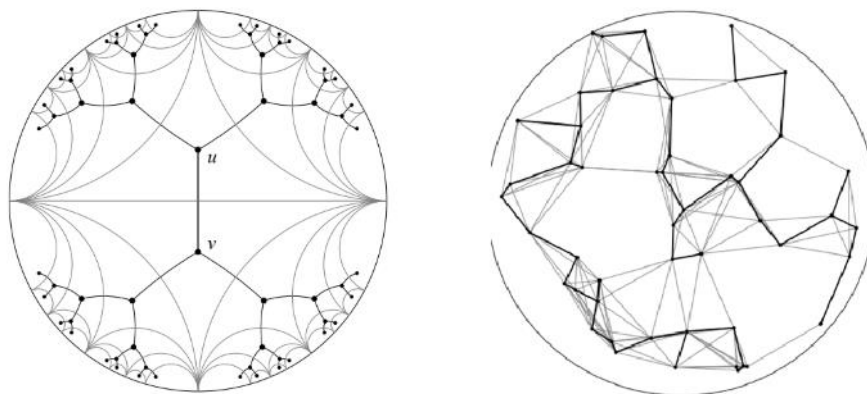


Рис. 2. Жадное вложение графа в плоскость

Чтоб использовать метод жадного продвижения в гиперболическом пространстве [3], необходимо вычислить расстояние между двумя узлами сети. Затем пакет данных будет перенесен на ближайший узел, который будет располагаться ближе всех к месту назначения.

В общем случае расстояние между двумя точками измеряется в этой метрике вдоль геодезических и равно:

$$\begin{aligned} \text{dist}(\langle x_1, y_1 \rangle, \langle x_2, y_2 \rangle) &= \text{arch}\left(1 + \frac{(x_2 - x_1)^2 + (y_2 - y_1)^2}{2y_1 y_2}\right) = \\ &= 2 \text{arsh} \frac{1}{2} \sqrt{\frac{(x_2 - x_1)^2 + (y_2 - y_1)^2}{y_1 y_2}} \\ &= 2 \ln \frac{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} + \sqrt{(x_2 - x_1)^2 + (y_2 + y_1)^2}}{2\sqrt{y_1 y_2}} \end{aligned}$$

где arch и arsh — это обратные гиперболические функции:

$$\text{arsh } x = \ln(x + \sqrt{x^2 + 1}), \quad \text{arch } x = \ln(x + \sqrt{x^2 - 1}); \quad x \geq 1.$$

Также можно вычислить расстояние между двумя точками с помощью длины дуги вдоль (евклидовой) полуокружности[6]:

$$\text{dist}(AB) = \left| \ln \left(\frac{|BA_\infty| |AB_\infty|}{|AA_\infty| |BB_\infty|} \right) \right|,$$

где A_∞ и B_∞ — точки полуокружности (концы), лежащие на граничной прямой, а $||$ — это евклидова длина сегмента окружности, соединяющей точки.

Кластеры в сети

Кластер – это совокупностью устройства и всех его дочерних узлов. Пример кластеров в сети изображен на рисунке 3.

Согласно общему определению, k -кластер - это набор узлов, которые взаимно достижимы не более чем в k хорд. На рисунке N представлен пример 2-кластер. Также, согласно другому определению, k -кластер определяется как набор узлов в пределах расстояния k -хор от данного узла, который обычно называется clusterhead [2].

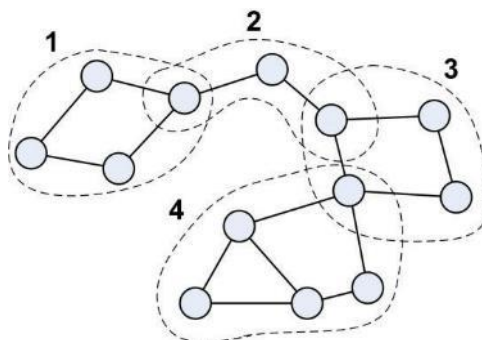


Рис. 3. 2-clusters

GPSR (Greedy Perimeter Stateless Routing)

Очень часто при передаче пакетов от узла к узлу, могут возникать пустоты или дыры [4][5]. Чтобы обойти эту проблему, можно использовать протокол GPSR. Greedy Perimeter Stateless Routing, является гибким и эффективным протоколом маршрутизации для мобильных, беспроводных сетей.

GPSR использует жадную пересылку пакетов в узлы, которые всегда постепенно приближаются к месту назначения (рис 4).

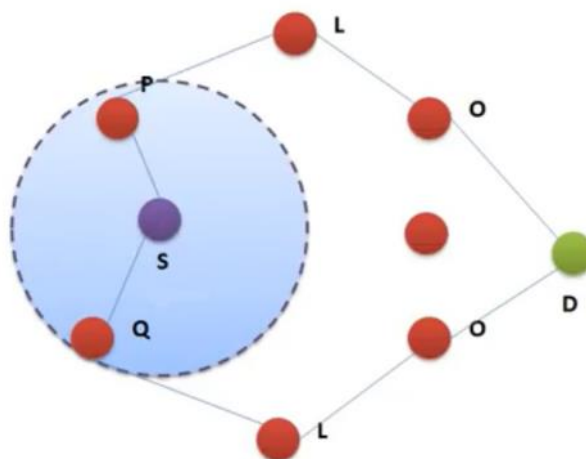


Рис. 4. Greedy Perimeter Stateless Routing

В регионах сети, где такой жадный путь не существует (т.е. путь временно удаляется от места назначения), пакет проходит попеременно более близкие грани до тех пор, пока не достигнет узла ближе к месту назначения, где возобновится жадная пересылка.

Литература

1. Kleinberg R. Geographic routing using hyperbolic space [Text] // INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE. – IEEE, 2007. – P. 1902-1909.
2. Kim Y. J. et al. Geographic routing made practical [Text] // Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation-Volume 2. – USENIX Association, 2005. – P. 217-230.
3. Timofeeva S. V. et al. Greedy forwarding for hyperbolic space in MANET [Text] // Telecommunication Forum (TELFOR), 2017 25th. – IEEE, 2017. – P. 1-4.



4. Sukhov A. M., Onoprienko A. V. Evaluating the effectiveness of geographic routing based on RIPE Atlas data [Text] //Telecommunications Forum Telfor (TELFOR), 2014 22nd. – IEEE, 2014. – P. 107-110.

5. Papadopoulos F. et al. Greedy forwarding in dynamic scale-free networks embedded in hyperbolic metric spaces [Text] //INFOCOM, 2010 Proceedings IEEE. – IEEE, 2010. – P. 1-9.

6. Sukhov A. M., Chemodanov D. Y. A metric for dynamic routing based on variational principles [Text] //Journal of High Speed Networks. – 2013. – Т. 19. – №. 2. – P. 155-163.

И.В. Казакова, С.Н. Попов, С.В. Востокин

МИКРОСЕРВИСНОЕ ПРИЛОЖЕНИЕ ДЛЯ РАСПРЕДЕЛЕННОЙ ОБРАБОТКИ ДАННЫХ НА ПРИМЕРЕ ЗАДАЧИ БЛОЧНОЙ СОРТИРОВКИ

(Самарский национальный исследовательский университет
имени академика С.П. Королёва)

При решении многих научных и технических задач возникает необходимость обработки больших массивов данных. Возможности последовательных приложений не позволяют эффективно использовать аппаратные ресурсы современных вычислительных систем, поэтому применение методов параллельного программирования вызывает активный интерес среди ученых и разработчиков программного обеспечения.

В работе на примере одной из актуальных задач распределенной обработки данных – блочной сортировки – рассматривается способ организации вычислений с применением акторной модели. Главными элементами модели являются акторы – активные агенты, выполняющие определенные действия согласно заданному сценарию [1]. Акторы были использованы для построения микросервисного приложения. Основным преимуществом данного стиля разработки является возможность такой организации приложения, при которой каждый из составляющих его сервисов выполняет часть задач, абстрагируясь от других, а взаимодействие между элементами происходит при помощи стандартных протоколов [2]. В результате нами разработано приложение, имеющее следующую структуру (рисунок 1).

Рассмотрим подробнее элементы структуры, представленной на рисунке 1, и схему их взаимодействия. Всю систему можно разделить на две подсистемы, а именно подсистему управления, включающую в себя акторы *sorter* (a), *producer* (b), *merger* (c), *stopper* (d), и подсистему выполнения, включающую акторы *everest* (f) и *timer* (g).

Предположим, что изначально есть массив данных, состоящий из M целых чисел, который необходимо отсортировать. Разделим его на N равных блоков. На каждый блок создается актор типа *sorter* (a). *Sorter* (a) взаимодействует с акторами *producer* (b) и *everest* (f).