

УДК 681.3

А. А. Григорьев

Московский физико-технический институт (государственный университет)

Алгоритмы маршрутизации в оппортунистических сетях

Для одной модели сети с оппортунистическим поведением узлов предложены алгоритмы маршрутизации, обобщающие известные алгоритмы Дейкстры и Беллмана–Форда. Сформулированы условия на метрики путей, выполнение которых гарантирует оптимальность решений задачи маршрутизации.

Ключевые слова: беспроводная сеть, оппортунистическая сеть, маршрутизация.

1. Введение

Функционирование больших коммуникационных сетей, таких как сеть Интернет, существенно опирается на те или иные процедуры автоматической маршрутизации, которые обеспечивают построение оптимальных маршрутов передачи пакетов от источника до получателя и адаптацию этих маршрутов к изменениям топологии сети. Вопросы маршрутизации достаточно хорошо проработаны для классических моделей сетей с фиксированными каналами, в которых каждый поступивший в узел пакет продвигается далее по одному и только по одному из каналов. Маршрут доставки пакета в таких сетях имеет простую линейную структуру последовательности каналов-узлов.

Алгоритмические основы маршрутизации, базирующиеся на технике обхода графа по ширине, были развиты еще в работах Дейкстры [1] и Беллмана [2] конца 60-х годов XX в. Современное изложение алгоритмов маршрутизации для сетей с линейными маршрутами можно найти в [3].

Широкое распространение беспроводных сетей вынуждает искать новые подходы. Эти сети обладают двумя принципиальными отличиями – это широковещательный характер передачи и возможность потери пакета. С одной стороны, пакет, переданный некоторым узлом, может быть принят сразу несколькими соседними узлами. С другой, есть вероятность потери пакета в канале из-за помех.

Все успешно принявшие пакет узлы становятся его обладателями и в принципе могут внести свой вклад в его доставку конечному получателю. Воплощение этой простой идеи приводит к моделям радиосетей с оппортунистическими стратегиями поведения промежуточных узлов. В оппортунистических сетях узлы-обладатели пакета так или иначе вступают во взаимодействие, стремясь коллективными усилиями решить общую задачу по доставке пакета получателю.

Исследовались различные модели оппортунистических сетей. В простых моделях предполагается, что узлы-обладатели до начала передачи проводят между собой состязание за право выступить ретранслятором. Реально передает пакет только один узел, выигравший состязание. В более сложных моделях все узлы-обладатели излучают пакет одновременно, используя технику пространственно-временного кодирования. Многочисленные результаты моделирования [4], [5] показывают, что оппортунистическое поведение дает значительный выигрыш в пропускной способности сети по сравнению с обычной линейной стратегией, когда единственный узел-ретранслятор оговорен заранее. Это и обуславливает интерес к исследованию оппортунистических сетей.

Классическая проблема маршрутизации – построение множества оптимальных в некоторой метрике маршрутов от источников до получателей – для оппортунистических сетей радикально усложняется. Прежде всего, для таких сетей само понятие маршрута оказывается довольно расплывчатым. Конкретная траектория движения пакета по сети не может

быть указана заранее. Она определяется динамически, в процессе взаимодействия ретрансляторов, подверженного случайным факторам. Так что сама постановка оптимизационной задачи маршрутизации требует как уточнения понятия маршрута, так и задания на множестве маршрутов некоторой метрики, которая позволила бы сравнивать маршруты между собой.

В данной работе рассматривается конкретизированная модель оппортунистической сети с состязательным поведением ретрансляторов. Для этой сети вводится понятие маршрута как дерева узлов-ретрансляторов и операция комбинирования (*сложения*), которая наделяет множество маршрутов свойствами алгебры. Мы не детализируем форму метрик маршрутов, ограничиваясь лишь выделением ряда их свойств, существенных для оптимальности предлагаемых алгоритмов. Это обеспечивает достаточную общность результатов. В работе предложено два алгоритма маршрутизации, обобщающие классические алгоритмы Дейкстры и Беллмана–Форда на случай оппортунистических сетей. Получены достаточные условия оптимальности предложенных алгоритмов, сформулированные в терминах свойств метрик путей.

2. Модель оппортунистической сети, алгебра маршрутов

Сеть представляется ориентированным графом (V, E) , рис. 1, со множеством V помеченных вершин (*узлов сети*) $v \in V$ и множеством $E \subset \{(v, u)\}$ ориентированных ребер (*каналов*). Каждый из каналов (v, u) может быть использован для передачи пакета от узла v к узлу u .

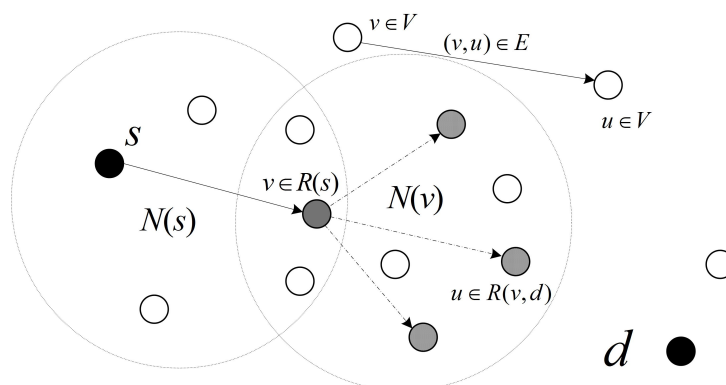


Рис. 1. Граф оппортунистической сети

Для каждого из узлов v определено множество соседей $N(v) \subset V$ – узлов, которые в принципе могут слышать пакеты, исходящие от v . Передача ведется широкоэвентально – сразу всем узлам из $N(v)$.

Предполагается следующий алгоритм продвижения пакетов по сети. Узел v , став так или иначе обладателем пакета, исходящего от источника s и предназначенного для получателя d , действует по следующей оппортунистической схеме. Во множестве своих соседей $N(v)$ от выбирает подмножество потенциальных ретрансляторов $R(v, d) \subset N(v)$ и упорядочивает их по приоритетам. Упорядоченный список $R(v, d) = (v_1, v_2, \dots, v_n)$, $v_j \in N(v)$ потенциальных ретрансляторов и задает локальную стратегию узла v . Информация об этой стратегии помещается в служебное поле передаваемого пакета. В списке ретрансляторов $R(v, d)$, в частности, может содержаться и сам узел назначения d . Тогда ему присваивается наивысший приоритет.

Успешно принявшие пакет ретрансляторы проводят состязание за право дальнейшего его продвижения, используя извлеченный из пакета список приоритетов. Выигрывает тот из них, которому стратегией отправителя присвоен наивысший приоритет. Продвижение пакета по сети завершается, если оказывается, что состязание выиграл узел назначения d .

Специфика рассматриваемой модели проявляется только при условии, что вероятность приема пакетов ретрансляторами заметно отличается от единицы. Тогда состав ретрансляторов, принимающих участие в состязании, оказывается случайным. В противном случае необходимость состязания просто отпадает, поскольку его всегда выигрывает наиболее приоритетный из ретрансляторов, и все сводится к обычной модели с линейными маршрутами.

Технически состязание ретрансляторов может быть организовано по-разному. Один из возможных приемов – это прослушивание радиоканала в течение времени, возрастающего с убыванием приоритета. Если в течение заданного приоритетом интервала прослушивания никто с более высоким приоритетом не начал передачу, то состязание выиграно, и узел начинает передачу, подавляя тем самым все менее приоритетные ретрансляторы. В противном случае пакет отбрасывается. Предполагается, что принявший пакет узел назначения d обеспечивает подавление ретрансляторов, передавая короткую квитанцию. Реализация этой схемы, разумеется, предполагает, что все ретрансляторы из $R(v, d)$ слышат друг друга.

В рассмотренной модели сети маршрут $S(s, d)$ от источника s до узла назначения d – это древовидная структура, которая конструируется рекурсивно сверху вниз – от источника к получателю. На первом этапе фиксируется некоторая локальная стратегия источника s – упорядоченный список ретрансляторов $R(s, d) \in N(s)$. Это позволяет представить маршрут $S(s, d)$ упорядоченным списком подмаршрутов:

$$S(s, d) = [S(v_1, d), \dots, S(v_n, d)], v_j \in R(s, d)$$

Затем фиксируется некоторая локальная стратегия $R(v_j, d)$ для каждого из ретрансляторов и каждый из подмаршрутов $S(v_j, d)$ в свою очередь заменяется списком:

$$S(v_j, d) = [S(v_{j,1}, d), \dots, S(v_{j,m_j}, d)], v_{j,i} \in R(v_j, d).$$

Маршрут от источника становится тогда деревом вложенных списков глубины 2:

$$S(s, d) = [[S(v_{1,1}, d), \dots, S(v_{1,m_1}, d)], \dots, [S(v_{n,1}, d), \dots, S(v_{n,m_n}, d)]].$$

Далее все маршруты глубины 2 расширяются до списков глубины 3. Построение ветви маршрута обрывается, когда в очередном списке ретрансляторов остается единственный узел назначения d .

Мощность множества всех возможных маршрутов $S(v, d)$ из узла v в узел назначения d весьма велика. Два маршрута, отличающиеся составом или порядком списка ретрансляторов на любом из уровней вложения, считаются различными. Возможность образования циклов маршрутизации делает эту мощность бесконечной.

Далее мы будем считать узел назначения d фиксированным, опуская упоминание о нем в обозначениях.

Введем операцию комбинирования маршрутов. Пусть $S(v)$ – маршрут из узла v , в списке $R(v) = (v_1, \dots, v_n)$ ретрансляторов которого отсутствует узел u , и пусть $S(u)$ – маршрут из вершины u . Суммой $S(v) \oplus S(u)$ назовем маршрут из v , в список ретрансляторов которого узел u добавлен в качестве самого низко приоритетного – $R(v) = (v_1, \dots, v_n, u)$.

Операция \oplus вводит на множестве маршрутов некоторое подобие алгебраической структуры. Правда, операция эта оказывается частично определенной. К тому же, как легко убедиться, она некоммутативна и неассоциативна. Однако именно в терминах этой операции мы и будем и конструировать маршруты и формулировать свойства метрик.

3. Метрики маршрутов

Будем предполагать, что на множестве маршрутов в узел d задана некоторая метрика $l(S(v))$, значения которой лежат в том или ином упорядоченном множестве, так что значения метрик маршрутов можно сравнивать. В конкретных моделях такие метрики, как

правило, строятся рекурсивно снизу вверх заданием того или иного правила F нахождения метрики маршрута $S(v) = [S(v_1), \dots, S(v_n)]$ по метрикам подмаршрутов, начинающихся в узлах-ретрансляторах:

$$l(S(v)) = F(l(S(v_1)), \dots, l(S(v_n))), \quad v_j \in R(v).$$

Обширный список конкретных примеров рекурсивно конструируемых метрик для оппортунистических сетей дан в [6].

Принятый здесь подход состоит в том, чтобы, не конкретизируя метрику, постулировать некоторые ее свойства. Для оптимальности предлагаемых ниже алгоритмов оппортунистической маршрутизации достаточно потребовать выполнения следующих трех свойств.

Свойство полезности ретрансляторов. Узел u назовем полезным для узла v , если условие

$$l(S(v) \oplus S(u)) < l(S(v))$$

тождественно условию

$$l(S(u)) < l(S(v)).$$

Это свойство играет ключевую роль, объявляя бесполезными те ретрансляторы из $N(v)$, метрики которых заведомо хуже текущей метрики данного узла v . Бесполезные узлы не будут использоваться при построении оптимальных маршрутов, что исключает возможность циклической маршрутизации.

Свойство сохранения предпочтения. Метрика сохраняет предпочтение, если из $l(S(u)) \leq l(S(v))$ вытекает $l(S(u)) \leq l(S(v) \oplus S(u))$ для $\forall v \in V$ и $\forall u \in N(v)$. Свойство строгого сохранения предпочтения предполагает замену нестрогого неравенства на строгое.

Это свойство означает, что добавление ретранслятора u в список ретрансляторов для узла v не способно сделать метрику узла v лучше метрики ретранслятора.

Свойство оптимальности по ретрансляторам. Это свойство касается поведения метрики узла при переупорядочении списка ретрансляторов. Пусть $R(v) = (v_1, \dots, v_n)$ – набор ретрансляторов для узла v . Обозначим через $R^*(v)$ тот же набор, но с узлами, упорядоченными в порядке возрастания метрик – $l(S(v_1)) \leq l(S(v_2)) \leq \dots \leq l(S(v_n))$. Условие оптимальности по ретрансляторам требует, чтобы именно этот порядок ретрансляторов давал наилучшее значение метрики для узла v :

$$l[S_{R^*(v)}(v)] \leq l[S_{R(v)}(v)].$$

Свойство оптимальности сильно упрощает ситуацию, сводя задачу выбора приоритетов ретрансляторов к простому упорядочению метрик подмаршрутов.

4. Классические алгоритмы

В основе классических алгоритмов маршрутизации лежит следующая процедура обхода графа $G(V, E)$ по ширине (Breadth-Search) [3].

С каждым из узлов $v \in V$ связаны переменные $dist(v)$ и $pred(v)$. Начальные значения $dist(v)$ устанавливаются равным символу ∞ . Это означает, что данный узел еще не был посещен при обходе. Для узла назначения d расстояние $dist(d)$ принимается равным нулю. Начальные значения переменных $pred(v)$ устанавливаются равными \emptyset – отсутствие родительского узла.

Процедура $BS(G, d)$ обеспечивает обход всех узлов $v \in V$ и присваивает переменным $dist(v)$ значения, равные рангам узлов v – расстояниям от v до d в числе ребер. В качестве рабочей структуры она использует очередь вершин Q с примитивами $Extract(Q)$ (*извлечь первый элемент очереди*) и $Insert(Q, v)$ (*поместить узел v в конец очереди*).

После инициализации в очереди находится единственный узел d . Именно он и извлекается при первом проходе цикла *while*.

Procedure BS(G, d)

- (1) *for all* $u \in V$ $dist(u) = \infty$; $pred(u) = \emptyset$;
- (2) $dist(d) = 0$
- (3) $Q = [d]$
- (4) **while** Q is not empty {
- (5) $u = Extract(Q)$;
- (6) **for all** $(v, u) \in E$ {
- (7) *if* ($dist(v) = \infty$) { $dist(v) = dist(u) + 1$; $pred(v) = u$; $Insert(Q, v)$;}
- (8) }
- (9) }

В теле цикла просматриваются все ребра (v, d) , и их узлы v помещаются в очередь как узлы ранга 1: ($dist(v) = 1, pred(v) = d$). По завершении работы *for* все узлы ранга 1 оказываются в очереди. Далее все они поочередно извлекаются и в очередь помещаются узлы ранга 2, удаленные на одно ребро от одной из узлов ранга 1. После извлечения из очереди последнего из узлов ранга 1, в ней окажутся в точности все узлы ранга 2.

Обход графа продолжается до тех пор, пока множество вершин ранга k ни окажется пустым. По завершении обхода значение переменной $dist(v)$ дает минимальное расстояние от v до d в числе ребер, а значения переменных $pred(v)$ позволяют эффективно построить путь минимальной длины из любой вершины v в вершину назначения d . Таким образом, обход графа по ширине дает оптимальное решение задачи маршрутизации для сети, в которой метрикой служит число ребер на маршруте.

Алгоритм Дейкстры обобщает процедуру обхода по ширине на случай, когда ребрам графа присвоены неотрицательные веса $l(u, v) \geq 0$ и ставится задача построения путей минимального веса из всех вершин $v \in V$ в заданную вершину d .

Procedure Dijkstra(G, l, d)

- (1) *for all* $u \in V$ $dist(u) = \infty$; $pred(u) = \emptyset$;
- (2) $dist(d) = 0$
- (3) $H = MakeQueue(V)$
- (4) **while** H is not empty {
- (5) $u = Extract(H)$;
- (6) **for all** $(v, u) \in E$ {
- (7) *if* ($dist(v) > dist(u) + l(v, u)$) { $dist(v) = dist(u) + l(v, u)$;
 $pred(v) = u$; $Rearrange(H, v)$;}
- (8) }
- (9) }

Основное его отличие состоит в использовании не простой, а приоритетной очереди H , элементы которой упорядочены по значениям $dist(v)$. Процедура *MakeQueue* инициализирует эту очередь, помещая в ее голову узел назначения с $dist(d) = 0$ и в произвольном порядке размещая в ней остальные узлы с $dist(v) = \infty$. Процедура *Extract(H)* извлекает самый высоко приоритетный элемент очереди. Процедура *Rearrange(H, v)* обеспечивает переупорядочение очереди после изменения метрики узла v .

На каждом шаге из очереди извлекается узел u , наиболее близкий к d среди всех, все еще находящихся в ней, и производится модификация метрик вершин $v \in N(u)$, соседних с этим узлом. Условие $l(v, u) \geq 0$ гарантирует неубывание значений метрик при модификации. Поэтому метрики извлеченных из очереди узлов не при каких условиях не могут превысить метрики узлов, оставшихся в ней.

Нетрудно понять, что всякий раз по завершении выполнения цикла *while* оказываются выполненными условия:

- 1) Существует значение $d > 0$ такое, что $dist(v) \leq d$ для всех извлеченных узлов и $dist(v) \geq d$ для всех узлов, оставшихся в очереди.
- 2) Для произвольного узла v из очереди значение $dist(v)$ либо равно ∞ , либо равно длине минимального пути из v в d . Причем отличные от конечной точки v узлы этого пути уже извлечены из очереди.

Их выполнение и гарантирует сходимость алгоритма к оптимальному решению.

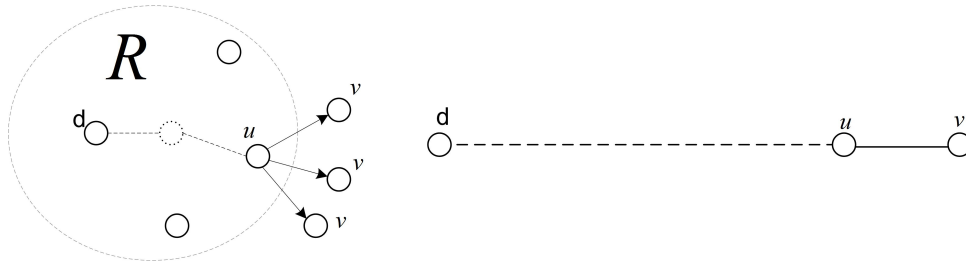


Рис. 2. К алгоритму Дейкстры

Работу алгоритма Дейкстры можно проиллюстрировать в терминах хорошо известного алгоритма Витерби. Рассмотрим рис. 2. Пусть область R на нем включает те узлы u , для которых оптимальные пути в узел назначения d уже построены (вначале эта область содержит только d). Стоит задача определения оптимальных путей для оставшиеся узлы. Пусть маршрут

$$d \leftarrow \dots \leftarrow u \leftarrow v$$

из узла v обладает наименьшим весом среди всех маршрутов из оставшихся узлов. Но тогда маршрут из u в d обладает еще меньшим весом: $dist(v) = dist(u) + l(v, u) \geq dist(u)$, а, следовательно, узел u обязан принадлежать R . Таким образом, оптимальный маршрут из очередного узла v всегда может быть построен как маршрут минимального веса, удлиняющий уже построенный оптимальный маршрут ровно на одно ребро. Причем длина удлиняющего ребра минимальна среди всех ребер, выходящих из вершины u . Нетрудно понять, что алгоритм Дейкстры, выделяющий на каждом шаге узел наименьшего веса среди еще не исключенных из очереди узлов, как раз и реализует присоединение очередного удлиняющего ребра.

Существует нетривиальное обобщение алгоритма Дейкстры, которое допускает распределенную реализацию. Это алгоритм Беллмана–Форда. К нему ведет наблюдение за изменением метрик вершин $dist(v)$ при работе алгоритма Дейкстры. В процессе работы этого алгоритма начальное значение ∞ метрики узла v может изменяться многократно, но всякий раз это изменение реализуется процедурой

$$Update(v, u) : \quad dist(v) = \min\{dist(v), dist(u) + l(v, u)\},$$

примененной для одного из соседних узлов $u \in N(v)$. Нетрудно сделать следующие простые наблюдения относительно свойств процедуры $Update(u, v)$:

- 1) применение $Update(v, u)$ не приводит к увеличению $dist(v)$,
- 2) оптимальному решению отвечает минимально возможное значение $dist(v)$. Поэтому избыточные применения процедуры $Update(u, v)$ безопасны в смысле влияния на окончательный результат.

Выберем некоторый узел t и рассмотрим оптимальный маршрут выделенного из него узла назначения d

$$d \rightarrow u_1 \rightarrow u_2 \rightarrow u_3 \rightarrow \dots \rightarrow u_k \rightarrow t.$$

Тогда значение метрики этого t может быть найдено как результат последовательности применений операторов *Update*:

$$\text{Update}(u_1, d), \text{Update}(u_2, u_1), \text{Update}(u_3, u_2), \dots, \text{Update}(t, u_k)$$

именно в указанном порядке. Вставка в это строку повторов или операторов *Update* со сторонними вершинами не повлияет на конечный результат. Важно только, чтобы вся последовательность операторов была выполнена.

Таким образом, если оптимальный маршрут из t в d известен, то оптимальная метрика $\text{dist}(t)$ легко находится. Как быть, если этот путь маршрут неизвестен? Неожиданный и крайне простой рецепт состоит в следующем: нужно применить оператор *Update* ко всем парам узлов $|V| - 1$ раз, где $|V|$ – число вершин графа. При этом заведомо будет выполнена любая оптимизирующая цепочка операторов. Возможно, будет выполнено много лишних преобразований метрик вершин, но окончательный результат от этого не изменится. Это и есть алгоритм Беллмана–Форда.

Procedure Bellman(G, l, d)

- (1) for all $u \in V$ $\text{dist}(u) = \infty$; $\text{pred}(u) = \emptyset$;
- (2) $\text{dist}(d) = 0$
- (3) repeat ($|V| - 1$) times{
- (4) for all $e \in E$ **Update**(e)
- (5) }

По завершении $|V| - 1$ итерации алгоритм определяет оптимальные метрики и пути для всех узлов. В реальности максимальная длина пути в графе сети бывает много меньше предельного значения $|V| - 1$. Тогда после определенного числа итераций метрики узлов перестают изменяться. Условие сохранения всех значений $\text{dist}(v)$, $v \in V$, на очередной итерации может служить критерием для выхода из цикла *repeat*.

5. Алгоритмы маршрутизации для оппортунистической сети

Предложенные здесь алгоритмы маршрутизации для рассматриваемой модели оппортунистической сети являются несложными обобщениями рассмотренных выше алгоритмов Дейкстры и Беллмана–Форда. Нетривиальный результат состоит в выявлении требований к метрикам маршрутов, которые гарантируют сходимость этих алгоритмов к оптимальным решениям. Оба алгоритма реализуют построение оптимальных маршрутов и заодно вычисляют значения их метрик снизу вверх – от узла получателя d ко всем узлам сети.

Алгоритм 1. Обобщение алгоритма Дейкстры приводит к процедуре *DijkstraOR*.

На этапе инициализации в качестве множества непросмотренных узлов Q принимается все множество узлов V , узлу получателю d присваивается метрика 0, всем остальным узлам – метрика ∞ .

На каждом шаге функция *ExtractBest*(Q) выделяет во множестве узлов Q узел u с минимальной метрикой маршрута до d . На первом шаге это будет сам узел d . Выбранный узел исключается из списка Q . Затем он предлагает себя всем оставшимся в Q узлам в качестве ретранслятора по доставке в d и становится таковым, если это дает снижение метрики. Алгоритм продолжается до исчерпания множества Q .

DijkstraOR(G, d)

- (1) $Q \leftarrow V$
- (2) **while** $Q \neq \emptyset$ **do** {
- (3) $u \leftarrow \mathbf{ExtractBest}(Q)$
- (4) **for each** $v \in N(u) \cap Q$ **do** {
- (5) **if** $l(S(v) \oplus S(u)) \prec l(S(v))$
- (6) $S(v) \leftarrow S(v) \oplus S(u)$
- (7) }
- (8) }

По завершении работы алгоритма для каждого из узлов $v \in V$ оказывается построенным дерево маршрута $S(v, d)$ доставки пакетов в узел d . Алгоритм повторяется для каждого из узлов получателей в сети.

Алгоритм 2. Алгоритм Беллмана–Форда обобщается для оппортунистических сетей следующим образом:

Procedure **BellmanOR**(G, d)

- (1) Инициализация: $l(S(v_i)) = \infty$; $l(S(d)) = 0$;
- (2) **repeat**{
- (3) Сортируем узлы в порядке убывания метрик.
- (4) **for** $i = 2$ **to** $(|V| - 1)$ **do** {
- (5) $l(S(v_i)) = \mathbf{FindRelays}(v)$
- (6) }
- (7) } **until** изменений не зарегистрировано,

где

Function **FindRelays**(v)

- (1) Сортируем ретрансляторы $u_j \in N(v)$ в порядке убывания метрик;
- (2) Инициализация: $S(v) = \emptyset$; $j = 1$;
- (3) **while** $l[S(v) \oplus S(u_j)] \prec l[S(v)]$ {
- (4) $S(v) = S(v) \oplus S(u_j)$; $j = j + 1$;
- (5) }
- (6) **return** $l(S(v))$

Оператор в строке (3) основной процедуры $\mathbf{BellmanOR}(G, d)$ сортирует узлы v в порядке убывания метрик $l(S(v))$ и нумерует их как v_i , так что малым значениям индекса i отвечают узлы с малыми значениями метрик. Вершине v_1 с метрикой 0 отвечает узел назначения d , который не участвует в итерациях.

Каждый вызов процедуры $\mathbf{FindRelays}(v)$ очищает список ретрансляторов для узла v и строит новый список, включая в него узлы из $N(v)$ в порядке убывания их действующих метрик до тех пор, пока подключение очередного ретранслятора дает снижение метрики $l(S(v))$.

Основной декларируемый здесь результат состоит в следующем.

Основная теорема. Алгоритмы оппортунистической маршрутизации $\mathbf{DijkstraOR}$ и $\mathbf{BellmanOR}$ приводят к построению оптимальных маршрутов для всех узлов сети, если метрика $l(S(v))$:

- (1) обладает свойством полезности ретрансляторов,
- (2) строго сохраняет предпочтения и
- (3) оптимальна по ретрансляторам.

Довольно простое по сути, но громоздкое индуктивное доказательство этого результата см. в [7], [8].

6. Заключение

Место полученных результатов можно охарактеризовать следующим образом. По существу, мы обобщили известные алгоритмы маршрутизации для некоторой модели оппортунистической сети и выявили требования к метрикам маршрутов, выполнение которых гарантирует оптимальность решений задачи маршрутизации. Это не снимает проблему маршрутизации для моделей, в которых приняты здесь требования к метрикам не выполняются.

Такие модели существуют. Одна из них, известная как модель *ETX*, была введена в [6]. Метрики маршрутов строятся в этой модели как оценки среднего числа повторных передач. *ETX*-метрики не удовлетворяют свойству оптимальности ретрансляторов и поэтому обобщенные Дейкстра-алгоритмы к ней неприменимы. Более того, можно показать, что для этой модели алгоритм *DijkstraOR* может давать результаты, отличающиеся от оптимальных сколь угодно сильно.

Так что задача построения алгоритмов маршрутизации для моделей сетей с метриками, для которых оптимальность Дейкстра-схем не гарантирована, остается актуальной и открытой.

Литература

1. *Dijkstra E. W.* A note of two problems in connexion with graphs // *Numerische Mathematic.* — 1959. — V. 1. — P. 269–271.
2. *Bellman R.* On a Routing Problem // *Quarterly of Applied Mathematic.* — 1958. — V. 16. — P. 87–90.
3. *Coremн T. H., Leiserson C. E. Rivest R. L.* Introduction to Algorithms. — MIT Press and McGraw-Hill, 2001.
4. *Yang Y., Wang J.* Design guidelines for routing metrics in multihop wireless networks // *Proceedings of IEEE INFOCOM.* — 2008. — P. 216–244.
5. *Wu J., Lu M., Li F.* Utility-based opportunistic routing in multihop wireless networks // *Proceedings of IEEE ICDS.* — 2008. — P. 126–132.
6. *Sobrinho J.* Algebras and algorithms for QoS path computation and hop-by-hop routing in the internet // *Proceedings of IEEE INFOCOM.* — 2001. — P. 727–735.
7. *Григорьев А. А.* Алгебры маршрутизации над деревьями – новый подход к оппортунистической маршрутизации. Отчет по теме № 713. — 2011.
8. *Григорьев А. А.* Оптимальность алгоритма Беллмана–Форда для задач оппортунистической маршрутизации. Отчет по теме № 713. — 2012.

Поступила в редакцию 18.02.2013.