

Санкт-Петербургский государственный университет

Программная инженерия

Кафедра системного программирования

Чернов Андрей Олегович

Система распределения временных слотов
в беспроводных сетях на основе протокола
локального голосования с учетом
приоритетов заданий

Бакалаврская работа

Научный руководитель:
д. ф.-м. н., профессор Граничин О. Н.

Рецензент:
к. ф.-м. н., ст. науч. сотрудник ИПМаш Иванский Ю. В.

Санкт-Петербург
2019

SAINT PETERSBURG STATE UNIVERSITY

Software engineering

Andrey Chernov

System for Local Voting Protocol Based
Time-Slot Distribution in Wireless Networks
with Different Task Priorities

Graduation Thesis

Scientific supervisor:
Doctor of Science, Professor O. N. Granichin

Reviewer:
PhD, Senior researcher IPME RAS Y. V. Ivanskiy

Saint Petersburg
2019

Оглавление

Введение	4
Цель работы	5
1. Обзор	6
1.1. DRAND	6
1.2. Алгоритм Люя	6
1.3. LoBaTS	6
1.4. LQF	6
1.5. Протокол локального голосования	7
2. Модификация протокола локального голосования для учета приоритетов заданий	8
2.1. Модель сети	8
2.2. Общий алгоритм работы	8
2.3. Исполнение заданий с учетом приоритетов	9
2.4. Балансировка нагрузки	9
3. Реализация системы	11
4. Сравнение алгоритмов	12
4.1. Сравнение модифицированного и немодифицированного алгоритма	12
4.2. Влияние коэффициентов важности приоритетов на работу сети	14
4.3. Сравнение различных алгоритмов распределения временных слотов	16
Заключение	20
Список литературы	21

Введение

Многоскачковые беспроводные сети являются видом беспроводных сетей, в которых узлы могут выступать как маршрутизатор, перенаправляя трафик следующим узлам. Примерами многоскачковых сетей являются ячеистые, сенсорные и самоорганизующиеся сети. Помимо сетей с фиксированной структурой, автомобильные самоорганизующиеся сети и умные сети электроснабжения положили начало сетям, в которых структура может меняться со временем. Такие сети, в частности, должны продолжать бесперебойную работу в случае удаления и добавления новых узлов. Для этого необходимы алгоритмы с возможностью самостоятельной перенастройки.

Одной из проблем в многоскачковых сетях является задача балансировки нагрузки. Оптимизация балансировки нагрузки является ключевой для эффективного распределения ресурсов для сетей следующего поколения. Множественный доступ с разделением по времени позволяет уменьшать задержку при доставке пакетов и увеличивать пропускную способность сети, при условии оптимального распределения временных слотов.

При таком подходе все время работы сети разбивается на равные промежутки времени – кадры, а те, в свою очередь, на еще более мелкие промежутки – временные слоты. На каждом кадре узел получает возможность передавать пакеты в отведенные ему временные слоты. Узлы, не имеющие общих соседей, могут использовать один и тот же временной слот, так как это исключает возможные коллизии и потерю данных.

На практике во многих сетях используется разделение заданий по степени важности. Такими заданиями могут являться системные запросы, скорейшее выполнение которых обеспечивает работу всей сети; или же пакеты привилегированных пользователей, запросы которых должны исполняться с минимальной задержкой.

Цель работы

Целью работы является создание прототипа системы, децентрализовано распределяющей временные слоты между узлами беспроводной сети на основании модифицированного мультиагентного протокола локального голосования, учитывающего приоритеты заданий.

Для достижения этой цели были поставлены следующие задачи:

1. Исследовать предметную область и изучить существующие алгоритмы распределения временных слотов.
2. Предложить модификацию протокола локального голосования, учитывающую приоритеты заданий при распределении временных слотов.
3. Реализовать прототип системы, моделирующий работу протокола.
4. Провести сравнение модифицированного алгоритма с немодифицированным, а также с другими алгоритмами распределения временных слотов.

1. Обзор

Проблема балансировки нагрузки в многоскачковых сетях исследовалась во множестве работ. В [1, 2, 3] был рассмотрен централизованный подход; в [4] – децентрализованный. Для задачи распределения временных слотов также было представлено несколько алгоритмов.

1.1. DRAND

Работа алгоритма DRAND (distributed randomized TDMA scheduling) описана в работе [6]. Этот метод является децентрализованной модификацией алгоритма RAND, представленного в [5]. В алгоритме DRAND каждый узел сети с некоторой вероятностью проверяет наличие временных слотов у соседей в двух скачках от себя и присваивает его себе, если этот слот не занят кем-либо из них.

1.2. Алгоритм Люя

Алгоритм Люя (Lyu), описанный в [7], основан на идее раскраски графа сети. В процессе работы каждый узел выбирает себе цвет – наименьшее положительное число, которое еще не назначено узлу из двухскачковой окрестности. Все последующие балансировки производятся на основании полученной раскраски.

1.3. LoBaTS

Алгоритм LoBaTS (Load-Based Transmission Scheduling) является модификацией алгоритма Люя. Принцип его работы описан в [8]. Основное отличие этого алгоритма от алгоритма Люя – наличие у узлов дополнительных цветов, также учитывающихся при балансировке.

1.4. LQF

В отличие от представленных выше алгоритмов, LQF (Longest queue first), описанный в [9], является централизованным. Этот алгоритм тре-

бует наличия информации о загрузке всех узлов сети, вследствие чего имеет малую практическую ценность и используется, в основном, в качестве эталонного теста.

1.5. Протокол локального голосования

В [10] было предложено применение мультиагентного протокола локального голосования для задачи распределения временных слотов. Затем в [11] было показано, что протокол демонстрирует наилучшие результаты по пропускной способности, задержке и времени доставки среди других алгоритмов. Протокол локального голосования был взят за основу представленной в работе модификации, учитывающей приоритеты заданий при распределении временных слотов.

2. Модификация протокола локального голосования для учета приоритетов заданий

2.1. Модель сети

Сеть представляет собой набор соединенных узлов. В сети возникают соединения между двумя узлами: узел-отправитель отправляет по сети кратчайшим маршрутом пакеты узлу-получателю. Пересылка пакетов происходит в отведенные узлу временные слоты. На каждом кадре пакет может перейти лишь в соседний узел. Когда все пакеты переданы, соединение может считаться законченным.

Задания можно распределить по k приоритетам. Каждый приоритет имеет коэффициент важности c_k .

В каждом узле все пакеты добавляются в очереди соответствующих им приоритетов. Задания внутри одного приоритета исполняются в порядке очереди. На кадре t в каждом узле i число заданий с приоритетом k обозначим как q_{ik}^t , общее число заданий как q_i^t , число назначенных узлу временных слотов – p_i^t . Также обозначим множество соседей узла i как N_i .

На каждом кадре t узел i будет вычислять значение u_i^t , на основании которого будет происходить обмен временными слотами между узлами.

2.2. Общий алгоритм работы

Все шаги алгоритма выполняются в служебные временные слоты, выделенные на стыке кадров.

В начале каждого кадра t , каждый узел i освобождает все зарезервированные слоты, если очередь его заданий пуста. Если очередь не пуста, узел пытается зарезервировать свободные временные слоты. Для каждого узла временной слот считается свободным, если он не зарезервирован соседом из двухскачковой окрестности.

Если для узла i значение u_i^t положительно, то он ищет среди сво-

их соседей узел j , имеющий минимальное отрицательное значение u_j^t . Далее между узлами происходит обмен временными слотами. Узел i получает, а узел j лишается $r = \min(u_i^t, u_i^t - u_j^t, p_j^t)$ временных слотов. После обмена u_i^t уменьшается, а u_j^t увеличивается на r . Если новое значение u_i^t узла i по-прежнему положительно, он ищет другого соседа j с минимальным отрицательным значением u_j^t . Если у всех соседей узла i значения u_j^t неотрицательны, обмен не происходит. Также следует учитывать, что обмен временных слотов не должен приводить к появлению коллизий, то есть временные слоты, которые передает узел j , не должны быть назначены узлам из двухскачковой окрестности узла i .

Далее происходит непосредственное выполнение заданий: каждый узел в отведенные ему временные слоты передает пакеты по кратчайшему маршруту до получателя пакета. После того, как узел i передал пакеты и получил новые от своих соседей, происходит подсчет нового значения u_i^{t+1} .

2.3. Исполнение заданий с учетом приоритетов

Исполнение заданий должно происходить таким образом, чтобы, с одной стороны, задания более высоких приоритетов исполнялись быстрее, чем задания низких. С другой стороны, не должна возникать проблема «голодания», из-за которой низкоприоритетные задания были бы вынуждены ожидать полного завершения высокоприоритетных заданий.

Для решения этой проблемы на каждом кадре t для каждого узла i разделим имеющиеся слоты p_i^t на k частей: $p_{ik}^t = \frac{p_i^t \cdot c_k}{\sum_{a_{ik}^t \neq 0} c_k}$. В случае, если число назначенных слотов для заданий определенного приоритета больше необходимого, оставшиеся неиспользованными временные слоты перераспределяются между заданиями других приоритетов.

2.4. Балансировка нагрузки

Ключевой частью протокола локального голосования является обмен слотами. Для каждого узла i можно подсчитать значение его по-

луобратной загрузки $x_i^t = \frac{p_i^t}{q_i^t}$. Когда узлы обмениваются временными слотами они приходят к консенсусу: значения x_i^t для узлов сети постепенно сходятся к одинаковому значению у всех узлов сети.

Для задач, имеющих разные приоритеты, консенсус будет не единым, а дифференцированным: для каждого класса важности k узлы будут иметь свое консенсусное значение $x_{ik}^t = \frac{p_{ik}^t}{q_{ik}^t}$.

Для того, чтобы сеть достигла консенсуса по каждому классу приоритетов, необходимо посчитать значение u_{ik}^t для каждого приоритета: $u_{ik}^t = \left[q_{ik}^t \cdot \frac{p_{ik}^t \cdot \sum_{j \in N_i} p_{jk}^t}{q_{ik}^t \cdot \sum_{j \in N_i} q_{jk}^t} \right] - p_{ik}^t$. Дробь $\frac{q_{ik}^t}{q_{ik}^t \cdot \sum_{j \in N_i} q_{jk}^t}$ является долей числа задач приоритета k этого узла от общего числа заданий приоритета k среди узлов в односкачковой окрестности. Часть выражения в скобках – такая же доля назначенных узлу временных слотов для задач приоритета k от числа всех таких слотов в односкачковой окрестности. Таким образом, u_{ik}^t – отклонение текущего числа назначенных временных слотов от необходимого для достижения консенсуса по приоритету k .

Одним из возможных вариантов является проведение нескольких раундов обмена для каждого из приоритетов. Однако, поскольку далее будет проведено перераспределение временных слотов внутри узла согласно коэффициентам c_k , то обмен слотами можно провести в один раунд. В таком случае, необходимое изменение слотов будет вычисляться как $u_i^t = \sum_k u_{ik}^t$.

3. Реализация системы

Прототип системы распределения временных слотов в беспроводных сетях реализован на языке программирования JAVA с использованием библиотеки JADE. Эта библиотека предназначена для создания мультиагентных приложений и предоставляет для этого базовый функционал: классы агентов, возможность передачи между ними сообщений, программируемые поведения агентов.

Каждый узел возможной сети представлен в виде агента, циклично выполняющий необходимый алгоритм. Синхронизация агентов выполняется на стыке кадров. Соединения представляют из себя массив статических объектов. Они необходимы для того, чтобы имитировать поступление новых заданий и собирать необходимую статистику по их прибытию к узлу-получателю. Для маршрутизации сеть использует кратчайшие пути. Сеть завершает свою работу, когда доставлены пакеты всех заданных соединений.

Конфигурация сети задается с помощью файла "JSON". Возможно задание любого числа временных слотов, соединений и приоритетов, узлов и связей между ними. В соединении можно задать число пакетов каждого приоритета. Оно может быть активно как с начала работы сети, так и активироваться в конкретный кадр. Также соединение может передавать все свои пакеты как одновременно, так и с заданной периодичностью.

4. Сравнение алгоритмов

При сравнения будут учитываться следующие характеристики соединений:

- время доставки – число кадров, за которое все пакеты соединения достигли узла-получателя;
- задержка – число кадров, за которое отдельный пакет достигает узла-получателя. Для соединения будет взято среднее значение задержки каждого его пакета;
- пропускная способность – отношение числа пакетов соединения к числу кадров, за которое они были доставлены.

Для всей сети будут использоваться минимальные, максимальные и средние значения этих характеристик. Кроме того, для них будет считаться индекс справедливости Джайна, описанный в [12].

4.1. Сравнение модифицированного и немодифицированного алгоритма

При моделировании проводилось сравнение трех алгоритмов: протокола локального голосования, учитывающего приоритеты лишь при исполнении заданий (LVP-1), протокола локального голосования, учитывающего приоритеты и при исполнении заданий, и при распределении временных слотов (LVP-2), а также двух вариантов немодифицированного протокола локального голосования, описанного в [11] (в первом из них временные слоты делятся поровну на задания каждого приоритета (LVP-3); во втором – сначала выполняются задания первого приоритета, и, только если их не осталось, второго (LVP-4)).

Моделирование проводилось на сети, представляющей из себя замкнутое кольцо из 10 узлов. В этой сети были созданы 10 соединений, каждое из которых должно было передать 50 пакетов первого приоритета и 200 пакетов – второго со следующими коэффициентами: $c_1 = 4$, $c_2 = 1$. Каждый кадр состоит из 10 временных слотов.

Таблица 1: Результаты моделирования для разных вариантов протокола локального голосования

Характеристика	LVP-1	LVP-2	LVP-3	LVP-4
Максимальное время доставки для заданий приоритета I, кадр	83	68	133	88
Максимальное время доставки для заданий приоритета II, кадр	246	243	247	245
Среднее время доставки для заданий приоритета I, кадр	51,2	47,7	77	58,9
Среднее время доставки для заданий приоритета II, кадр	170,4	172,2	166,5	166
Справедливость для времени доставки заданий приоритета I	0,783	0,859	0,809	0,809
Справедливость для времени доставки заданий приоритета II	0,889	0,902	0,828	0,838
Средняя задержка для заданий приоритета I, кадр	38,476	35,418	57,934	47,112
Средняя задержка для заданий приоритета II, кадр	145,837	141,837	136,348	137,897

Как видно из таблицы 1, все протоколы показывают примерно одинаковое максимальное время задержки для заданий второго приоритета. Эта характеристика очень важна: она показывает, за сколько кадров сеть выполняет все свои задания, то есть завершает свою работу. Можно видеть, что минимального времени на исполнение задач первого приоритета удастся добиться с помощью представленной в работе модификации протокола локального голосования. При этом не сильно ухудшаются задержка и время доставки для задач второго приоритета. Кроме того видно, что в этой версии достигается максимальная справедливость времени задержки для обоих приоритетов среди всех четырех вариантов протокола.

4.2. Влияние коэффициентов важности приоритетов на работу сети

Важную роль в распределении временных слотов играют задаваемые пользователем коэффициенты важности c_k . В этом разделе представлены результаты моделирования для сети с четырьмя разными значениям c_1 и c_2 .

Моделирование проводилось на сети, представляющей из себя кольцо из 15 узлов, в котором помимо соседей каждый узел соединен со своим пятым соседом слева и справа. В этой сети были созданы 15 соединений с 25 пакетами первого приоритета и 100 пакетами – второго. Каждый кадр состоит из 30 временных слотов.

Из таблицы 2 видно, что с ростом значения c_1 для заданий первого приоритета также росла пропускная способность, а время доставки и задержка уменьшались, пока не достигли предела. Обратное им изменялись соответствующие значения для заданий второго приоритета. Для коэффициентов $c_1 = 2$, $c_2 = 1$ и $c_1 = 4$, $c_2 = 1$ отношения задержки и времени доставки задач первого коэффициента ко второму близко к отношению самих коэффициентов c_1 и c_2 .

Таблица 2: Результаты моделирования для разных значений коэффициентов c_k

Характеристика	$c_1 = 2,$ $c_2 = 1$	$c_1 = 4,$ $c_2 = 1$	$c_1 = 8,$ $c_2 = 1$	$c_1 = 16,$ $c_2 = 1$
Среднее время доставки заданий приоритета I, кадр	24	17,267	15,933	14,2
Среднее время доставки заданий приоритета II, кадр	56,934	55,267	56,8	55,2
Средняя задержка заданий приоритета I, кадр	19,296	13,997	13,256	11,563
Средняя задержка заданий приоритета II, кадр	46,017	45,746	45,868	45,574
Средняя пропускная способность для заданий приоритета I, пакет/кадр	1,936	2,294	3,185	3,302
Средняя пропускная способность для заданий приоритета II, пакет/кадр	2,229	2,034	1,947	1,946

4.3. Сравнение различных алгоритмов распределения временных слотов

В этом разделе представлены результаты моделирования алгоритмов, перечисленных в разделе «Обзор». Для этих алгоритмов нет модификаций, учитывающих приоритеты заданий, поэтому в них будет использоваться учет приоритетов заданий лишь при их исполнении, как в представленной модификации протокола локального голосования.

Моделирование проводилось на сети, представляющей из себя кольцо из 20 узлов, в котором помимо соседей каждый узел также соединен с противоположным – десятым соседом (с любой стороны). Задания имеют коэффициенты $c_1 = 4$ и $c_2 = 1$. В каждом кадре 20 временных слотов. Каждое соединение доставляет 30 пакетов первого приоритета и 120 – второго. Не все эти соединения активны сразу: они появляются в течение 30 первых кадров. Пакеты соединения также отправляются периодически: по 10 заданий первого приоритета каждые 20 кадров; по 40 заданий второго приоритета – каждые 5.

Как можно видеть из таблиц 3, 4 и 5, протокол локального голосования с учетом приоритетов показывает примерно те же значения, что и без них (сравнение алгоритмов без приоритетов можно найти в [11]).

Таблица 3: Результаты моделирования для различных алгоритмов (время доставки)

Характеристика времени доставки	LVP	LQF	DRAND	Lyui	LoBaTS
Среднее, приоритет I, кадр	45,45	42,95	65,2	63	54,95
Среднее, приоритет II, кадр	101	122,3	181,6	204,1	192,25
Максимальное, приоритет I, кадр	57	52	104	97	82
Максимальное, приоритет II, кадр	171	194	362	373	358
Минимальное, приоритет I, кадр	26	25	29	28	26
Минимальное, приоритет II, кадр	18	45	53	76	52
Справедливость, приоритет I	0,971	0,969	0,876	0,921	0,935
Справедливость, приоритет II	0,825	0,889	0,785	0,858	0,826

Таблица 4: Результаты моделирования для различных алгоритмов (задержка)

Характеристика задержки	LVP	LQF	DRAND	Lyui	LoBaTS
Средняя, приоритет I, кадр	11,213	6,525	23,68	21,587	15,902
Средняя, приоритет II, кадр	81,94	95,843	146,918	163,457	152,579
Максимальная, приоритет I, кадр	23,333	12,667	62,1	50,6	34,067
Максимальная, приоритет II, кадр	162,167	172,658	334,908	338,5	323,5
Минимальная, приоритет I, кадр	2,5	2,233	4	4	4
Минимальная, приоритет II, кадр	7,283	20,942	24,142	37,75	22,183
Справедливость, приоритет I	0,792	0,825	0,637	0,698	0,708
Справедливость, приоритет II	0,759	0,824	0,719	0,797	0,775

Таблица 5: Результаты моделирования для различных алгоритмов (пропускная способность)

Характеристика пропускной способности	LVP	LQF	DRAND	LyuI	LoBaTS
Средняя, приоритет I, пакет/кадр	0,687	0,727	0,537	0,525	0,591
Средняя, приоритет II, пакет/кадр	1,813	1,141	0,918	0,716	0,849
Максимальная, приоритет I, пакет/кадр	1,154	1,2	1,034	1,071	1,154
Максимальная, приоритет II, пакет/кадр	6,667	2,667	2,264	1,579	2,308
Минимальная, приоритет I, пакет/кадр	0,526	0,577	0,288	0,309	0,366
Минимальная, приоритет II, пакет/кадр	0,702	0,169	0,331	0,322	0,335
Справедливость, приоритет I	0,949	0,951	0,858	0,894	0,914
Справедливость, приоритет II	0,853	0,841	0,726	0,803	0,691

Заключение

В ходе работы были получены следующие результаты:

1. проведен обзор алгоритмов распределения временных слотов в беспроводных многоскачковых сетях;
2. для алгоритма локального голосования представлена модификация, реализующая распределение слотов с учетом приоритетов заданий;
3. прототип системы распределения временных слотов в беспроводных сетях реализован на языке Java с использованием библиотеки JADE;
4. проведены сравнения модифицированного протокола локального голосования с другими алгоритмами распределения временных слотов в беспроводных сетях и с немодифицированным протоколом локального голосования.

Список литературы

- [1] *R. Gunasekaran, S. Siddharth, P. Krishnaraj, M. Kalaiarasan, V. R. Uthariaraj* Efficient algorithms to solve broadcast scheduling problem in wimax mesh networks // Computer Communications, vol. 33, no. 11, pp. 1325-1333, 2010.
- [2] *D. Arivudainambi, D. Rekha* Heuristic approach for broadcast scheduling, problem in wireless mesh networks // AEU-International Journal of Electronics and Communications, vol. 68, no. 6, pp. 489-495, 2014.
- [3] *D. Arivudainambi, S. Balaji* Improved memetic algorithm for energy efficient sensor scheduling with adjustable sensing range // Wireless Personal Communications, pp. 1-22, 2016.
- [4] *C.-T. Chiang, H.-C. Chen, W.-H. Liao, K.-P. Shih* A decentralized minislot scheduling protocol (dmsp) in tdma-based wireless mesh networks // Journal of Network and Computer Applications, vol. 37, pp. 206-215, 2014.
- [5] *S. Ramanathan, E. L. Lloyd* Scheduling algorithms for multihop radio networks // IEEE/ACM Transactions on Networking, vol. 1, no. 2, pp. 166-177, 1993.
- [6] *I. Rhee, A. Warriar, J. Min, L. Xu* DRAND: distributed randomized tdma scheduling for wireless ad-hoc networks // Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing. ACM, pp. 190-201, 2006.
- [7] *W.-P. Lyoui* Design of a new operational structure for mobile radio networks // Ph.D. dissertation, Clemson Univ., Clemson, SC, 1991.
- [8] *B. J. Wolf, J. L. Hammond, H. B. Russell* A distributed load-based transmission scheduling protocol for wireless ad hoc networks // Proceedings of the 2006 International Conference on Wireless Communications and Mobile Computing. ACM, pp. 437-442, 2006.

- [9] *A. Dimakis, J. Walrand* Sufficient conditions for stability of longest-queue-first scheduling: Second-order properties using fluid limits // *Advances in Applied Probability*, pp. 505-521, 2006.
- [10] *D. J. Vergados, N. Amelina, Y. Jiang, K. Krilevska, O. Granichin* Local voting: optimal distributed node scheduling algorithm for multihop wireless networks // *2017INFOCOM-Workshop-Proceedings*, Atlanta, GA, USA, 1-4 May 2017, pp. 931-932, 2017
- [11] *D. J. Vergados, N. Amelina, Y. Jiang, K. Krilevska, O. Granichin* Towards Optimal Distributed Node Scheduling in a Multihop Wireless Network through Local Voting // *IEEE Transactions on Wireless Communications*, 17 (1), pp. 400-414, 2018.
- [12] *R. Jain, D. Chiu, W. Hawe* A quantitative measure of fairness and discrimination for resource allocation in shared computer systems // *Digital Equipment Corporation, Maynard, MA, USA, DEC Research Report TR-301*, 1984.