

РАСПРЕДЕЛЕНИЕ ВРЕМЕННЫХ СЛОТОВ НА ОСНОВЕ ПРОТОКОЛА ЛОКАЛЬНОГО ГОЛОСОВАНИЯ С УЧЕТОМ ПРИОРИТЕТОВ ЗАДАНИЙ В БЕСПРОВОДНЫХ СЕТЯХ

Чернов А. О., 4 курс, кафедра системного программирования СПбГУ,
a.o.chernov@mail.ru

Руководитель: Граничин О. Н., д.ф.-м.н., профессор кафедры системного
программирования СПбГУ, o.granichin@spbu.ru

Аннотация

Многоскачковые беспроводные сети требуют оптимальной балансировки нагрузки для эффективной работы. Для этого необходимо решать задачу распределения временных слотов между узлами. Однако, некоторые задания сети могут требовать более быстрого исполнения относительно других заданий. В статье будет рассмотрена модификация протокола локального голосования, позволяющая оптимально распределять временные слоты в сети, задания которой имеют приоритеты.

Введение

Многоскачковые беспроводные сети являются типом соединения в беспроводных сетях, 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 данном 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.

Алгоритмы для распределения временных слотов были рассмотрены в некоторых статьях. Такими являются алгоритмы DRAND представленный в [1], алгоритм Люя – в [2], LoBaTS – в [3]. Кроме того в [4] предложен алгоритм LQF, являющийся централизованным и поэтому неприменимый на практике, но который может являться эталонным тестом.

В [5] был предложен мультиагентный протокол локального голосования. Затем в [6] было показано, что алгоритм демонстрирует наилучшие результаты по пропускной способности, задержке и справедливости среди алгоритмов.

На практике, во многих сетях используется разделение заданий по степени важности. К таким приоритетам можно отнести системные запросы, скорейшее выполнение которых обеспечивает работу всей сети; или же наличие привилегированных пользователей, запросы которых должны исполняться с минимальной задержкой.

Цель работы

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

Для достижения данной цели были поставлены следующие задачи:

1. Модифицировать протокол локального голосования для того, чтобы при распределении временных слотов учитывались не только загрузка узла, но и приоритеты заданий в очереди.
2. Реализовать прототип системы, моделирующей работу данного алгоритма
3. Провести сравнение модифицированного и стандартного алгоритмов

Протокол локального голосования, учитывающий приоритеты заданий

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. В каждом узле i число заданий с приоритетом k обозначим как q_{ik} , число назначенных узлу временных слотов – p_i . Также обозначим множество соседей узла i как N_i .

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

Общий алгоритм работы

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

В начале каждого кадра, каждый узел 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 положительно, то он ищет среди своих соседей узел j , имеющий минимальное отрицательное значение u_j . Далее между узлами происходит обмен узлами. Узел i получает, а узел j лишается $r = \min(u_i, u_i - u_j, p_j)$ временных слотов. После обмена u_i уменьшается, а u_j увеличивается на r . Если новое значение u_i узла i по-прежнему положительно, он ищет другого соседа с минимальным отрицательным значением u . Если у всех соседей узла i значения u 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 данного пакета. После того, как узел передал пакеты и получил новые от своих соседей, происходит подсчет нового значения u .

Исполнение заданий с учетом приоритетов

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

Для решения этой проблемы разделим имеющиеся слоты p_i на k частей: $p_{ik} = \frac{p_i \cdot c_k}{\sum_{q_{ik} \neq 0} c_k}$. В случае, если число назначенных слотов для заданий определенного приоритета больше необходимого, лишние временные слоты перераспределяются между заданиями других приоритетов.

Балансировка нагрузки

Ключевой частью протокола локального голосования является обмен слотов. Для каждого узла i можно подсчитать значение его полуобратной загрузки $x_i = \frac{p_i}{q_i}$. Когда узлы обмениваются временными слотами они приходят к консенсусу: значения x_i для узлов сети постепенно сходятся к одинаковому значению.

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

Для того, чтобы сеть достигла консенсуса по каждому классу приоритетов, необходимо посчитать значения u для каждого класса: $u_{ik} = \left[q_{ik} * \frac{p_{ik} + \sum_{j \in N_i} p_{jk}}{q_{ik} + \sum_{j \in N_i} q_{jk}} \right] - p_{ik}$. Дробь $\frac{q_{ik}}{q_{ik} + \sum_{j \in N_i} q_{jk}}$ является долей числа задач приоритета k данного 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 – отклонение текущего числа назначенных временных слотов от

необходимого для достижения консенсуса.

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 важности приоритетов, то обмен слотами можно провести в один раунд. В таком случае, необходимое изменение слотов будет вычисляться как $u_i = \sum_k u_{ik}$.

Результаты

На рисунках 1, 2 и 3 изображены графики изменения полуобратной загрузки для двух приоритетов $k_1 = 4, k_2 = 1$. На рисунке 1 использован немодифицированный алгоритм локального голосования. На рисунке 2 разделение временных слотов внутри узла выполнено с учетом приоритетов, однако обмен слотов между узлами использует значение u из немодифицированного протокола локального голосования. На рисунке 3 показаны графики для алгоритма локального голосования, учитывающего приоритеты заданий.

Как можно видеть, в модифицированном алгоритме задания более высокого приоритета завершаются быстрее, чем задания низкого приоритета, и при этом не наблюдается «голодание» задач низкого приоритета.

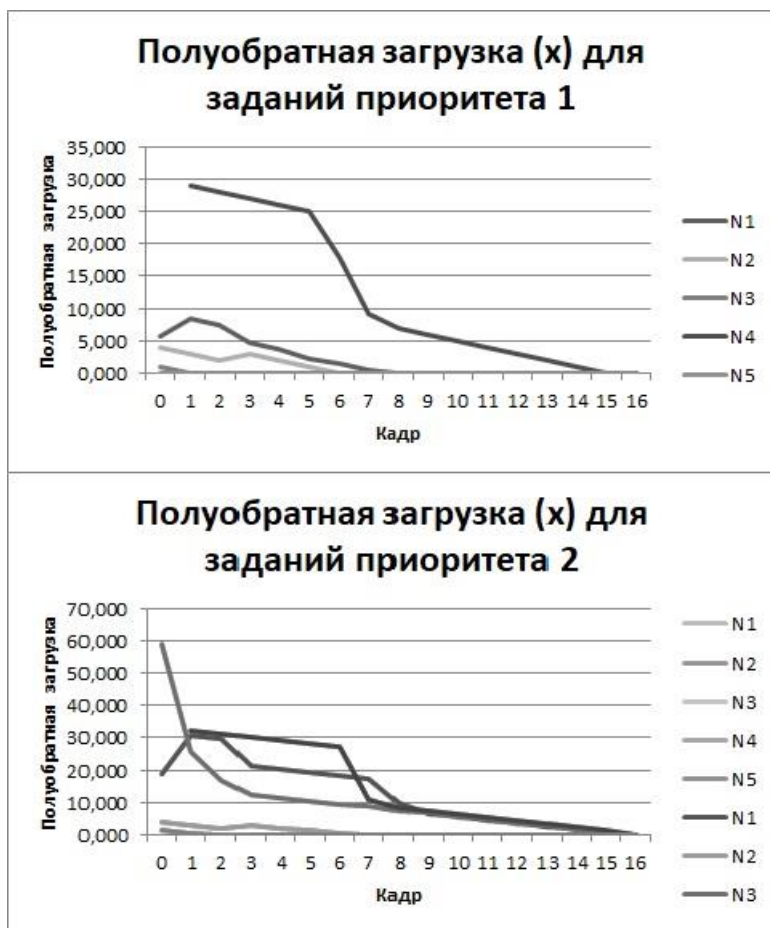


Рисунок 1: Изменение полуобратной загрузки с течением времени для стандартного протокола локального голосования

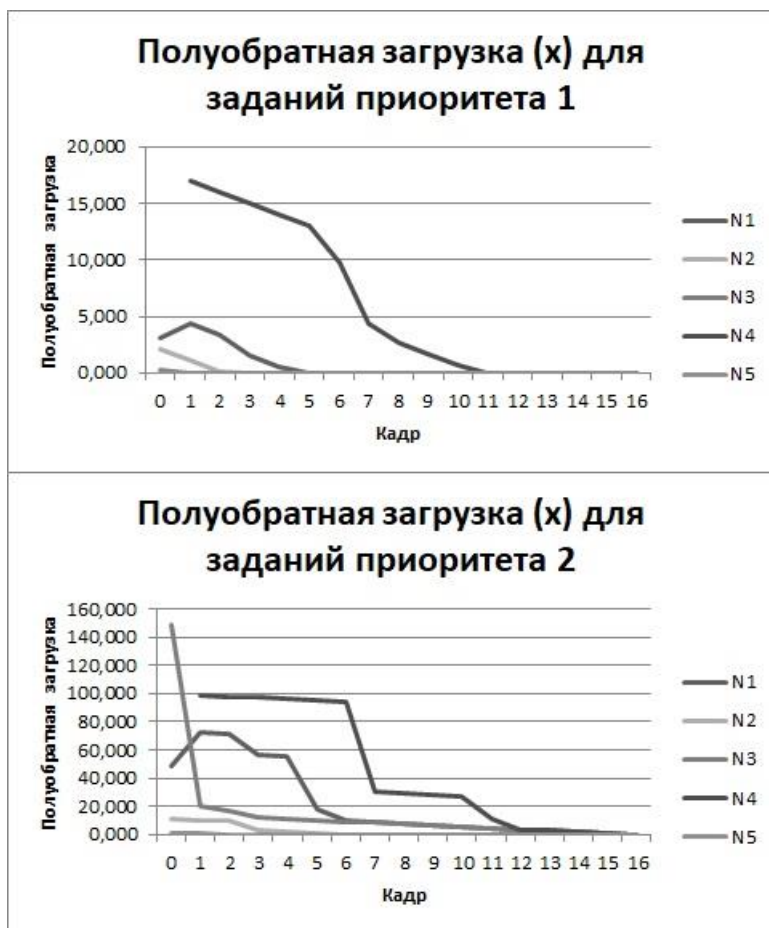


Рисунок 2: Изменение полуобратной загрузки с течением времени для стандартного протокола локального голосования, учитывающего приоритеты при исполнении заданий

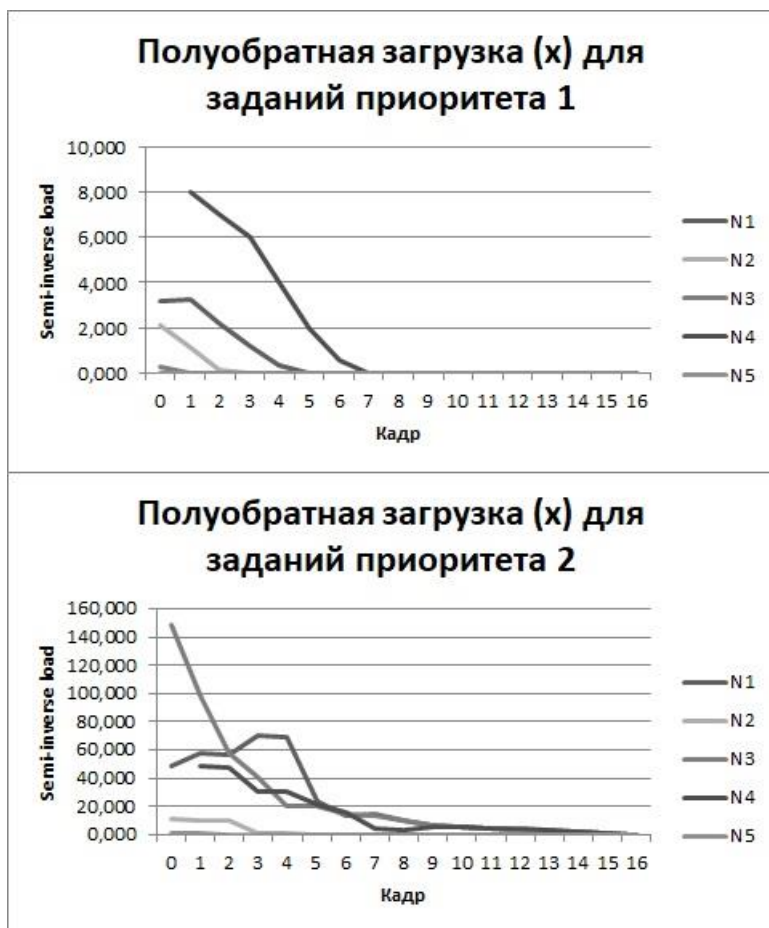


Рисунок 3: Изменение полуобратной загрузки с течением времени для стандартного протокола локального голосования с учетом приоритетов заданий

Заключение

Представлена модификация протокола локального голосования для распределения временных слотов в беспроводной сети с учетом приоритетов заданий.

Реализован прототип системы, моделирующей распределение временные слоты в беспроводной сети протоколом локального голосования с учетом приоритетов заданий и без.

Проведено сравнение стандартного и модифицированного алгоритмов по значению полуобратной загрузки в узлах сети.

Литература

1. I. Rhee, A. Warrier, 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
2. W.-P. Lyui Design of a new operational structure for mobile radio networks // Ph.D. dissertation, Clemson Univ., Clemson, SC, 1991
3. 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
4. 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
5. Vergados D.J., Amelina N., Jiang Y., Kravetska K., Granichin O. Local voting: optimal distributed node scheduling algorithm for multihop wireless networks // In: 2017INFOCOM-Workshop-Proceedings, Atlanta, GA, USA, 1-4 May 2017, 2017, pp. 931-932
6. Vergados D.J., Amelina N., Jiang Y., Kravetska K., Granichin O. Towards optimal distributed node scheduling in a multihop wireless network through Local Voting // IEEE Transactions on Wireless Communications, Jan 2018, vol. 17, issue 1, pp. 400-414