

УДК 004.3.067

НОВЫЙ МЕТОД ОЦЕНКИ АЛГОРИТМОВ РАСПРЕДЕЛЕНИЯ OVSF-КОДОВ В СТАНДАРТЕ WCDMA

И. А. Мельник,
ведущий инженер-программист
ООО «Научно-производственная фирма "Беркут"»

OVSF (Orthogonal Variable Spreading Factor) коды используются для поддержки разных скоростей передачи данных для различных пользователей в WCDMA системах. В данной работе, проанализированы существующие алгоритмы распределения кодов и решаемые ими проблемы. Для того чтобы оценить эффективность алгоритмов, мы предлагаем метод, который учитывает возникающие блокировки между кодами во время распределения. На базе этого метода можно построить аналитические модели для оценки алгоритмов.

Orthogonal Variable Spreading Factor (OVSF) codes are used to support different transmission rates for different users in WCDMA systems. In this paper, we analyze existing code allocation algorithms and problems that they have to solve. To estimate the efficiency of the algorithms we propose a method, which consider blockings between codes during assignment. On basis of this method it is possible to build analytical models for an estimation of different algorithms.

Введение

WCDMA (Wideband CDMA) – системы третьего поколения (3G), которые поддерживают переменную скорость передачи данных для приложений с различным «качеством услуги» (QoS – Quality of Service). Появление 3G систем было связано с появлением возможности передачи данных на больших скоростях (до 2 Мбит/с). В любой CDMA системе множественный доступ к каналам передачи данных обеспечивается с помощью кодов, при этом полоса частот – одна на всех пользователей. Поэтому схемы управления кодами являются составляющей частью системы, и главной целью их является эффективное использование ресурсов сети. В данной статье предлагается вероятностный метод оценки алгоритмов распределения OVSF-кодов в стандарте WCDMA.

Каналообразующие коды в WCDMA

WCDMA – это телекоммуникационная система, которая разделяется на две основные части: радиосеть (UTRAN) и проводная сеть. В радиосети передаваемые пакеты информации распределяются с помощью RNC (Radio Network Controller), а именно, распределение представлено MAC-протоколом. MAC распределяет пакеты данных в оба выделенных канала: контрольный канал (DPCCH – Dedicated

Physical Control Channel) и канал данных (DPDCH – Dedicated Physical Data Channel), а также в разделяемый канал (PDSCH – Physical Dedicated Shared Channel, далее DSCH). В DSCH может происходить распределение каналообразующих кодов между подключенными абонентами или мобильными станциями каждый раз для фрейма данных [1].

Отображение каналов в системе следующее. Логические каналы отображаются на транспортные, которые характеризуются тем, как информация будет передаваться по радиоинтерфейсу. Они предоставляются физическим уровнем для MAC-уровня. Транспортные каналы отображаются в физические (DPCH – Dedicated Physical Channel) на физическом уровне. В DPCH транспортные каналы мультиплексируются с помощью каналообразующих кодов. Далее информация из всех DPCH суммируется и скремблируется с помощью кода Голда. Одна базовая станция использует один код Голда, т. е. с помощью кодов Голда разделяются между собой сигналы базовых станций в прямом канале (от базовой станции к мобильной). Физический уровень контролируется с помощью RRC (Radio Resource Controller) [2].

В WCDMA в качестве каналообразующих кодов используются ортогональные OVSF-коды (Orthogonal Variable Spreading Factor), которые обеспечивают множественный доступ, а также используются для

поддерживаем с подходящим заблокированным корневым кодом, выделение которого произойдет в соответствии со стратегией выдачи кодов). Помимо этого, алгоритм перераспределения должен определить, куда будут перемещены занятые коды (т. е. поиск свободных кодов в соседних частях кодового дерева, в соответствии со стратегией выдачи кодов, с SF, равными SF занятых потомков подходящего заблокированного кода).

Основными можно назвать три стратегии [6]:

1. Случайный доступ (random).
2. Использование самого левого (leftmost).
3. Самый заполненный – первый (crowded-first).

Первая стратегия представляет собой случайный доступ к узлам дерева с требуемым SF с равными вероятностями. Если при поступлении вызова есть несколько свободных кодов с нужным SF, то случайным образом выбирается какой-нибудь код. В противном случае запрос отклоняется.

Суть второй стратегии состоит в том, что при наличии нескольких подходящих кодов выбирается самый левый. Другими словами, правая часть дерева оставляется для кодов с меньшим SF, или в правой части концентрируется вся оставшаяся емкость дерева.

В случае использования третьей стратегии при наличии нескольких подходящих кодов выбирается тот, чье поддерево до ближайшего корня заполнено в большей степени. При повторном возникновении конфликта уровень корня понижается, и сравниваются полученные поддеревья. Если конфликт не разрешен с учетом описанного алгоритма, то выбирается либо самый левый, либо самый правый коды, в зависимости от настройки.

Стратегии перечислены в порядке увеличения эффективности и увеличения вычислительных затрат при реализации.

Примеры других стратегий и алгоритмов можно найти в работах [1], [7] – [10].

Вероятностный метод оценки

Определим состояние как количество занятых кодов на уровнях дерева. Предположим, что имеется дерево из четырех уровней. Тогда состояние – это вектор из K_i элементов, $i = 1, 2, 3, 4$; K_i – число занятых кодов на уровне i . Увеличение значения K_i или переход в другое состояние означает выдачу кода, уменьшение – высвобождение. Первый переход происходит с интенсивностью запросов с QoS, требующих коды на различных уровнях дерева. Второй переход происходит с интенсивностью обработки вызова или, другими словами, зависит от среднего времени использования абонентом услуги (среднего времени занятости кода).

Прием запроса ограничивается емкостью дерева или выражением

$$K_1 + 2K_2 + 4K_3 + 8K_4 \leq C_{\max}, \quad (1)$$

где C_{\max} – количество листов кодового дерева; K_i – коды с наибольшим SF и наименьшей скоростью (листья).

В начальном состоянии все значения элементов K_i равны нулю. Построим граф возможных переходов из начального состояния. Пусть поток запросов кодов, поступающий в систему, – простейший. Простейший поток характеризуется следующими свойствами.

1. Интенсивность потока постоянна и не зависит от времени.

2. События в потоке появляются по одному, а не группами, одновременно по несколько событий сразу.

3. События, образующие поток, появляются в те или иные моменты времени независимо друг от друга; каждое из них вызвано своими собственными причинами.

В этом случае переход из состояния в новое состояние означает увеличение или уменьшение какого-то элемента K_i на единицу. Интенсивность переходов при увеличении значения K_i может быть неодинаковой и зависеть от скоростных характеристик кода или от услуг, запрашиваемых абонентами. Сопоставим каждому переходу, или ребру графа, вероятность перехода, или вероятность запроса абонентом услуги, которая предоставляется с помощью кода с определенными скоростными характеристиками. Сумма вероятностей переходов из состояния в другие возможные состояния равна единице.

В случае, когда используется алгоритм перераспределения кодов, то можно сказать, что если при поступлении запроса емкость дерева позволяет «вместить» его, то он будет обработан с вероятностью 1, т. е. если при поступлении запроса выполняется неравенство (1), то запрос будет обработан.

Вероятность блокировки запросов – это доля времени, в течение которого система находится в состояниях, когда переход в новое состояние нарушает условие (1).

Общая пропускная способность системы – это сумма произведений пропускной способности каждого состояния на долю времени, в течение которого система находится в этом состоянии (предельная вероятность). Пропускная способность состояния – это количество информации, например, в байтах, которое можно передать с помощью кодов, используемых в этом состоянии.

В случае, если алгоритм перераспределения не используется, то даже при выполнении условия (1) возможны ситуации, когда код не выдается. Это может происходить ввиду блокировок, возникающих при выдаче кода. Для отражения состояния по заблокированным кодам, каждому K_i сопоставим значение B_i , которое показывает количество заблокированных кодов на этом уровне. Теперь состояние – это количество занятых кодов и количество заблокированных кодов на каждом уровне дерева, т. е. состояние описывается двумя векторами: $\{K\}_N$, $\{B\}_N$, где N – количество уровней дерева.

При запросе кода на уровне i вероятность перехода на следующий уровень будет равняться нулю или блокировка вызова будет возникать тогда, ког-

да $B_i + K_i = c_i$, где c_i – максимальное количество кодов на уровне i . Соответственно, для принятия вызова необходимо, чтобы выполнялось условие

$$B_i + (K_i + 1) \leq c_i. \quad (2)$$

Выражение (1) при этом дополняется выражением (2), т. е. невыполнение одного из условий влечет за собой отказ в обработке запроса на код.

Вероятность блокировки в этом случае – доля времени, в течение которого система находится в состояниях, переход из которых в следующее нарушит условие (1) или (2).

Выдача кода дерева влечет за собой блокировку всех потомков и родителей кода из-за отсутствия ортогональности между ними. Таким образом, увеличение K_i влечет за собой увеличение значений B на всех уровнях, меньших i (дочерние коды), и может повлечь увеличение значений B на следующих уровнях, больших i (родительские коды).

Обозначим N количество уровней в дереве, n – уровень, на котором происходит выдача кода, $n - 1$ – уровень кодов с наибольшим SF и наименьшей скоростью (листы). Тогда при выдаче кода на уровне n количество блокируемых кодов на уровнях $1 \leq i \leq (N - n)$ (все дочерние) равно $B_i = 2^{N-n-i+1}$. Общее количество блокируемых дочерних кодов равно

$$B_{desc} = \sum_{i=1}^{N-n} 2^{N-n-i+1}. \quad (3)$$

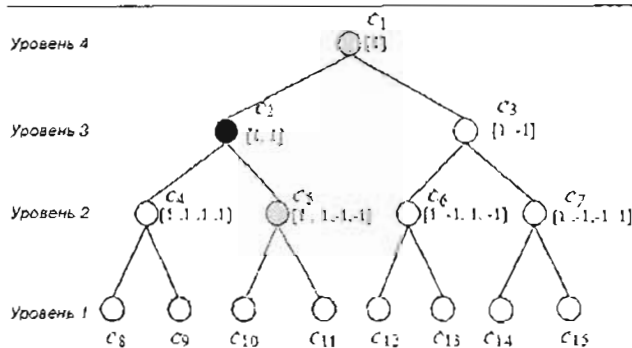
Таким образом, если не учитывать количество блокируемых родительских кодов, то при обработке запроса на код уровня n , т. е. при увеличении K_n , изменение вектора K будет следующим:

$$K_N, K_{N-1}, \dots, K_{n+1}, \dots, K_1.$$

При этом вектор B изменится следующим образом:

$$B_N, B_{N-1}, \dots, B_{N-n} + 2, B_{N-n-1} + 4, \dots, B_1 + 2^{N-n}. \quad (4)$$

Следовательно, выдача любого кода на уровне n всегда влечет за собой блокировку фиксированного количества дочерних кодов, рассчитываемых по формуле (3), при этом вектор B принимает вид (4). Другими словами, количество блокируемых дочерних кодов для каждого уровня фиксировано,



■ Рис. 3. Пример блокировки дочерних кодов при выдаче кода на третьем уровне:
 ● – выделенные коды;
 ◐ – заблокированные коды;
 ○ – свободные коды

или наличие вектора B не увеличивает количество состояний, в которые можно перейти из текущего состояния.

На рис. 3 показан пример дерева, где выдан код c_2 . Количество заблокированных дочерних кодов равно шести. При выдаче кода c_3 количество заблокированных дочерних кодов будет тем же.

Рассмотрим блокировку родительских кодов. Количество блокируемых кодов зависит от того, какие «родственники» кода уже выданы. Интерес представляет любой «родственник» на том же уровне, что и целевой код, у которого родительский код является родителем целевого и имеет наименьшее расстояние (в уровнях) до него. Количество заблокированных кодов выражается формулой

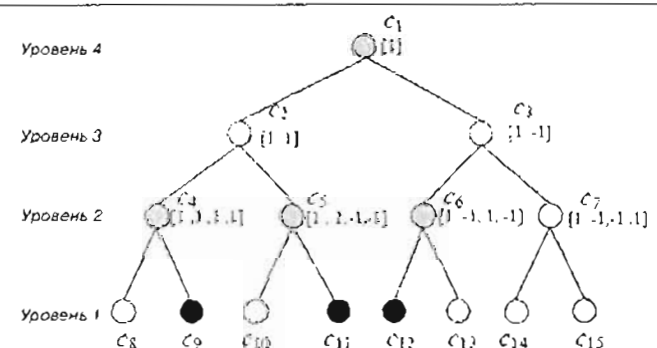
$$B_{asc} = l - i - 1, \quad (5)$$

где i – уровень целевого кода $c_{i,j}$; l – уровень родительского кода c_l , который одновременно является родителем выданного кода $c_{i,k}$. Из всех имеющихся $c_{l,k}$ – занятых кодов на уровне l – выбирается такой c_l , что расстояние от $c_{i,j}$ до c_l минимально

$$\min_l B_{asc} = l - i - 1.$$

В случае, если $c_{i,j}$ – первый выдаваемый код во всем дереве, то количество блокируемых родительских кодов всегда равно $N - i$, где N – общее количество уровней в дереве. Нужно отметить, что на каждом уровне, большем, чем n , может быть заблокирован только один родительский код. Например, если количество блокируемых родительских кодов равно двум, то это означает, что один из двух кодов находится на уровне $n + 1$, а второй – на уровне $n + 2$.

На рис. 4 для кода c_{11} занятыми родственниками на том же уровне являются коды c_9 и c_{12} . Ближайший общий родитель у кодов c_{11} и c_{12} – это код c_1 . Ближайший общий родитель у кодов c_{11} и c_9 – это код c_2 . В качестве l выбираем уровень 3, так как код c_2 ближе к коду c_{11} , чем код c_1 . По формуле (5) рассчитываем, что количество блокируемых родительских кодов при выдаче кода c_{11} , равно 1 – это код c_5 . Если бы код c_9 был свободен, то в качестве l мы выбрали бы уровень 4, и тогда количество блокируемых кодов равнялось бы 2 – это коды c_5 и c_2 .



■ Рис. 4. Пример блокировки родительских кодов при выдаче кода на первом уровне:
 ● – выделенные коды;
 ◐ – заблокированные коды;
 ○ – свободные коды

Определение количества блокируемых родительских кодов при выдаче кода $C_{i,j}$ можно описать следующим алгоритмом

1. Если $C_{i,j}$ – это первый выдаваемый код, то $B_{asc} = N - i$, где N – общее количество уровней в дереве. Переход на шаг 5.

2. Отбросим все уровни, меньшие i , так, чтобы в полученном дереве код $C_{i,j}$ оказался листом.

3. Понизим корень текущего дерева и рассмотрим одно из двух поддеревьев, в котором находится $C_{i,j}$.

4. Есть ли хоть один занятый код на уровне i ?

4.1. Если истина и длина рассматриваемого поддерева равна 2, то $B_{asc} = 0$. Переход на шаг 5.

4.2. Если истина и длина рассматриваемого поддерева больше 2 – переход на шаг 3.

4.3. Если ложь, то $B_{asc} = N - i - 1$, где N – длина рассматриваемого поддерева. Переход на шаг 5.

5. Конец.

При начальном условии, когда все K_i равны нулю (т. е. ни один код не выдан), то количество блокируемых родительских кодов всегда фиксировано и равно $N - i$, где N – общее количество уровней в дереве; i – уровень выдаваемого кода. В случае, если хоть один K_i не равен нулю, то количество блокируемых кодов в общем случае может быть от нуля до $N - i - 1$. Элементы B_k , $k > i$, увеличиваются на единицу у всех уровней из промежутка

$$N \leq k < i.$$

Так как количество блокируемых дочерних кодов всегда фиксировано, то максимальное количество возможных новых состояний (альтернатив для перехода), при вводе вектора B , при выдаче кода на уровне j равно $N - i - 1$, в зависимости от раскладки уже занятых кодов на этом уровне.

Перестраивая граф состояний, получим дополнительные альтернативы перехода из состояния. Например, если дерево имеет четыре уровня и поступил запрос на код второго уровня, то до ввода вектора B была возможна только одна альтернатива, т. е. переход, после которого вектор состояний K имеет элемент K_2 (на единицу больше). После ввода вектора B изменения вектора K остаются теми же, но появляются три новых варианта.

1. Блокировка двух родительских кодов по одному на уровнях 3 и 4.

2. Блокировка одного родительского кода на уровне 3.

3. Ни один родительский код не блокируется.

Первый вариант возможен только в том случае, если выдаваемый код – первый во всем дереве. Таким образом, количество альтернатив перехода такое же, как и до ввода вектора B . Если выдаваемый код – не первый, то появляется дополнительная альтернатива, т. е. в данном случае имеется два возможных варианта перехода вместо одного. По большому счету, количество имеющихся альтернатив определяет минимально возможное количество кодов-кандидатов для выдачи: если есть две альтер-

нативы, то, как минимум, есть два подходящих кандидата для выдачи. Какой из вариантов будет выбран, зависит от используемого алгоритма выдачи кодов. Например, если используется случайный выбор (стратегия random), то вероятность переходов в обоих случаях равна 0,5, или, другими словами, данный алгоритм выбирает код из всех имеющихся кандидатов с равной вероятностью.

Таким образом, мы можем построить граф состояний, вероятности переходов у которого определены используемым алгоритмом распределения кодов, вероятностью запросов кодов различных характеристик, а также средней продолжительностью использования услуг абонентами. Далее, применяя к описанной системе известные формулы из теории массового обслуживания, можно получить характеристики системы, которые будут отражать работу того или иного алгоритма распределения кодов.

Заключение

В данной статье проведен обзор алгоритмов распределения каналовобразующих кодов в стандарте WCDMA. Предложен вероятностный метод оценки алгоритма распределения кодов, который позволяет получить такие характеристики, как блокировка вызова, средняя пропускная способность при использовании того или иного алгоритма. Дальнейшие исследования будут направлены на получение графа состояний для конкретного OVFSF-дерева, на применение к полученной системе формул и оценке имеющихся на текущий момент алгоритмов распределения кодов данным методом.

Литература

1. Legg P., Villier E. Efficient channelisation code allocation in UMTS 2001 (<http://www.priorartdatabase.com/IPCOM/000004839/>).
2. TSG RAN WG1 Physical layer – General description Tech. Spec. TS25.201, 3GPP, December 2003 (<http://www.3gpp.org>).
3. TSG RAN WG1 Spreading and modulation (FDD) Tech. Spec. TS25.213, 3GPP, September 2003 (<http://www.3gpp.org>).
4. Yu C. W., Huang R. O. OVFSF Code Management Schemes on Ad Hoc Networks ICC, 2004, Paris (<http://www.chu.edu.tw/~cwyu/publication.htm>).
5. TSG RAN WG1 Radio resource management strategies Tech. Rep. TS25.922, 3GPP, December 2003 (<http://www.3gpp.org>).
6. Tseng Yu-Chee, Chao Chih-Min Code Placement and Replacement Strategies for Wideband CDMA OVFSF Code Tree Management IEEE Transactions on Mobile Computing 1(4): 293-302 (2002)
7. Chen Wen-Tsuen, Wu Ya-Ping, Hsiao Hung-Chang A Novel Code Assignment Scheme for W-CDMA Systems, Taiwan 300, R.O.C.
8. Dell'Amico M., Merani M. L., Maffio F. A Tree Partitioning Dynamic Policy for OVFSF Codes Assignment in Wideband CDMA, DISMI, DSI, DEI, Italy, March 2003.
9. Park Jun-Seong, Lee D. C. Enhanced Fixed and Dynamic Code Assignment Policies for OVFSF-CDMA Systems, Los Angeles, CA 90089, U. S. A.
10. Rouskas A. N., Skoutas D. N. Comparison of Code Reservation Schemes at the Forward Link in WCDMA, Samos 83200, Greece.