

Системы массового обслуживания

PACS 02.50.-г, 02.50.Еу, 02.50.Фз

© 2009 г. С.Д. АНДРЕЕВ,
Е.В. ПУСТОВАЛОВ,
А.М. ТЮРЛИКОВ, канд. техн. наук
(Санкт-Петербургский государственный университет
аэрокосмического приборостроения)

ДРЕВОВИДНЫЙ АЛГОРИТМ РАЗРЕШЕНИЯ КОНФЛИКТА, УСТОЙЧИВЫЙ К НЕПОЛНОМУ ПОГАШЕНИЮ ИНТЕРФЕРЕНЦИИ

Рассмотрен подход, объединяющий древовидный алгоритм разрешения конфликта на канальном уровне многоабонентной системы связи с процедурой последовательного погашения интерференции на физическом уровне. Предлагается устойчивый к неполному погашению интерференции алгоритм с единичной памятью, стабильный в рамках классической модели множественного доступа с бесконечным числом абонентов. На примере его анализа демонстрируется простая методика расчета скорости, применимая к классу древовидных алгоритмов со свойством последовательного погашения интерференции. Предложенный алгоритм может быть использован как альтернатива заданной в стандарте IEEE 802.16 схеме разрешения конфликтов между запросами на ресурс канала.

1. Введение

Подходы, предполагающие объединение возможностей двух и более уровней эталонной модели взаимодействия открытых систем, позволяют достичь большей эффективности, чем это возможно при традиционном взаимодействии уровней. В частности, достаточно перспективна совместная работа *физического* уровня и подуровня *управления доступом к среде* (УДС), поскольку УДС является наиболее «узким местом» современных телекоммуникационных систем. На сегодняшний день используется множество алгоритмов подуровня УДС [1, 2], что дает разработчику широкие возможности по выбору подходящего решения. Алгоритмы *случайного множественного доступа* (СМД) часто используются при наличии в системе связи трафика со всплесками и позволяют обеспечить достаточно низкую задержку *пакета* данных даже в случае неограниченно большого числа абонентов в системе.

Вообще говоря, каждый алгоритм СМД включает в себя *алгоритм доступа к каналу* (АДК) и *алгоритм разрешения конфликта* (АРК). В то время как первый регламентирует процедуру доступа абонентов к общей среде передачи, второй задает правило для разрешения конфликтов между пакетами данных, которые воз-

никают при одновременной передаче от двух и более абонентов. В семействе алгоритмов, основанных на алгоритме АЛОХА [3], таких как *тактированная АЛОХА с повторением*, *двоичная экспоненциальная «отсрочка»*, *множественный доступ с прослушиванием несущей*, АРК отсутствует. Эти алгоритмы отличаются достаточно простой реализацией, а основная идея их работы заключается в том, что при возникновении конфликта повторная передача пакета откладывается на некоторый случайный промежуток времени.

Напротив, древовидные алгоритмы СМД, независимо предложенные в [4] и [5], определяют АРК и этим достигают более высоких показателей производительности по сравнению с вышеуказанными алгоритмами. В настоящее время принято различать *стандартный древовидный алгоритм* (СДА) и *модифицированный древовидный алгоритм* (МДА). Будем далее называть СДА и МДА *традиционными древовидными алгоритмами*.

При работе традиционных древовидных алгоритмов СМД неявно предполагается, что в случае возникновения конфликта (интерференции) между абонентами на приемной стороне не удастся извлечь никакой осмысленной информации. Однако в последнее время стало возможным использование на физическом уровне процедуры *последовательного «погашения» интерференции* (SIC, Successive Interference Cancellation) [6–9]. Процедура SIC позволяет поочередно восстанавливать попавшие в конфликт пакеты (сигналы) от различных абонентов беспроводного канала связи. Использование процедуры SIC, тем не менее, целесообразно только в восходящем канале централизованных сетей связи, поскольку для работы процедуры необходима информация, доступная на базовой станции [6]. Таким образом, описываемый подход может быть естественно применен в региональных беспроводных сетях связи стандарта IEEE 802.16 [10].

В [11] впервые был предложен алгоритм, сочетающий в себе использование традиционного древовидного алгоритма с преимуществами процедуры SIC (SICTA, SIC in a Tree Algorithm). В той же работе был описан и проанализирован новый *протокол*, в котором процедура SIC обрабатывает принятые сигналы, содержащие информацию о конфликте. Такие *конфликтные сигналы* сохраняются до своей обработки в памяти приемника, которая в [11] предполагается неограниченной. Такой *базовый алгоритм* SICTA по эффективности вдвое превышает традиционный СДА.

Позднее было рассмотрено несколько модификаций базового алгоритма SICTA [12–16], в том числе использующих одну ячейку памяти для хранения сигнала на приемной стороне. В частности, в [13] рассмотрен алгоритм, являющийся альтернативой стандартному алгоритму двоичной экспоненциальной «отсрочки», который регламентирует процесс разрешения конфликтов между запросами на ресурс нисходящего канала в восходящем канале сетей типа IEEE 802.16. Все вышеупомянутые алгоритмы можно условно разделить на две группы: неустойчивые к неполному погашению интерференции, т.е. подверженные эффекту *запирания* из-за ошибок погашения, и устойчивые к неполному погашению интерференции, но неработоспособные в модели с бесконечным числом абонентов (нестабильные). В данной работе отмеченный пробел заполняется предложением достаточно простого для практической реализации устойчивого алгоритма с единичной памятью, стабильного в модели с бесконечным числом абонентов.

Работа построена следующим образом. Во втором разделе формулируются допущения *классической* модели СМД и предлагается ее расширение с учетом особенностей работы процедуры SIC. Вводятся необходимые определения и описываются некоторые алгоритмы. Третий раздел демонстрирует подход к анализу производных от SICTA алгоритмов на примере предложенного алгоритма. В четвертом разделе проводится сравнение полученных аналитических результатов с известными и обсуждаются особенности практической реализации предложенного алгоритма.

2. Модель системы и описание алгоритмов

2.1. Классическая модель системы множественного доступа

Следуя множеству работ, например [1, 2, 17], сформулируем ряд допущений относительно системы связи с СМД и правил функционирования абонентов в ней. Данную совокупность допущений можно считать *классической* моделью системы множественного доступа. Несмотря на ряд упрощающих по отношению к реальности предположений, данная модель позволяет сравнивать между собой различные алгоритмы с единых позиций. Подробное обсуждение описываемых допущений можно найти в [1].

Допущение 1. Синхронная система. Время работы системы подразделяется на равные интервалы времени, называемые *слотами*. Все передаваемые пакеты имеют одинаковую длину, и передача каждого пакета занимает в точности один слот. Слоты нумеруются целыми неотрицательными числами, и слоту с номером t соответствует интервал времени $[t, t + 1)$. Далее будем называть слот с номером t просто слотом t для краткости. Границы слотов известны всем абонентам системы, и каждый абонент может начинать передачу имеющегося у него пакета данных только в начале очередного слота.

Допущение 2. Информативность канала связи. В каждом слоте возможно возникновение одного и только одного из следующих *событий*:

- только один из абонентов передает пакет («УСПЕХ»);
- ни один из абонентов не передает пакет («ПУСТО»);
- два и более абонентов передают пакеты одновременно («КОНФЛИКТ»).

Допущение 3. Тип обратной связи. Информация об одном из трех произошедших в слоте событий (см. допущение 2) поступает на подуровень УДС абонента к *концу* данного слота.

Допущение 4. Достоверность обратной связи. Информация *обратной связи* (см. допущение 3) поступает на подуровень УДС абонента *достоверно*.

Допущение 5. Бесконечное число абонентов. В системе имеется *бесконечное* число абонентов, у каждого из которых имеется буфер для хранения одного пакета данных. Пакет хранится от момента поступления к абоненту до момента окончания его успешной передачи. Интервалы времени между двумя следующими один за другим моментами появления пакетов данных являются статистически независимыми случайными величинами, распределенными экспоненциально с параметром $\frac{1}{\lambda}$, где λ – *интенсивность* входного потока новых пакетов в систему. Каждый новый пакет поступает к новому абоненту, т.е. пакеты данных и абоненты, к которым они поступили, тождественны.

2.2. Традиционные древовидные алгоритмы

Напомним, что каждый древовидный алгоритм задает АДК и АРК, регламентирующие соответственно порядок передачи пакетов в канал и процесс разрешения возникающих при этом конфликтов. Начнем их описание с последнего. При возникновении конфликта в слоте t абоненты, вступившие в конфликт, разбиваются на два подмножества. Например, каждый из участвующих в конфликте абонентов «бросает (неправильную) монету» и выбирает первое подмножество с вероятностью p и второе с вероятностью $(1 - p)$. Абоненты первого подмножества передают свои пакеты в слоте $t + 1$, и если этот слот был пуст или в нем была успешная передача, то абоненты второго подмножества передают в слоте $t + 2$. В противном случае, если в слоте $t + 1$ возникает еще один конфликт, первое подмножество снова разбивается

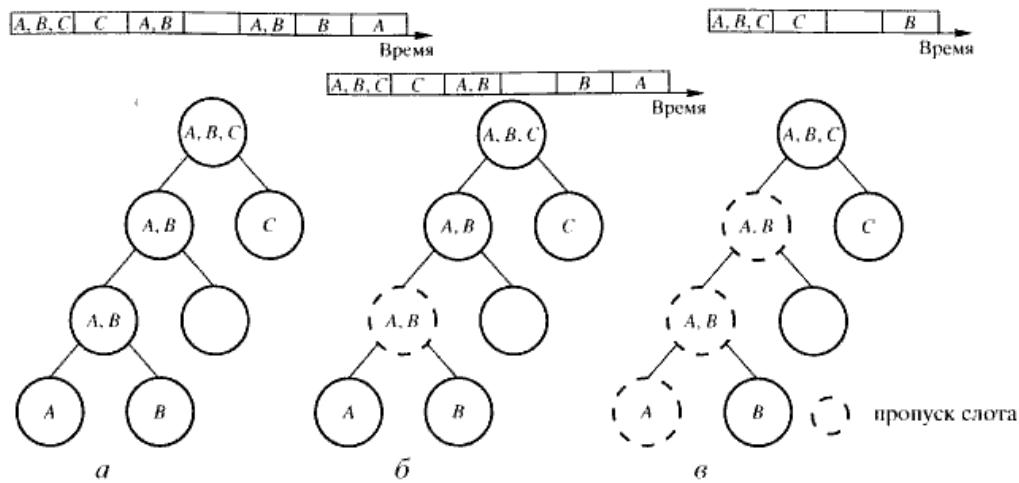


Рис. 1. Примеры работы: *a* – СДА, *б* – МДА, *в* – SICTA.

на два и для них повторяются описанные действия, тогда как второе «ожидает» разрешения конфликта между абонентами первого подмножества.

Описанный АРК удобно представлять в виде двоичного дерева (см. рис. 1, *a*), в котором корневая вершина соответствует множеству абонентов, вступивших в первоначальный конфликт. Остальные вершины соответствуют подмножествам (возможно, пустым) абонентов, передающих свои пакеты в каждом слоте *периода разрешения конфликта* (ПРК), т.е. до завершающего успеха или пустого окна. Ребра дерева отражают процесс разбиения, т.е. из вершин с двумя и более абонентами «вырастает» по две ветви. При этом *правая* вершина такого поддерева соответствует первому подмножеству при разбиении, а *левая* – второму. Поскольку каждая вершина дерева однозначно определяет слот, в котором передает ее подмножество абонентов, будем далее для краткости использовать термины *левый* и *правый слот дерева разрешения конфликта* (ДРК).

Заметим, что в приведенном в качестве примера ДРК (см. рис. 1, *a*) конфликт в слоте 5 неизбежен, поскольку за конфликтом в слоте 3 следует пустой правый слот 4, что свидетельствует о выборе всеми вступившими в конфликт абонентами левого поддерева. Следовательно, целесообразно пропускать слот с неизбежным конфликтом (слот 5) и переходить непосредственно на следующий уровень ДРК. Совокупность описанных выше правил задает АРК для традиционного МДА (см. рис. 1, *б*) в рамках классической модели системы СМД. Для формализации АДК необходимо определить правила, в соответствии с которыми передают свои пакеты абоненты, не участвующие в разрешении конфликта.

Каждый древовидный алгоритм может использовать один из трех альтернативных АДК: *шлюзовой*, *оконный* или *неблокированный*. При работе шлюзового АДК передача новых пакетов данных, поступающих в течение некоторого ПРК, откладывается до начала следующего. Как только текущий конфликт разрешен, все отложенные пакеты передаются одновременно. Если таких пакетов два и более, неизбежно возникает новый конфликт, который дает начало соответствующему ПРК. Оконный АДК представляет собой обобщение схемы с шлюзовым доступом для случая, когда новый ПРК формируется не с участием всех отложенных на предыдущем ПРК пакетов, а лишь их части, поступившей в течение некоторого временного окна. При надлежащем управлении размером данного окна удастся повысить эффективность работы алгоритма. Алгоритмы с шлюзовым и оконным доступом часто называются *блокированными* алгоритмами или алгоритмами с *блокированным доступом*. Наконец, в неблокированных алгоритмах новый пакет данных передается непосредственно вслед за своим поступлением. При этом понятие «ПРК» не возник-

кает, а алгоритм становится проще для реализации. Это происходит в силу того, что теперь включение нового абонента в систему не требует предварительного наблюдения событий, происходящих в канале связи. Неблокированные алгоритмы иногда именуют алгоритмами со свободным доступом.

Одной из важнейших характеристик любого древовидного алгоритма является его скорость. Максимальная *скорость* стабильной передачи определяется как наименьшая верхняя граница интенсивностей (см. допущение 5) входного потока новых пакетов в систему, при которых система стабильна. Система с множественным доступом считается *стабильной* при заданной интенсивности входного потока, если средняя задержка пакета (предел или среднего по времени, или среднего по ансамблю, когда время стремится к бесконечности) конечна [1]. Традиционный МДА с шлюзовым доступом [4,5] достигает скорости в 0,375 при «справедливом» разбиении абонентов с $p = \frac{1}{2}$. Однако для данного алгоритма оптимальным является «несправедливое» разбиение с $p = 0,582$, что позволяет получить скорость в 0,381 [18]. Несмотря на увеличение скорости при оптимизации параметра p , в данной работе для простоты анализа ограничимся рассмотрением лишь алгоритмов с шлюзовым доступом и справедливым разбиением. Некоторые замечания относительно неблокированного доступа будут даны в заключительном разделе.

2.3. Последовательное погашение интерференции

Развитие телекоммуникационного оборудования привело к возможности использования процедуры последовательного погашения интерференции (SIC) на физическом уровне сетей связи [6–9]. Для понимания того, как процедура SIC позволяет повысить эффективность работы древовидного алгоритма, вслед за [12] рассмотрим простой пример на рис. 2. Пусть в некотором слоте t два абонента передают пакеты A и B , что приводит к конфликту. Предположим также, что шумы в канале связи отсутствуют. За y_t обозначим сигнал, принятый к концу слота t , а за x_A – сигнал, соответствующий пакету A . Получив сигнал $y_2 = x_A$ в конце второго слота, приемник с SIC успешно декодирует пакет A . Далее процедура SIC нейтрализует интерференцию, действующую со стороны сигнала x_A на сигнал x_B в первом слоте, т.е. $x_B = \tilde{y}_1 = y_1 - x_A$. Таким образом, пакет B также восстанавливается успешно, и дальнейшее разрешение конфликта не требуется. В рассмотренном примере длительность ПРК составляет на один слот меньше, чем аналогичная длительность при работе любого традиционного древовидного алгоритма.

Заметим, что наиболее часто процедура SIC применяется для итеративного декодирования конфликтного (составного) сигнала [6,9]. В частности, приемник с SIC вначале выделяет наиболее мощный сигнал в принятом составном сигнале и декодирует его. Затем данный сигнал снова кодируется и «вычитается» из исходного. Таким образом, возникает новый составной сигнал, и процедура погашения интерференции выполняет следующую итерацию.

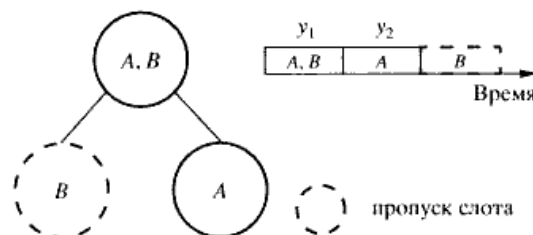


Рис. 2. Пример функционирования процедуры погашения интерференции.

Необходимость кодирования сигнала на приемной стороне значительно ограничивает применимость вышеописанной процедуры SIC. Для функционирования рассматриваемых ниже алгоритмов такое сильное ограничение не требуется. В частности, достаточная для работы базового алгоритма SICTA [11] структура приемопередатчика с процедурой SIC описана в Приложении п. В. Из данного описания следует, что в реальных системах связи информация о произошедшем в слоте событии не поступает непосредственно на подуровень УДС абонента. В силу особенностей приемопередающего оборудования данная информация модифицируется физическим уровнем. Вообще говоря, объем обратной связи, поступающей от физического уровня на подуровень УДС, может быть сохранен, расширен или сужен.

В алгоритмах, основанных на использовании процедуры SIC, объем обратной связи, доступной на подуровне УДС абонента, расширен по сравнению с троичной. Это связано с тем, что используется дополнительная информация, полученная при погашении интерференции. Видоизменим допущение 3 о типе обратной связи с учетом данной особенности.

Допущение 3'. *Обратная связь при погашении интерференции.* На физическом уровне абонента функционирует процедура последовательного погашения интерференции. Объем обратной связи, доступной на подуровне УДС абонента, расширяется с учетом возможностей данной процедуры. Информация обратной связи по-прежнему доступна к концу текущего слота.

Для своей работы базовый алгоритм SICTA [11] требует от физического уровня расширенной обратной связи типа K -«ПУСТО»-«КОНФЛИКТ», где K представляет собой число успешно декодированных пакетов (сигналов), сложенное с числом левых слотов в ДРК, помеченных как пустые после выполнения процедуры SIC (подробнее см. Приложение, п. А.1). Напомним, что для надлежащей работы SICTA на приемной стороне предполагается наличие неограниченной памяти для хранения принятых из канала связи конфликтных сигналов. Дополним модель системы допущением 6 об объеме сигнальной памяти.

Допущение 6. *Объем сигнальной памяти.* На физическом уровне абонента имеется неограниченное количество ячеек памяти для хранения принимаемых из канала связи сигналов.

Рассмотрим пример работы базового алгоритма SICTA на рис. 1,6, где ПРК длится всего лишь 4 слота. После успешного приема во втором слоте ПРК сигнала $y_2 = x_C$ содержимое соответствующего левого слота определяется в результате погашения интерференции. Такая операция по погашению интерференции обозначалась выше как $\tilde{y}_1 = y_1 - x_C$. В конце второго слота на подуровень УДС поступит обратная связь $K = 1$ от физического уровня. Поскольку слот 3 пуст, соответствующий ему левый слот пропускается по правилам МДА. Наконец, успешный прием сигнала x_B в слоте 4 позволяет выделить сигнал x_A как $x_A = \tilde{y}_1 = \tilde{y}_1 - x_B = y_1 - x_C - x_B$. Поскольку при работе SICTA левое поддерево алгоритма СДА пропускается целиком, скорость SICTA составляет 0,693, т.е. в точности равна удвоенной скорости СДА. Более строгий анализ скорости алгоритма SICTA будет проведен в разделе 3.

2.4. Ограничение памяти на приемной стороне

Наличие на приемной стороне потенциально неограниченного объема памяти для хранения принятых конфликтных сигналов практически нереализуемо. С учетом данного факта в [15] впервые была предложена модификация алгоритма SICTA с единичной памятью. Поскольку данный случай представляет наибольший интерес для практики в силу простоты реализации процедуры SIC, модифицируем допущение 6 об объеме сигнальной памяти.

Допущение 6'. *Единичная сигнальная память.* На физическом уровне абонента имеется одна ячейка памяти для хранения принимаемых из канала связи сигналов.

Важно отметить, что в [15] к каждому передаваемому пакету данных предлагалось добавлять один бит (дополнительное управляющее поле), в котором указывалось, передается пакет впервые или повторно. Такое изменение системы допущений приводит к несравнимости полученного в [15] значения скорости со скоростью СИСТА. Наконец, в [16] был рассмотрен наилучший на сегодняшний день алгоритм СМД – алгоритм *разбиения*, обладающий свойством *передачи в порядке поступления* (ППП или FCFS, First Come First Served) [19] и достигающий скорости в 0,4871 для оконного АДК. Были предложены две модификации данного алгоритма с использованием процедуры SIC, которые в рамках рассматриваемой модели имеют скорости 0,6048 и 0,6173 соответственно.

Первая модификация *с погашением при успехе* (FCFS/SC, FCFS with Success Cancellation) позволяет вычитать из конфликтного сигнала, хранящегося в памяти, только успешно принятый сигнал. Таким образом, работа процедуры SIC по сравнению с описанной в Приложении, п. В существенно упрощается. Вторая модификация *с погашением при успехе и конфликте* (FCFS/SCC, FCFS with Success and Collision Cancellation) позволяет вычитать из хранимого сигнала как успешный, так и конфликтный принятый сигнал. Данной особенностью объясняется увеличение скорости алгоритма. Оно достигается за счет некоторого усложнения работы физического уровня приемника, так как на практике можно ожидать, что вычитание двух конфликтных сигналов более проблематично. Тем не менее ограничимся в дальнейшем рассмотрением модификаций второго типа, так как именно они позволяют реализовать возможности процедуры SIC в полной мере.

К сожалению, основным недостатком основанных на алгоритме разбиения с ППП алгоритмов является требование наличия чрезвычайно точной службы отсчета времени в системе связи, что исключительно сложно для реализации. Временное разрешение должно быть таково, чтобы любая пара поступивших пакетов имела различные временные метки. В противном случае, работа алгоритма разбиения с ППП невозможна [1]. Кроме того, все упомянутые в данном подразделе алгоритмы с единичной памятью [15, 16] неустойчивы к неполному погашению интерференции, о котором пойдет речь ниже.

2.5. Учет неполного погашения интерференции

Отметим, что в реальных приемных устройствах с процедурой SIC [7] возможно появление ошибок погашения интерференции, т.е. остаточных сигналов после вычитания принятого сигнала из исходного составного сигнала. К примеру, после вычитания сигнала x_A из составного сигнала $x_A + x_B$ в некотором слоте t результирующий сигнал содержит $\tilde{y}_t = x_B + n_A$, где n_A – остаточный сигнал x_A . После вычитания сигнала x_B аналогично имеем $\tilde{y}_t = n_A + n_B$.

Если мощность сигнала $n_A + n_B$ достаточно высока, приемник ошибочно принимает решение о том, что данный слот не пуст, т.е. считает, что произошел несуществующий конфликт между абонентами. Для простоты будем полагать, что в силу ошибок погашения интерференции рассматриваемое событие возникает с некоторой постоянной вероятностью, зависящей от особенностей приемника. Иными словами, после очередной операции по погашению интерференции приемник не декодирует сигнал успешно с заданной вероятностью. Для учета рассматриваемых ошибок модифицируем допущение 4 о достоверности обратной связи.

Допущение 4'. *Неточная обратная связь.* В силу неполного погашения интерференции информация обратной связи, поступающая на подуровень УДС абонента, *неточна*. Неточность вызвана ошибками при декодировании сигналов. В частности, после вычитания успешно принятого сигнала из исходного получаем ошибку декодирования сигнала с вероятностью q . Аналогично, вычитая конфликтный сигнал из исходного, имеем ошибку декодирования сигнала с вероятностью q' .

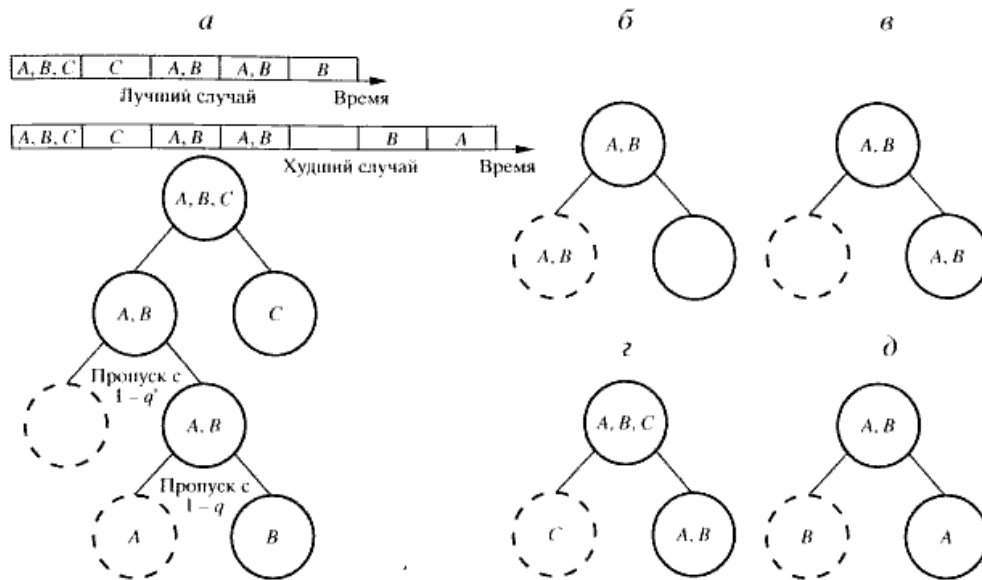


Рис. 3. а - пример работы R-SICTA/SCC; б, в, г, д - некоторые поддеревья ДРК R-SICTA/SCC.

На практике можно ожидать, что $q' \geq q$, т.е. при вычитании конфликтного сигнала чаще возникают ошибки декодирования сигнала.

Несмотря на свою высокую скорость, описанный в Приложении, п. А.1 базовый алгоритм SICTA уязвим к ошибкам погашения интерференции. Действительно, предположим, что в примере на рис. 1, в последняя операция по вычитанию сигнала x_B из первого сохраненного сигнала дала некоторый остаточный уровень шума n_B , т.е. $\tilde{y}_1 = y_1 - x_C - x_B + n_B$. При достаточно высоком уровне n_B сигнал x_A может не быть восстановлен успешно и процесс разрешения конфликта продолжится. Рано или поздно, когда пакет A все же будет передан успешно, остаточный уровень шума может оказаться настолько велик, что приемник определит наличие еще одного конфликта в левом слоте. Согласно правилам работы алгоритма, процесс разрешения конфликта между несуществующими абонентами будет продолжаться до тех пор, пока не будет искусственно прерван извне. Таким образом, возникает эффект *запирания*.

Проделанный анализ использующих SIC алгоритмов позволяет заключить, что на сегодняшний день отсутствует устойчивый к неполному погашению интерференции алгоритм, стабильный в рамках классической модели множественного доступа с бесконечным числом абонентов (допущение 5). Предлагаемый в данной работе алгоритм (см. описание в Приложении, п. А.2) устойчив к ошибкам погашения интерференции, что достигается ценой некоторого снижения скорости его работы. Кроме того, он использует единичную память на приемной стороне по аналогии с подходами из [15] и [16]. Увеличение объема доступной памяти приведет к росту скорости алгоритма (которая ограничена сверху скоростью SICTA, равной 0,693), но затруднит его практическую реализацию. Анализ работы алгоритмов, имеющих более одной ячейки памяти, является самостоятельной исследовательской задачей, выходящей за рамки данной работы.

Основная идея алгоритма (см. рис. 3, а) заключается в отказе от пропуска некоторых конфликтных слотов (таких как слот 3), пропуск которых мог бы привести к эффекту запирания. Ниже будем называть данный алгоритм *устойчивым SICTA с погашением при успехе и конфликте* (R-SICTA/SCC, robust SICTA with Success and Collision Cancellation). На рис. 3, а временная диаграмма для лучшего случая соответствует двум успешным операциям по погашению интерференции, тогда как диаграмма для худшего случая соответствует двум неуспешным операциям. Фор-

Правило	Канал – У1	Содержимое памяти	У1 – У2	Сохранение
1	«КОНФЛИКТ»	$ss - cs = 0$	C/skip	cs
2	«КОНФЛИКТ»	$ss - cs = ms$	C/skip	cs
3	«КОНФЛИКТ»	иначе	C/-	cs
4	«УСПЕХ»	$ss - cs = ms$	S/skip	0
5	«УСПЕХ»	иначе	SE/-	0
6	«ПУСТО»	$ss \neq 0$	E/skip	ss
7	«ПУСТО»	$ss = 0$	SE/-	0

мальное описание алгоритма приведено в Приложении, п. А.2. Отметим, что ввиду ошибок погашения интерференции правила 3, 5 и 7 (см. таблицу) не должны допускать пропуск слота в ДРК, поскольку такой пропуск ведет к неустойчивой работе алгоритма.

3. Анализ древовидных алгоритмов с погашением интерференции

3.1. Общая процедура вычисления скорости

Опишем подход к вычислению скорости древовидных алгоритмов с последовательным погашением интерференции, модифицируя способ пересчета среднего времени разрешения конфликта для МДА из [17]. Затем покажем применение данного подхода к анализу производных от SICTA алгоритмов. Обозначим через v длительность ПРК в слотах (время разрешения конфликта кратности k , число вершин в соответствующем ДРК), которая представляет собой дискретную случайную величину. Условное математическое ожидание $E[v|разрешается\ конфликт\ кратности\ k]$ определяет среднюю длину ПРК для конфликта с участием k абонентов. Шаги предлагаемой процедуры вычисления скорости алгоритма с процедурой SIC могут быть сформулированы следующим образом:

1. Рассмотрим некоторый древовидный алгоритм A с процедурой SIC. Обозначим среднее время разрешения конфликта кратности k для данного алгоритма T_k^A .

Из [4] следует, что, используя отношение $\frac{k}{T_k^A}$, можно установить следующие пределы для скорости древовидного алгоритма A , которую обозначим R_A :

$$(1) \quad \liminf_{k \rightarrow \infty} \frac{k}{T_k^A} < R_A < \limsup_{k \rightarrow \infty} \frac{k}{T_k^A}.$$

2. Рассмотрим СДА, среднее время разрешения конфликта при работе которого обозначим T_k . По аналогии с (1) запишем пределы для скорости СДА, опуская нижний индекс и обозначая ее просто R :

$$(2) \quad \liminf_{k \rightarrow \infty} \frac{k}{T_k} < R < \limsup_{k \rightarrow \infty} \frac{k}{T_k}.$$

Отметим, что пределы для R были вычислены в [20] и составили:

$$(3) \quad 0,34657320 < R < 0,34657397.$$

В [17] была прослежена взаимосвязь между значениями пределов из (3) и следующими выражениями:

$$(4) \quad \left(\frac{2}{\ln 2} + c \right)^{-1} < R < \left(\frac{2}{\ln 2} - c \right)^{-1},$$

где $c = 3,127 \cdot 10^{-6}$.

3. Считая число вершин в ДРК СДА равным v , имеем $T_k = E[v]$. Число успешных, конфликтных и пустых слотов в течение ПРК обозначим v_s , v_c и v_e соответственно. Поскольку $v_s + v_c + v_e = v$, воспользуемся [17], чтобы получить:

$$(5) \quad \begin{aligned} v_s &= k, \\ v_c &= \frac{v-1}{2}, \\ v_e &= \frac{v+1}{2} - k. \end{aligned}$$

Будем рассматривать ДРК алгоритма A как ДРК СДА, в котором время просмотра некоторых вершин дерева равно нулю в силу функционирования процедуры SIC. Вычислим математическое ожидание числа таких вершин $E[r]$ и, обозначая через u число вершин с ненулевым временем просмотра в ДРК алгоритма A , запишем:

$$(6) \quad E[u] = E[v] - E[r]$$

или

$$(7) \quad T_k^A = T_k - E[r].$$

Подставляя (7) в (1) и преобразовывая, можно выразить пределы для скорости R_A через известные пределы для скорости СДА (3).

Отметим, что первые два шага описанной процедуры являются однотипными для любого алгоритма A . Поэтому в дальнейшем ограничимся выполнением только третьего шага, т.е. нахождением $E[r]$ и получением окончательных пределов для скорости R_A .

3.2. Базовый алгоритм SICТА

В качестве примера рассмотрим вычисление скорости базового алгоритма SICТА [11], подробное описание которого представлено в Приложении, п. А.1. Строгое доказательство можно провести, следуя подходу из [17] и описывая алгоритм SICТА в терминах теории графов. Ниже для краткости ограничимся описанием основной идеи доказательства. Заметим, что при работе алгоритма SICТА содержимое первого слота левого поддерева для любой вершины ДРК всегда устанавливается с помощью процедуры SIC без просмотра соответствующей вершины. Таким образом, для нахождения среднего числа вершин с ненулевым временем просмотра в ДРК SICТА (T_k^S) следует вычесть из среднего числа вершин в ДРК СДА (T_k) половину успешных, конфликтных (кроме начального слота ДРК) и пустых слотов:

$$(8) \quad \begin{aligned} T_k^S &= T_k - \frac{1}{2}E[v_s] - \frac{1}{2}(E[v_c] - 1) - \frac{1}{2}E[v_e] = \\ &= T_k - \frac{1}{2}T_k + \frac{1}{2} = \frac{T_k + 1}{2}. \end{aligned}$$

Таким образом, учитывая (1) и (4), получаем следующие пределы для скорости алгоритма SICТА (R_S):

$$(9) \quad \left(\frac{1}{\ln 2} + c \right)^{-1} < R_S < \left(\frac{1}{\ln 2} - c \right)^{-1}.$$

В заключение анализа отметим, что верхний и нижний пределы скорости лежат достаточно близко друг к другу ввиду небольшого значения константы c , поэтому

скорость алгоритма SICTA удобно записать в виде:

$$(10) \quad R_S \approx \ln 2 \approx 0,693,$$

что совпадает с известным результатом из [11], но результат авторов получен без использования громоздких вычислительных конструкций.

3.3. Предлагаемый алгоритм R-SICTA/SCC

Перейдем теперь к рассмотрению предлагаемого алгоритма R-SICTA/SCC, устойчивого к неполному погашению интерференции. Данный алгоритм вычитает из хранимого в памяти конфликтного сигнала как успешные, так и конфликтные принятые сигналы (см. подраздел 2.4), и устойчив к ошибкам погашения интерференции (см. подраздел 2.5).

Вспомним (допущение 6), что после вычитания успешно принятого сигнала из исходного результирующий сигнал не удается успешно декодировать с вероятностью q , а после вычитания конфликтного сигнала результирующий сигнал не удается успешно декодировать с вероятностью q' ($q' \geq q$). Выразим в соответствии с описанной в подразделе 3.1 процедурой нахождения скорости T_k^{RS} – среднее число вершин с ненулевым временем просмотра в ДПК R-SICTA/SCC. Для этого вычтем из T_k – среднего числа вершин в ДПК СДА – следующие вершины, время просмотра которых равно нулю благодаря процедуре SIC:

- рис. 3,б с вероятностью 1;
- рис. 3,в,г с вероятностью $1 - q'$;
- рис. 3,д с вероятностью $1 - q$.

Учитывая (5), после элементарных преобразований получаем:

$$(11) \quad T_k^{RS} = \left(\frac{1}{2} + \frac{1}{4}q' \right) T_k - \frac{1}{2} + \frac{1}{4}q' + \frac{k}{2} - \frac{1}{2}(q' - q)N_k,$$

где N_k – среднее число конфликтов кратности два в ДПК СДА начальной кратности k . Для завершения анализа необходимо оценить отношение $\frac{N_k}{k}$ при неограниченно больших значениях k . Очевидно, что $N_0 = N_1 = 0$, поскольку в данных ДПК не может быть «конфликтных» вершин. Легко показать, что $N_2 = 2$. В общем случае, с учетом свойств ДПК, для $k > 2$ имеем:

$$(12) \quad N_k = \frac{\sum_{i=1}^{k-1} C_k^i N_i}{2^{k-1} - 1}.$$

Рекуррентное выражение (12) легко вычисляется для любого конечного числа k .

Рассмотрим преобразование Пуассона [21] от последовательности величин $N_0, N_1, \dots, N_i, \dots$, которое обозначим как

$$(13) \quad N(s) \triangleq \sum_{k \geq 0} N_k \cdot \frac{s^k}{k!} e^{-s}, \quad s \in \mathcal{R}.$$

Следуя подходу из [22], можно получить следующее рекуррентное выражение для нахождения $N(s)$:

$$(14) \quad N(s) = 2N\left(\frac{s}{2}\right) + \frac{s^2}{2}e^{-s}.$$

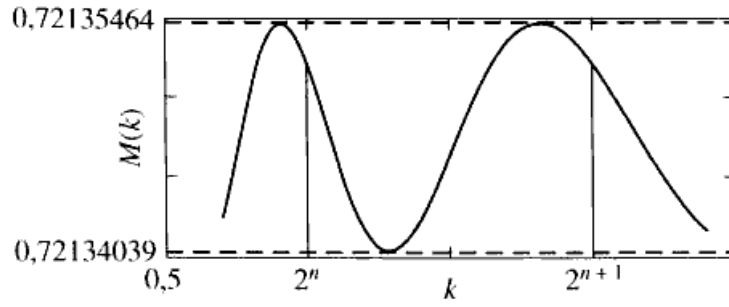


Рис. 4. «Периодический» характер $M(k)$.

Введем в рассмотрение нормированное преобразование Пуассона по аналогии с [20], которое будем обозначать как

$$(15) \quad M(s) \triangleq \frac{N(s)}{s}.$$

Используя (14), можно записать (15) в виде

$$(16) \quad M(s) = M\left(\frac{s}{2}\right) + \frac{s}{2}e^{-s}.$$

Функция $M(s)$ «периодична» при больших значениях своего аргумента, что можно использовать для ее вычисления. Будем вычислять нормированное преобразование Пуассона для достаточно больших значений аргумента $2^n r$, где $n \in \mathcal{Z}$ и $r \in \mathcal{R}$. Формально подставим $2^n r$ в (16):

$$(17) \quad M(2^n r) = M(2^{n-1} r) + \frac{2^{n-1} r}{2} e^{-2^{n-1} r}.$$

При достаточно больших n изменение аргумента функции $M(2^n r)$ от 2^n до 2^{n+1} соответствует одному «периоду» функции $M(2^n r)$. Следовательно, для $1 \leq r \leq 2$ и некоторого n получаем наибольшее и наименьшее значения функции для всех последующих значений ее аргумента. Рассмотрим более подробно равенство (17) и выполним рекуррентный переход $n - 1$ раз:

$$(18) \quad M(2^n r) = M(r) + \sum_{i=1}^n \frac{2^i r}{2} e^{-2^i r} = M(r) + H_n(r).$$

Ряд $H_n(r)$ быстро сходится и его легко вычислить с заданной точностью. Значения $M(r)$ для небольшого r легко вычислять с учетом (13) и (15). Теперь, используя (18), исследуем поведение исходной функции $M(2^n r)$ на одном «периоде», когда $n \geq 20$. Можно показать, что при этом точность значений функции не ниже 10^{-8} . Эти значения будем вычислять в целых точках k , т.е. $M(2^n r) = M(k)$:

$$(19) \quad \max_{2^{20} \leq k \leq 2^{21}} M(k) = \limsup_{k \rightarrow \infty} M(k) < 0,72135464 + 1 \cdot 10^{-8}$$

и

$$\min_{2^{20} \leq k \leq 2^{21}} M(k) = \liminf_{k \rightarrow \infty} M(k) > 0,72134039 - 1 \cdot 10^{-8}.$$

На рис. 4 в логарифмическом масштабе показаны значения функции $M(k)$ на одном «периоде» для достаточно больших k .

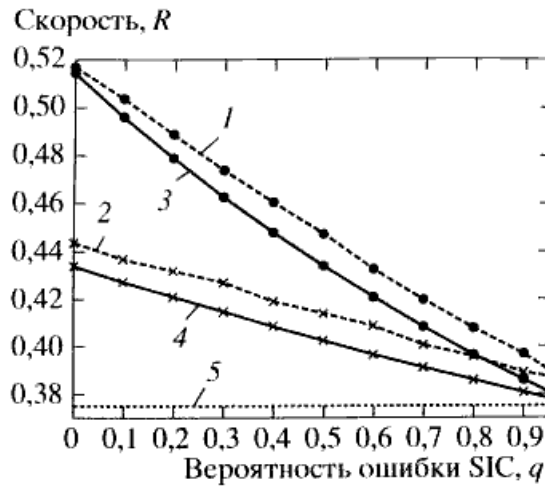


Рис. 5. Скорость R-SICTA/SCC при неполном погашении интерференции: 1 - R-SICTA/FA, $q' = q$; 2 - R-SICTA/FA, $q' = 1$; 3 - R-SICTA/SCC, $q' = q$; 4 - R-SICTA/SCC, $q' = 1$; 5 - МДА.

Следуя подходу из [23] (теорема 1), можно показать, что верхний и нижний пределы для функции $M(k)$ (19) справедливы также для отношения $\frac{N_k}{k}$. Заметим, что поскольку предел отношения $\frac{N_k}{k}$ не существует, неизбежно возникает интервал, на котором невозможно сделать никакого вывода относительно поведения величины $\frac{N_k}{k}$. Длина этого интервала не превышает 0,00001425, т.е. равна разности между верхним и нижним пределами отношения $\frac{N_k}{k}$.

Для упрощения формы представления конечного результата заметим, что $\limsup_{k \rightarrow \infty} \frac{N_k}{k} = \liminf_{k \rightarrow \infty} \frac{N_k}{k} = \gamma$ с точностью как минимум три десятичных разряда и $\gamma = 0,721$. Также для простоты откажемся от нахождения верхнего и нижнего пределов скорости R-SICTA/SCC (R_{RS}), считая, что они лежат достаточно близко друг к другу. Тогда окончательное приближение для скорости предложенного алгоритма можно получить следующим образом:

$$(20) \quad R_{RS} \approx \frac{4R}{2 + q' + 2R(1 - (q' - q)\gamma)}$$

В частности, когда $q' = q = 0$, т.е. при отсутствии ошибок погашения интерференции, $R_{RS} \approx 0,5147$. Подчеркнем два важных частных значения вероятности q' (см. рис. 5). Первое - когда $q' = q$, т.е. вычитание успешных и конфликтных сигналов неразличимо в вероятностном смысле. Тогда скорость алгоритма составляет $\frac{4R}{2 + q + 2R}$. Второе - когда $q' = 1$, т.е. вычитание конфликтных сигналов невозможно. Тогда получаем *устойчивый* алгоритм SICTA с погашением при успехе (R-SICTA/SC, R-SICTA with Success Cancellation), имеющий скорость $\frac{4R}{3 + 2R(1 - (1 - q)\gamma)}$.

4. Выводы. Сравнение алгоритмов

Комбинирование процедуры погашения интерференции на физическом уровне и древовидных алгоритмов на подуровне УДС представляет собой многообещающее

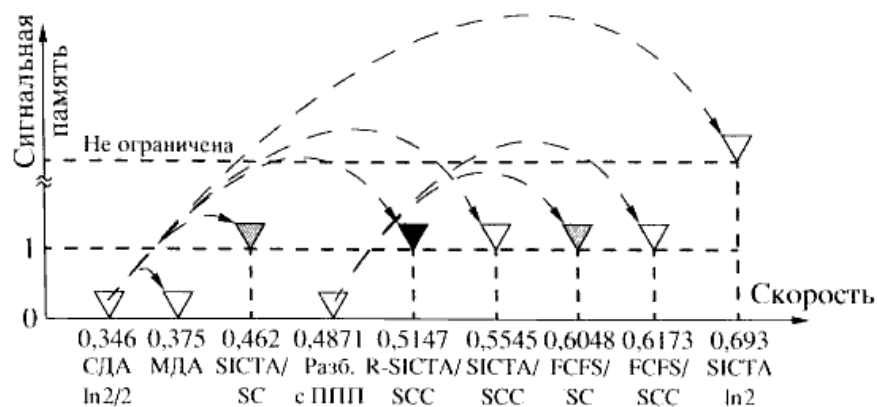


Рис. 6. Сравнение скорости алгоритмов с заблокированным доступом.

направление в развитии систем связи, поскольку позволяет достичь существенного повышения скорости передачи данных при незначительном увеличении сложности оборудования. На сегодняшний день известно целое семейство алгоритмов, следующих данному подходу, из которых базовый алгоритм SICTA имеет наивысшую скорость стабильной передачи 0,693 в модели с бесконечным числом абонентов (см. рис. 6). Однако для своей работы алгоритм SICTA требует неограниченного объема памяти на приемной стороне, что практически нереализуемо. Напротив, выше были отмечены работы, в которых рассматриваются модификации данного алгоритма для единичной памяти приемника. На рис. 6 можно заметить, что все стабильные заблокированные алгоритмы с последовательным погашением интерференции основаны либо на СДА (0,346), либо на алгоритме разбиения с ППП (0,4871).

Практическое функционирование всех решений, основанных на процедуре погашения интерференции, ухудшается вследствие неполного ее погашения. В данной работе модель системы множественного доступа была расширена с учетом этой особенности. Напомним, что базовый алгоритм SICTA, а также модификации алгоритма разбиения с ППП, имеющие ненулевую скорость в модели с бесконечным числом абонентов, при наличии ошибок погашения интерференции подвержены эффекту записывания. Это обусловило необходимость разработки алгоритма, устойчивого к неполному погашению интерференции и стабильного в рамках классической модели множественного доступа с бесконечным числом абонентов.

Предложенная модификация алгоритма SICTA (R-SICTA/SCC), устойчивая к ошибкам погашения интерференции, имеет скорость 0,5147 в случае безошибочной работы процедуры погашения (см. рис. 6). Однако алгоритм работает удовлетворительно даже при высокой вероятности ошибки (рис. 5) и демонстрирует *постепенное* ухудшение характеристик. В худшем случае, когда из-за ошибок погашение интерференции вообще невозможно, он имеет гарантированную скорость 0,375, совпадающую со скоростью МДА.

Видоизменяя правила предложенного алгоритма, описанные в Приложении, п. А.2, можно получить два *неустойчивых* при неполном погашении интерференции алгоритма [24]. Их рассмотрение здесь представляет собой теоретический интерес. Первый алгоритм использует процедуру погашения интерференции только в случае приема успешного сигнала. Второй вычитает как успешные, так и конфликтные сигналы, что приводит к росту скорости алгоритма за счет усложнения физического уровня. С использованием вышеописанного подхода к анализу данных алгоритмов можно получить их скорости, которые составляют 0,462 и 0,5545 для первого (SICTA/SC) и второго (SICTA/SCC) алгоритма соответственно (см. рис. 6). Подробное описание данных алгоритмов, а также определение их скорости, однако, выходит за рамки данной работы.

Заметим для справедливости, что аналогичная идея использования процедуры погашения интерференции может быть применена для модификации непосредственно алгоритма разбиения. Известно, что соответствующие модификации данного алгоритма [16] при единичной памяти приемника имеют скорость 0,6048 (FCFS/SC) и 0,6173 (FCFS/SCC) (см. рис. 6). К сожалению, практическая реализация алгоритма разбиения затруднена необходимостью наличия неограниченно точной службы отсчета времени. Напротив, предложенный алгоритм хорошо сбалансирован по показателям сложности и производительности.

5. Заключение

Отметим, что блокированный алгоритм R-SICTA/SCC, описанный в данной работе, легко видоизменяется для использования с *неблокированным* АДК [25]. Использование свободного доступа не изменяет правил работы, описанных в Приложении, п. А.2, а лишь модифицирует порядок выхода в канал связи новых абонентов, который теперь обладает рядом преимуществ. Во-первых, новые абоненты могут входить в систему без начальной задержки, связанной с ожиданием разрешения текущего конфликта. Во-вторых, неблокированные алгоритмы, как правило, имеют более высокую скорость.

Неблокированная модификация предлагаемого алгоритма (R-SICTA/FA, R-SICTA with Free Access), устойчивая к неполному погашению интерференции, действительно превышает по скорости свой блокированный аналог (см. рис. 5), поэтому в реальной системе связи целесообразно использование именно такой модификации. Вычисление скорости данного алгоритма представляет собой отдельную задачу, также выходящую за рамки данного текста, поэтому на рис. 5 скорость R-SICTA/FA получена методом имитационного моделирования.

ПРИЛОЖЕНИЕ

А. Описание древовидных алгоритмов с процедурой погашения интерференции

А.1. Базовый алгоритм SICTA

Для работы алгоритма SICTA должна быть доступна следующая обратная связь от физического уровня к подуровню УДС приемника:

1. «КОНФЛИКТ»;
2. «ПУСТО»;
3. K – число успешно декодированных пакетов (сигналов), сложенное с числом левых слотов в ДРК, помеченных как пустые после выполнения процедуры погашения интерференции ($K \geq 1$).

Правила функционирования физического уровня прекодера изложены в п. В.

Каждый абонент, участвующий в разрешении конфликта, должен на подуровне УДС отслеживать переменную L_t , соответствующую его уровню в ДРК в слоте t . Абонент передает имеющийся у него пакет данных, только если к началу слота $L_t = 0$. В начале работы алгоритма $L_0 = 0$. В конце некоторого слота t абонент обновляет значение переменной L_t следующим образом с учетом вероятности p :

1. «КОНФЛИКТ»: если $L_t > 0$, то $L_{t+1} = L_t + 1$,
иначе если $L_t = 0$, то $L_{t+1} = \begin{cases} 0, & p, \\ 1, & 1 - p. \end{cases}$

2. «ПУСТО»: если $L_t > 1$, то $L_{t+1} = L_t$,
иначе если $L_t = 1$, то $L_{t+1} = \begin{cases} 0, & p, \\ 1, & 1 - p. \end{cases}$
3. K : $L_{t+1} = L_t - K$.
Если $L_t = 0$, то $L_{t+1} = \begin{cases} 0, & p, \\ 1, & 1 - p, \end{cases}$
иначе если $L_t < 0$, то пакет принят успешно.

А.2. Алгоритм R-SICTA/SCC

Для работы алгоритма R-SICTA/SCC должна быть доступна следующая обратная связь от физического уровня к подуровню УДС приемника:

1. «КОНФЛИКТ» и пропуск уровня (содержимое левого слота извлечено) (C/skip);
2. «КОНФЛИКТ» и нет пропуска уровня (C/-);
3. «УСПЕХ»/«ПУСТО» и нет пропуска уровня (SE/-);
4. «УСПЕХ» и пропуск уровня (содержимое левого слота извлечено) (S/skip);
5. «ПУСТО» и пропуск уровня (неизбежный конфликт в следующем слоте) (E/skip).

Обозначим принятый сигнал cs , хранимый в памяти сигнал ss и некоторый успешно декодированный сигнал ms . Работа физического уровня описана в таблице. Для краткости обозначим физический уровень У1, а подуровень УДС – У2.

Каждый абонент, участвующий в разрешении конфликта, должен на подуровне УДС отслеживать переменную L_t , соответствующую его уровню в ДРК в слоте t . Абонент передает имеющийся у него пакет данных, только если к началу слота $L_t = 0$. В начале работы алгоритма $L_0 = 0$. В конце некоторого слота t абонент обновляет значение переменной L_t следующим образом с учетом вероятности p :

1. C/skip: если $L_t \geq 2$, то $L_{t+1} = L_t$,
иначе если $L_t = 1$, то пакет принят успешно,
иначе $L_{t+1} = \begin{cases} 0, & p, \\ 1, & 1 - p. \end{cases}$
2. C/-: если $L_t > 0$, то $L_{t+1} = L_t + 1$,
иначе $L_{t+1} = \begin{cases} 0, & p, \\ 1, & 1 - p. \end{cases}$
3. SE/-: если $L_t > 0$, то $L_{t+1} = L_t - 1$,
иначе пакет принят успешно.
4. S/skip: если $L_t \geq 2$, то $L_{t+1} = L_t - 2$,
иначе пакет принят успешно.
5. E/skip: если $L_t \geq 2$, то $L_{t+1} = L_t$,
иначе если $L_t = 1$, то $L_{t+1} = \begin{cases} 0, & p, \\ 1, & 1 - p. \end{cases}$

В. Описание процедуры погашения интерференции

На рис. 7 показана упрощенная структура приемопередатчика с погашением интерференции, который может быть использован для реализации базового алгоритма SICTA из [11], описанного в Приложении, п. А.1. На рисунке сплошными стрелками отмечены направления передачи данных, а штриховыми – передачи управляющей информации.

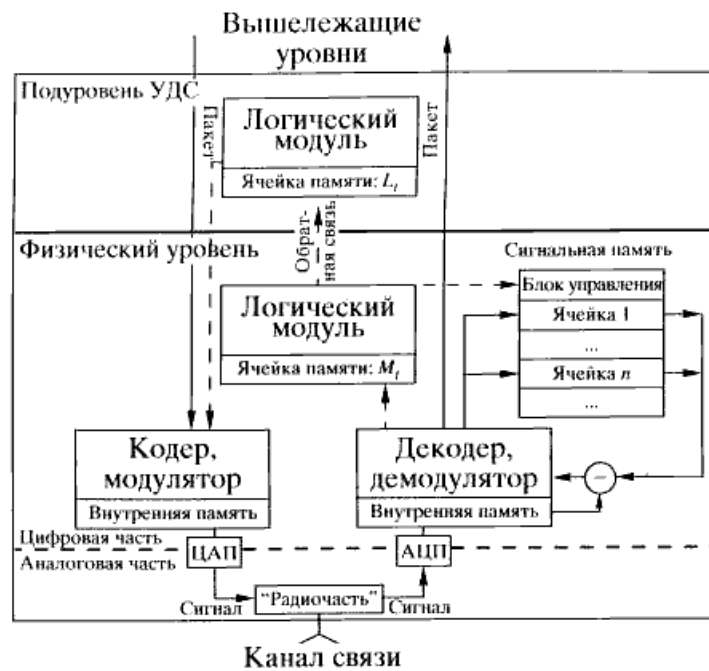


Рис. 7. Упрощенная структура приемопередатчика с погашением интерференции.

Рассмотрим работу приемопередатчика, изображенного на рис. 7, в конце очередного слота. Передающая часть реализована в виде нисходящего тракта данных, поступающих с вышележащих уровней системы связи в виде пакетов. После прохождения *кодера* и *модулятора* цифровые данные преобразуются в аналоговую форму и передаются в канал связи. При этом «*радиочасть*» осуществляет преобразование последовательности отсчетов в радиосигнал и обратно. Заметим, что *логический модуль подуровня УДС*, реализующий работу некоторого древовидного алгоритма, осуществляет управление кодером, указывая последнему моменты времени, в которые следует начинать дальнейшее преобразование данных для передачи. Для этой цели логический модуль отслеживает переменную L_t , имеющую смысл текущего положения абонента в ДРК в слоте t (см. подраздел 2.2 и Приложение, п. А). Абонент передает имеющийся у него пакет данных, только если $L_t = 0$.

Работа принимающей части после преобразования полученного из канала связи аналогового сигнала в цифровую форму может быть описана следующим образом. Предпринимается попытка демодуляции и декодирования принятого сигнала, который для этого временно сохраняется во внутренней (регистрационной) памяти *декодера*. В случае неудачи (был принят конфликтный сигнал), декодер сообщает о результате *логическому модулю физического уровня* и сохраняет оцифрованные отсчеты сигнала в *сигнальной памяти*. При этом логический модуль осуществляет управление процессом записи, воздействуя на *блок управления* памяти и указывая очередную свободную ячейку. Для этого данный модуль должен отслеживать количество записанных в памяти конфликтных сигналов M_t в слоте t .

В случае удачного декодирования принятого из канала связи (успешного) сигнала декодер также сообщает о результате логическому модулю и предпринимает попытку декодирования разности принятого сигнала и каждого из сигналов, хранящихся в памяти, поочередно. Результаты декодирования каждый раз сообщаются логическому модулю, который по-прежнему осуществляет управление процессом доступа к памяти, а полученные после вычитания «новые» конфликтные сигналы подменяют собой «старые».

Каждый успешно декодированный при работе описываемой процедуры погашения интерференции сигнал приводит к новой итерации последовательного декодирования. При этом на вход декодера теперь поступают разности данного сигнала и

всех остальных сигналов, хранящихся в памяти. В ходе такого итеративного декодирования все успешно восстановленные пакеты данных передаются на вышележащие уровни по восходящему тракту данных, а соответствующие им хранимые сигналы удаляются из памяти. Логический модуль физического уровня обрабатывает результаты каждой попытки декодирования и передает по ее завершении обратную связь некоторого заранее установленного типа логическому модулю подуровня УДС.

СПИСОК ЛИТЕРАТУРЫ

1. *Bertsekas D.* Сети передачи данных. М.: Мир, 1989.
2. *Rom R., Sidi M.* Multiple access protocols: Performance and analysis. N.Y.: Springer-Verlag, 1990.
3. *Abramson N.* The Aloha system – another alternative for computer communications // Proc. AFIPS Conf. 1970. V. 36. P. 295–298.
4. *Цыбаков Б.С., Михайлов В.А.* Свободный синхронный доступ пакетов в широкополосный канал с обратной связью // Пробл. передачи информ. 1978. Т. 14. № 4. С. 32–59.
5. *Capetanakis J.I.* Tree algorithms for packet broadcast channels // IEEE Transactions Inform. Theory. 1979. V. 25. № 4. P. 505–515.
6. *Pedersen K.I., Kolding T.E., Seskar I., Holtzman J.M.* Practical implementation of successive interference cancellation in DS/CDMA systems // Proc. Int. Conf. Universal Personal Commun. 1996. P. 321–325.
7. *Andrews J.* Analysis of cancellation error for successive interference cancellation with imperfect channel estimation // Multiuser Wireless Commun. 2002. P. 1–17.
8. *Agrawal A., Andrews J.G., Cioffi J.M., Meng T.* Iterative power control for imperfect successive interference cancellation // IEEE Transactions Wireless Commun. 2005. V. 4. P. 878–884.
9. *Weber S., Andrews J.G., Yang X., Veciana G.D.* Transmission capacity of wireless ad hoc networks with successive interference cancellation // IEEE Transactions Inform. Theory. 2007. V. 53. P. 2799–2814.
10. IEEE Std 802.16e-2005, Piscataway, NJ, USA, December 2005.
11. *Yu Y.* SICTA: A 0.693 contention tree algorithm using successive interference cancellation // Proc. IEEE Conf. Comput. Commun. 2005. V. 3. P. 1908–1916.
12. *Yu Y., Giannakis G.B.* High-throughput random access using successive interference cancellation in a tree algorithm // IEEE Transactions Inform. Theory. 2007. V. 53. № 12. P. 4628–4639.
13. *Wang X., Yu Y., Giannakis G.B.* Combining random backoff with a cross-layer tree algorithm for random access in IEEE 802.16 // Proc. IEEE Wireless Commun. Networking Conf. 2006. V. 2. P. 972–977.
14. *Wang X., Yu Y., Giannakis G.B.* A deadlock-free high-throughput tree algorithm for random access over fading channels // Proc. Conf. Inform. Sci. Syst. 2006. V. 22. P. 420–425.
15. *Peeters G.T., Houdt B.V., Blondia C.* A multiaccess tree algorithm with free access, interference cancellation and single signal memory requirements // Performance Evaluat. 2007. V. 64. № 9–12. P. 1041–1052.
16. *Houdt B.V., Peeters G.T.* FCFS tree algorithms with interference cancellation and single signal memory requirements // Proc. Int. Workshop on Multiple Access Commun. 2008. V. 1. P. 1–6.
17. *Евсеев Г.С., Тюрликов А.М.* Взаимосвязь характеристик блокированных стек-алгоритмов случайного множественного доступа // Пробл. передачи информ. 2007. Т. 43. № 4. С. 83–92.
18. *Massey J.L.* Collision resolution algorithm and random access communications / Multiuser Commun. Syst. G. Longo: CISM Course and Lecture Notes. 1981. P. 73–131.
19. *Цыбаков Б.С., Михайлов В.А.* Случайный множественный доступ пакетов. Алгоритм дробления // Пробл. передачи информ. 1980. Т. 16. № 4. С. 65–79.

20. *Gyorfi L., Gyori S., Massey J.L.* Principles of stability analysis for random accessing with feedback // Proc. NATO Security Through Sci. Ser. Inform. Commun. Security. 2007. V. 10. P. 214–250. .
21. *Szpankowski W.* Average case analysis of algorithms on sequences. N.Y.: Wiley, 2001.
22. *Михайлов В.А.* Об одном рекуррентном уравнении в теории случайного множественного доступа // IX Симпоз. по проблеме избыточности в информ. системах. 1986. V. 2. P. 148–150.
23. *Gyorfi L., Gyori S.* Analysis of tree algorithm for collision resolution // Proc. Int. Conf. Anal. Algorithms. 2005. P. 357–364.
24. *Andreev S., Pustovalov E., Turlikov A.* SICTA modifications with single memory location and resistant to cancellation errors // Proc. Int. Conf. Next Generation Teletraffic and Wired/Wireless Advanced Networking. 2008. P. 13–24.
25. *Andreev S., Pustovalov E., Turlikov A.* Tree algorithms with free access and interference cancellation in presence of cancellation errors // Proc. Int. Symp. Wireless Personal Multimedia Commun. 2008. V. 1. P. 1–5.

Статья представлена к публикации членом редколлегии В.М. Вишневым.

Поступила в редакцию 07.08.2008