

УДК 004.728.3.057.4

# РАЗРЕШЕНИЕ КОНФЛИКТОВ В СИСТЕМАХ РАДИОЧАСТОТНОЙ ИДЕНТИФИКАЦИИ С ИСПОЛЬЗОВАНИЕМ ИДЕНТИФИКАТОРОВ МЕТОК И ПРОЦЕДУРЫ ПОСЛЕДОВАТЕЛЬНОЙ КОМПЕНСАЦИИ КОНФЛИКТНЫХ СИГНАЛОВ

**С. Г. Марковский,**

канд. техн. наук, доцент

**Н. В. Марковская,**

канд. техн. наук, доцент

Санкт-Петербургский государственный университет аэрокосмического приборостроения

Предлагается модификация алгоритма разрешения конфликтов, описанного в стандарте ISO/IEC 18000-6. В отличие от стандартного алгоритма, для разрешения конфликтов используются идентификаторы меток и процедура последовательной компенсации конфликтных сигналов. Показывается, что алгоритм может быть представлен в терминах стек-алгоритма, устойчивого к ошибкам восстановления процедуры компенсации конфликтных сигналов. Исследуется влияние ошибок процедуры последовательной компенсации конфликтных сигналов на работу алгоритма разрешения конфликта.

**Ключевые слова** — системы радиочастотной идентификации, идентификаторы меток, процедура последовательной компенсации конфликтных сигналов, алгоритм разрешения конфликтов.

## Введение

В настоящее время системы радиочастотной идентификации (Radio Frequency Identification Devices — RFID) находят широкое применение в различных областях [1]. В общем случае в системе имеется некоторое центральное устройство (считыватель) и множество меток. Считыватель должен уметь определять, какие метки находятся в его зоне действия. Для решения этой задачи считыватель по радиоканалу запрашивает метки. Все метки, которые успешно приняли этот запрос, посылают ответ, содержащий идентификатор метки. В системе должны быть приняты меры, которые позволяют либо разнести по времени сигналы от разных меток, либо обеспечить успешный прием сигналов даже при их наложении. В настоящей работе рассматриваются только способы, обеспечивающие разнесение сигналов по времени с помощью некоторого алгоритма разрешения конфликта. Такие способы описываются, в частности, в стандарте ISO/IEC 18000-6 [2], где упоминаются два варианта алгоритма

разрешения конфликта. Первый вариант основан на известном алгоритме ALOHA, второй — на древовидном алгоритме разрешения конфликтов, впервые предложенном в работах [3, 4]. В данной статье рассматривается второй вариант, который имеет преимущество перед первым [3–6].

Упрощенно работу системы идентификации, функционирующую согласно стандарту ISO/IEC 18000-6 при использовании древовидных алгоритмов разрешения конфликта, можно описать следующим образом. Все время функционирования системы разделено на непересекающиеся интервалы — сеансы. В начале очередного сеанса считыватель посылает *начальный запрос*, на который отвечают все метки, которые находятся в радиусе действия считывателя. Если нет меток, которые отвечают на начальный запрос, или есть только одна такая метка, то сеанс заканчивается. В противном случае происходит конфликт, т. е. наложение сигналов от разных меток. Во время сеанса конфликт разрешается с помощью древовидного алгоритма разрешения конфликта. В течение сеанса счи-

тыватель передает повторные запросы, на которые могут отвечать только те метки, которые отвечали на начальный запрос. Остальные метки не могут передавать до окончания сеанса даже в том случае, если окажутся в зоне действия считывателя. Одной из характеристик системы является *задержка идентификации* — интервал времени от момента появления метки в зоне действия считывателя до того момента, когда считыватель идентифицирует метку, т. е. успешно примет сигнал от метки. В статье предлагается для уменьшения задержки идентификации видоизменить алгоритм разрешения конфликтов, который описан в стандарте ISO/IEC 18000-6. Совместно рассматриваются следующие два направления: использование при разрешении конфликта уникальных идентификаторов меток и специальная обработка сигналов на считывателе, называемая *процедурой последовательного погашения конфликтных сигналов*. Следует отметить, что ранее, по отдельности, оба эти направления улучшения алгоритма разрешения конфликта уже рассматривались. Идея использовать адреса абонентских устройств обсуждалась в самых первых работах, посвященных древовидным алгоритмам. Различные варианты реализации этой идеи описаны и проанализированы в работах [5, 6]. Использование процедуры последовательного погашения конфликтных сигналов в древовидных алгоритмах впервые рассмотрено в публикации [7]. В этой работе был описан алгоритм, реализация которого на практике связана с существенными техническими проблемами. В дальнейшем были предложены алгоритмы, которые допускают более простую реализацию [8–10].

### Описание модели системы

При описании модели системы будем использовать устоявшуюся терминологию для работ, в которых исследуются алгоритмы разрешения конфликтов в системах со случайным множественным доступом (СМД). Интервал времени, который начинается в момент передачи запроса и заканчивается моментом окончания передачи ответа на данный запрос, будем называть *окном*. При этом мы абстрагируемся от детального рассмотрения последовательности событий, которые происходят в окне. Для нас важным является лишь тот факт, что каждая находящаяся в зоне действия считывателя метка к началу окна получает от считывателя некоторую информацию о предыдущем окне, и на основе этой информации метка принимает решение передавать или не передавать в этом окне. Длительность окна зависит от особенностей реализа-

ции системы идентификации и, согласно данным, приведенным в стандарте ISO/IEC 18000-6, составляет порядка 4 мс. Далее будет введена модель системы, с помощью которой будут исследованы такие характеристики системы радиочастотной идентификации, как скорость идентификации и средняя задержка идентификации. Эти характеристики являются относительными. За единицу времени принята длительность окна. При этом под *скоростью идентификации* понимается некоторая условная величина, которая показывает среднее число меток, идентифицируемых в единицу времени при условии, что все метки постоянно находятся в зоне действия считывателя. В реальных системах перемещение меток зависит от назначения системы идентификации. Разработка математической модели, описывающей такое перемещение, является отдельной задачей. В большинстве работ, в которых исследуются алгоритмы разрешения конфликтов в системе идентификации [11, 12], процесс поступления меток в зону действия считывателя описывается пуассоновским потоком. В настоящей работе мы будем рассматривать следующую модель поступления меток. Если метка находится вне зоны действия считывателя, то к началу каждого очередного окна метка с некоторой вероятностью  $p$  перемещается в зону действия считывателя и с вероятностью  $(1 - p)$  остается вне зоны. Если в некотором окне считыватель успешно принимает сигнал от метки, то считается, что произошла идентификация метки, и метка удаляется из зоны действия считывателя. Все последующее рассмотрение делается в предположении, что метка не может быть удалена из зоны действия считывателя до идентификации. *Задержка идентификации* — это случайная величина, равная числу окон, в течение которых метка находится в зоне действия считывателя. Математическое ожидание данной случайной величины — *средняя задержка идентификации*.

Далее в работе метки называются *абонентами*, а считыватель — *базовой станцией* (БС). С учетом вышесказанного систему можно рассматривать как систему передачи сообщений от абонентов к БС.

Данная система будет состоять из  $M$  абонентов и БС. Абоненты передают сообщения на БС по восходящему каналу связи. Ответную информацию о ситуации в восходящем канале абоненты получают от БС по нисходящему каналу связи. Каждому абоненту присваивается адрес, представленный в двоичной системе счисления. Диапазон возможных адресов изменяется от 0 до  $M - 1$ . Будем полагать  $M = 2^l$ , где  $l$  — разрядность адреса.

Сформулируем основные допущения модели, описывающей функционирование системы СМД. Совокупность данных допущений представлена в большинстве работ, посвященных СМД, и образует *классическую модель* СМД. Как правило, в этих работах классическая модель модифицируется или изменением отдельных допущений, или введением новых допущений.

*Допущение 1. Синхронная система.* Все время работы системы подразделяется на окна. Моменты разделения окон известны всем абонентам системы. Абонент может начинать передачу только в начале окна.

*Допущение 2. Одинаковая длина сообщений.* Все передаваемые сообщения имеют равную длину, принимаемую за единицу времени и совпадающую с длиной окна. Для таких сообщений будем использовать термин *пакет*. Окну с номером  $t$  (далее — просто окно  $t$ ) будет соответствовать интервал времени  $[t - 1, t)$ .

*Допущение 3. Троичная обратная связь.* В каждом окне абонент может различать три ситуации в канале:

- нет передачи пакетов («ПУСТО»);
- передает один абонент («УСПЕХ»);
- два и более пакета передаются одновременно («КОНФЛИКТ»).

Следуя работам Цыбакова [3, 5], для трех вышеописанных ситуаций будем использовать обозначения П, У и К.

*Допущение 4. Мгновенная обратная связь.* Все абоненты узнают о состоянии канала в конце текущего окна до того, как необходимо принять решения о действиях в следующем окне.

*Допущение 5. Безошибочная обратная связь.* Канал связи свободен от ошибок. Все абоненты получают правильную информацию о состоянии в канале.

*Допущение 6. Вернулливский входной поток.* Будем полагать, что в одном окне у каждого абонента может появиться не более одного пакета. Вероятность появления пакета будем обозначать через  $p$ . Тогда

$$E \left[ \sum_{i=0}^{M-1} X_i(t) \right] = pM = \lambda,$$

где  $X_i(t)$  — случайная величина, равная числу пакетов, появившихся в окне с номером  $t$  у абонента с номером  $i$ ;  $\lambda$  — среднее число пакетов, возникающих у всех абонентов в одном окне, и  $p = E[X_i(t)]$  для всех  $t \geq 0$  и  $i = 0, \dots, M - 1$ .

Применительно к данной системе допущений далее сформулируем алгоритм, для которого будет приведено выражение для скорости алгоритма и получено значение средней задержки алгоритма.

## Основные определения

Число пакетов, передаваемых в канал в окне  $t$ , образует *кратность конфликта* в окне  $t$ . Пустому окну соответствует *конфликт кратности 0*. Если в окне передает только один абонент, то это *конфликт кратности 1*.

Пусть в окне  $t$  возникает первоначальный конфликт кратности  $k$ . Правило, в соответствии с которым абоненты, участвующие в конфликте, принимают решение о передаче пакетов, называется *алгоритмом разрешения конфликта* (АРК).

*Сеанс* — интервал времени, соответствующий длительности АРК в окнах. Кратность сеанса  $k$  — число пакетов, передаваемых в первом окне сеанса. Если  $k = 0$  или  $k = 1$ , то сеанс длится одно окно.

Каждому сеансу можно поставить в соответствие двоичное *дерево разрешения конфликта* (ДРК). Вершины дерева соответствуют окнам АРК. Корень дерева образует вершина, в которой наблюдается первоначальный конфликт. Концевые вершины дерева — окна с ситуациями {П} или {У}. При  $k = 0$  или  $k = 1$  корневая вершина дерева совпадает с концевой. ДРК, в котором в первоначальном конфликте все абоненты системы передают пакеты, образует *ДРК максимальной кратности*.

Правило, в соответствии с которым пакеты, поступающие в систему, могут присоединиться к АРК, называется *алгоритмом доступа к каналу* (АДК). *Блокированный АДК* предполагает, что пакеты, поступающие в систему в течение текущего сеанса, могут быть переданы в канал только в первом окне следующего сеанса. В данной статье будет использоваться именно блокированный АДК.

## Древовидные алгоритмы разрешения конфликта

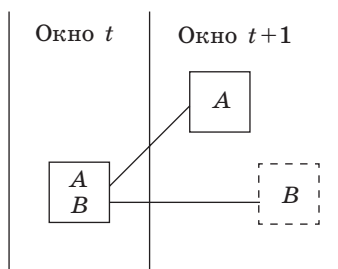
Данный класс алгоритмов показывает высокую эффективность при разрешении конфликтов и обеспечивает конечную задержку передачи сообщения в системе при сколь угодно большом числе абонентов. Древовидные алгоритмы были детально изучены как для схемы разбиения, использующей механизм подбрасывания монеты для разрешения конфликтов (*случайные паспорта*) [3, 4], так и для схемы, использующей адреса абонентов (*фиксированные паспорта*) [3, 4]. Эти алгоритмы при разрешении конфликтов не используют дополнительную информацию физического уровня.

Развитие методов цифровой обработки сигналов и методов помехоустойчивого кодирования привело к возможности использовать процедуру *последовательной компенсации конфликтных сиг-*

налов (ПККС) на физическом уровне [7]. ПККС можно рассматривать как декодирование с использованием некоторой дополнительной информации. Рассмотрим пример [7], иллюстрирующий, как ПККС позволяет повысить эффективность работы древовидных алгоритмов (рис. 1).

Пусть в окне  $t$  абоненты  $A$  и  $B$  одновременно передают пакеты, что приводит к их наложению. Обозначим через  $y_t$  сигнал, принятый БС к концу окна  $t$ , а через  $x_A$  и  $x_B$  — сигналы, соответствующие пакетам данных. Сигнал  $y_t$  формируется в результате наложения на входе приемника сигналов  $x_A$  и  $x_B$  и шумов. Далее при рассмотрении этого примера будем считать, что шумы в канале отсутствуют. Тогда можно записать:  $y_t = x_A + x_B$ , где знак «+» означает операцию смешивания сигналов. После обработки сигнала  $y_t$  БС выносит решение о том, что в окне  $t$  произошел конфликт. Сигнал  $y_t$  сохраняется БС в сигнальной памяти. Получив сигнал  $y_{t+1} = x_A$  в конце окна  $t + 1$ , БС успешно выделяет сигнал  $x_A$ . По выделенному сигналу восстанавливается пакет абонента  $A$ . Далее БС выполняет процедуру ПККС и нейтрализует выделенный сигнал  $x_A$  из сохраненной смеси сигналов  $y_t$ . Можно записать  $x_B = y_t - x_A$ , где знак «-» означает операцию нейтрализации сигнала из смеси сигналов. В результате этой операции успешно восстанавливается сигнал  $x_B$ . Таким образом, дальнейшее разрешение конфликта не требуется. В рассмотренном примере длительность сеанса сокращается на одно окно по сравнению с традиционными древовидными алгоритмами.

Алгоритм, использующий ПККС, впервые был предложен в работе [7]. Авторы назвали его SICTA (Successive Interference Cancellation in a Tree Algorithm). При этом алгоритм был рассмотрен только для системы с бесконечным числом абонентов и неограниченной сигнальной памятью, используемой для хранения конфликтных сигналов. Наличие у БС потенциально неограниченного объема памяти для хранения конфликтных сигналов практически нереализуемо. В работах [8, 9] была рассмотрена модель алгоритма SICTA с единичной памятью, которая позволяет на прак-



■ Рис. 1. Использование процедуры ПККС



■ Рис. 2. Устройство базовой станции

тике существенно упростить процедуру ПККС. Введем новое допущение в модель системы относительно объема сигнальной памяти БС.

*Допущение 7. БС с единичной сигнальной памятью.* На физическом уровне БС имеется возможность сохранять только один принятый из канала связи сигнал.

Устройство БС представлено на рис. 2. БС имеет единичную сигнальную память и блоки, реализующие операцию определения ситуации в канале и процедуру ПККС.

Предлагаемый в данной работе алгоритм для системы с конечным числом абонентов и единичной сигнальной памятью будем рассматривать как комбинацию двух алгоритмов: алгоритма работы БС и алгоритма работы абонента.

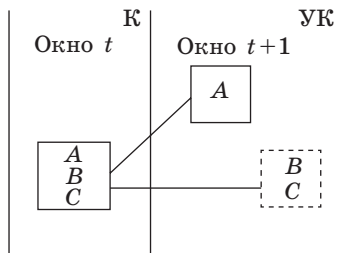
### Алгоритм работы базовой станции

Базовая станция определяет ситуацию в канале. Согласно допущению 3 классической модели, в восходящем канале связи могут быть три различные ситуации {П, У, К}. Если ситуация в канале {У} и в последнем непустом окне был конфликт, БС выполняет процедуру ПККС, используя единичную сигнальную память. После данной процедуры по обратному нисходящему каналу связи абонентам передается расширенная информация о ситуации в канале.

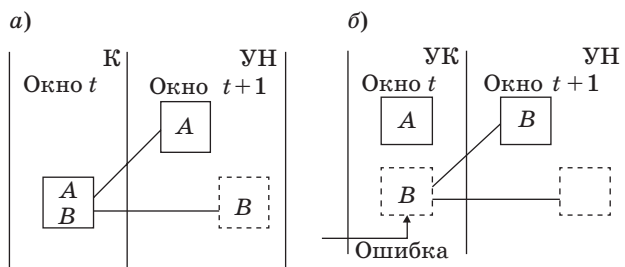
*Допущение 8. Расширенная обратная связь.* Объем обратной связи, доступной на подуровне управления доступом к среде (УДС) абонента, расширяется с учетом возможностей процедуры ПККС.

Расширенная обратная связь определяет 5 возможных ситуаций в канале:

- 1) П — «ПУСТО» (отсутствие передачи);
- 2) У — «УСПЕХ» (передает один абонент);
- 3) УН — «УСПЕХ и НЕ КОНФЛИКТ» (передает один абонент, а процедура ПККС позволяет восстановить оставшийся сигнал как пакет второго абонента или отсутствие передачи);



■ Рис. 3. Ситуация {УК} после процедуры ПККС



■ Рис. 4. Ситуация {УН} после процедуры ПККС

4) УК — «УСПЕХ и КОНФЛИКТ» (передает один абонент, а процедура ПККС позволяет восстановить оставшийся сигнал как конфликт); ситуация имеет место, когда в предыдущем окне был конфликт кратности 3 и более (рис. 3);

5) К — «КОНФЛИКТ» (одновременно передают два и более абонента).

Ситуация УН соответствует двум случаям:

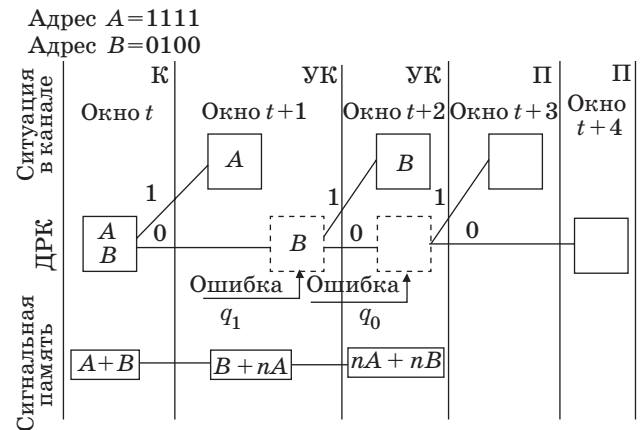
- процедура ПККС успешно восстанавливает оставшийся сигнал как пакет данных (в предыдущем окне был конфликт кратности 2, рис. 4, а);
- процедура ПККС восстанавливает оставшийся сигнал как отсутствие передачи (в предыдущем окне был ложный конфликт кратности 1 при возникновении ошибки в процедуре ПККС, рис. 4, б). Ошибки процедуры ПККС подробнее будут рассмотрены в следующем разделе.

Сигнал восходящего канала сохраняется в сигнальной памяти только при ситуациях {К} и {УК}.

### Возможные ошибки процедуры ПККС

В реальных приемных устройствах с процедурой ПККС возможно появление ошибок, обусловленных наличием «остаточных сигналов» после нейтрализации принятого сигнала в исходном составном сигнале. Для примера рассмотрим ДРК между абонентами А и В (рис. 5).

После окна  $t$ , в котором имела место ситуация {К}, в сигнальной памяти сохраняется смесь сигналов А и В. В окне  $t + 1$  БС успешно принимает сигнал А и выполняет процедуру ПККС, пытаясь восстановить сигнал В, используя сигнальную



■ Рис. 5. Пример ДРК с ошибками восстановления в процедуре ПККС

память и нейтрализуя сигнал А. При этом может произойти ошибка в процедуре ПККС ввиду наличия остаточного сигнала  $n_A$  при нейтрализации сигнала А. В этом случае декодер не может правильно декодировать сигнал В и делает вывод о том, что пакет с данными искажен. БС определяет ситуацию как {УК}, и в сигнальной памяти сохраняется смесь сигналов В и  $n_A$ . Следовательно, возникает ложный конфликт кратности 1, который далее надо разрешать. Вышеописанная ошибка называется *ошибкой восстановления сигнала пакета (полезного сигнала) после нейтрализации успешно принятого сигнала*. Вероятность возникновения такой ошибки будем обозначать  $q_1$ . Согласно АРК, в окне  $t + 2$  повторно передается пакет В. БС успешно принимает сигнал В и выполняет процедуру ПККС. При этом возможна другая ошибка ПККС, когда в результате восстановления сигнала  $n_A$  и наличия остаточного сигнала  $n_B$  БС воспринимает ситуацию как ложный конфликт кратности 0, который разрешается в последующих окнах. БС определяет ситуацию как {УК}, и в сигнальной памяти сохраняется смесь сигналов  $n_A$  и  $n_B$ . Такую ошибку будем называть *ошибкой восстановления шумового сигнала после нейтрализации успешно принятого сигнала*, а через  $q_0$  обозначать вероятность возникновения данной ошибки. Разрешение ложного конфликта потребует двух дополнительных пустых окон  $t + 3$  и  $t + 4$ . Если ошибка в окне  $t + 2$  не происходит (вероятность события  $1 - q_0$ ), БС выделяет остаточный сигнал  $n_A$ , который не связан с полезным сигналом и формирует ситуацию {УН}. В этом случае сеанс завершается в окне  $t + 2$ .

Все вышесказанное сформулируем в следующем допущении.

*Допущение 9. Неточная обратная связь.* В силу ошибок восстановления информация обратной

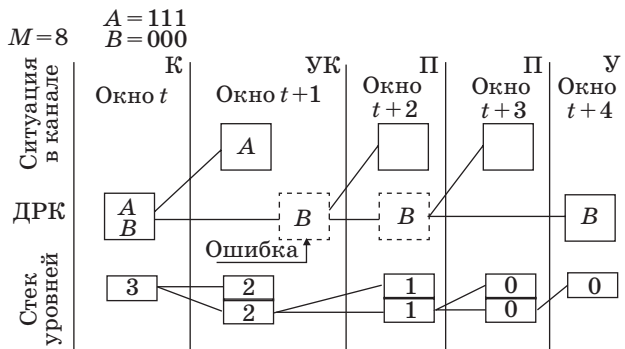


Рис. 6. Пример выхода алгоритма из заикливания

связи, поступающая на подуровень УДС абонента, не точна. Значения вероятностей ошибок восстановления после нейтрализации успешно принятого сигнала  $q_0$  и  $q_1$  одинаковы для всех окон, в которых выполняется процедура ПКС.

В работе [10] показано, что при возникновении ложного конфликта кратности 1 алгоритм SICTA со случайными паспортами заикливается. Авторы [10] разработали алгоритм SICTA с компенсацией при успехе и конфликте, устойчивый к ошибкам восстановления. При этом для устранения эффекта заикливания пришлось отказаться от пропуска некоторых конфликтных окон, что привело к снижению скорости алгоритма SICTA. Рассматриваемый в работе алгоритм для схемы с фиксированными паспортами свободен от проявления эффекта заикливания, так как ДРК является конечным и алгоритм выходит из заикливания (рис. 6) при достижении конечных вершин ДРК максимальной кратности. Для обнаружения этого события служит стек уровней вершин, в котором нулевое содержимое верхней ячейки говорит о том, что вершина ДРК является концевой. Правила работы стека уровней вершин будут рассмотрены далее при описании алгоритма работы абонента.

### Устройство абонента

Устройство абонента представлено на рис. 7. Каждый абонент имеет свой уникальный адрес, предназначенный для разрешения конфликтов. Для хранения пакета, поступающего в течение сеанса, используется буфер, состоящий из одной ячейки. Если за время сеанса к абоненту поступает большее число пакетов, то они не обрабатываются абонентом и считаются потерянными. В момент начала нового сеанса пакет из буфера переписывается в верхнюю ячейку стека пакетов. Подобная модель буфера на две ячейки детально описана [6]. В настоящей работе роль второй ячейки выполняет верхняя ячейка стека пакетов.

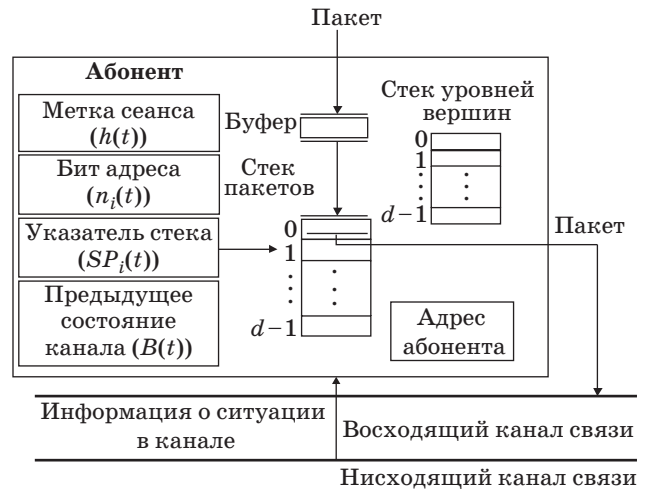


Рис. 7. Устройство абонента

*Стек пакетов* — запоминающее устройство, состоящее из  $d = \log_2 M + 1$  ячеек. В процессе разрешения конфликта абонент перемещает пакет по стеку в соответствии с инструкциями алгоритма, т. е. всегда занята только одна ячейка стека, на которую показывает указатель стека  $SP_i(t)$ . Если  $SP_i(t) = 0$ , то абонент передает пакет в канал.

Переменная  $h(t)$  служит для определения границ сеансов. Если  $h(t) = 0$ , то текущий сеанс завершен и начинается новый сеанс. Переменная  $B(t)$  хранит ситуацию, которая наблюдалась в канале в последнем непустом окне и может принимать значения {К} или {У}. Бит адреса  $n_i(t)$  определяет номер текущего бита адреса  $a_i(n_i(t))$ , используемого для разрешения конфликта. И, наконец, абонент имеет *стек уровней вершин дерева*, который имеет глубину  $\log_2 M + 1$  и строится на основе ДРК. Он служит для определения уровней вершин ДРК. В ячейках стека уровней вершин хранятся номера уровней, соответствующие отдельным вершинам ДРК. Подобный стек используется в работе [13] для определения концевых вершин ДРК при наличии ложных конфликтов в канале под названием «терминальный стек». Отличие стека уровней вершин от терминального стека заключается в содержании ячеек стека. Если терминальный стек хранит в ячейке число абонентов, способных передавать в определенной вершине дерева, то стек уровней вершин — номер уровня вершины. Кроме того, используемая в инструкциях для терминального стека операция деления на 2 заменена в стеке уровней вершин операцией вычитания.

Следует отметить, что переменные  $SP_i(t)$  и  $n_i(t)$  определяют движение пакета абонента по дереву (стеку пакетов) и являются индивидуальными для каждого абонента, в то время как переменные  $h(t)$  и  $B(t)$  являются глобальными и вычисля-

ются всеми абонентами одинаково. Стек уровней вершин — глобальный стек (заполняется одинаково всеми абонентами).

**Алгоритм работы абонента**

В начале первого окна сеанса для алгоритма задаются начальные значения переменных. Если в буфере абонента есть пакет, то он перемещается в верхнюю ячейку стека и  $SP_i(t) = 0$ , в противном случае  $SP_i(t) = -1$ . Кроме того, задаются следующие значения переменных:  $n_i(t) = l$ ,  $h(t) = 1$ ,  $B(t) = \{Y\}$ ,  $TS(0) = l$ ,  $TS(1) = \dots = TS(d - 1) = -1$ , где  $TS(i)$  — содержимое ячейки стека уровней вершин с номером  $i$ . В верхнюю ячейку стека уровней вершин заносится номер уровня, соответствующий корневой вершине ( $TS(0) = l$ ).

Обозначим через  $TS(i)(t)$  — содержимое ячейки стека уровней вершин с номером  $i$  в окне  $t$ . Если  $TS(i)(t) = 0$ , то это говорит о том, что вершина дерева является концевой в ДПК максимальной кратности и абонент использовал все биты адреса для разрешения конфликта.

Приведем инструкции алгоритма.

**Инструкции работы для стека уровней вершин.**

1. Если  $\eta(t) = \{K\}$ , то  $TS(0)(t + 1) = TS(1)(t + 1) = TS(0)(t) - 1$ ;  $TS(2)(t + 1) = TS(1)(t)$ ;  $\dots$   $TS(d - 1)(t + 1) = TS(d - 2)(t)$ .
2. Если  $\eta(t) = \{Y\}$ , или  $\eta(t) = \{\Pi\}$  и  $B(t) = \{Y\}$ , или  $\eta(t) = \{YK\}$  и  $TS(0)(t) = 0$  (концевая вершина ДПК максимальной кратности), то  $TS(0)(t + 1) = TS(1)(t)$ ;  $TS(1)(t + 1) = TS(2)(t)$ ;  $\dots$   $TS(d - 2)(t + 1) = TS(d - 1)(t)$ ;  $TS(d - 1)(t + 1) = -1$ .
3. Если  $\eta(t) = \{\Pi\}$  и  $B(t) = \{K\}$  или  $\eta(t) = \{YK\}$  и  $TS(0)(t) \neq 0$ , то  $TS(0)(t + 1) = TS(1)(t + 1) = TS(1)(t) - 1$ .
4. Если  $\eta(t) = \{YH\}$ , то  $TS(0)(t + 1) = TS(2)(t)$ ;  $TS(1)(t + 1) = TS(3)(t)$ ;  $\dots$   $TS(d - 3)(t + 1) = TS(d - 1)(t)$ ;  $TS(d - 2)(t + 1) = TS(d - 1)(t + 1) = -1$ .

**Инструкции для  $SP_i(t)$ .**

1. Если пакет передавался в окне, т. е.  $SP_i(t) = 0$ , то:
  - если  $\eta(t) = \{Y\}$ , или  $\eta(t) = \{YH\}$ , или  $\eta(t) = \{YK\}$ , то пакет покидает стек и абонент прекращает вычисление величины  $SP_i(t)$  ( $SP_i(t) = -1$ );
  - если  $\eta(t) = \{K\}$ , то  $SP_i(t + 1) = 1 - a_i(n_i(t))$ .
2. Если пакет не передавался в окне и  $SP_i(t) = 1$ , то:

- если  $\eta(t) = \{K\}$ , то  $SP_i(t + 1) = 2$ ;
- если  $\eta(t) = \{Y\}$  или  $\eta(t) = \{YK\}$  и  $TS(0)(t) = 0$  (ошибка), то  $SP_i(t + 1) = 0$ ;
- если  $\eta(t) = \{YH\}$ , то пакет покидает стек и абонент прекращает вычисление величины  $SP_i(t)$  ( $SP_i(t) = -1$ );
- если  $\eta(t) = \{\Pi\}$  и  $B(t) = \{Y\}$ , то  $SP_i(t + 1) = 0$ ;
- если  $\eta(t) = \{\Pi\}$  и  $B(t) = \{K\}$  или  $\eta(t) = \{YK\}$  и  $TS(0)(t) \neq 0$ , то  $SP_i(t + 1) = 1 - a_i(n_i(t))$ .

3. Если пакет не передавался в окне и  $SP_i(t) \geq 2$ , то:

- если  $\eta(t) = \{K\}$ , то  $SP_i(t + 1) = SP_i(t) + 1$ ;
- если  $\eta(t) = \{Y\}$  или  $\eta(t) = \{\Pi\}$  и  $B(t) = \{Y\}$ , то  $SP_i(t + 1) = SP_i(t) - 1$ ;
- если  $\eta(t) = \{YH\}$ , то  $SP_i(t + 1) = SP_i(t) - 2$ ;
- если  $\eta(t) = \{\Pi\}$  и  $B(t) = \{K\}$  или  $\eta(t) = \{YK\}$ , то  $SP_i(t + 1) = SP_i(t)$ .

**Инструкции для  $n_i(t)$ .**

1. Если пакет передавался в окне, т. е.  $SP_i(t) = 0$  и  $\eta(t) = \{K\}$ , или если  $SP_i(t) = 1$ ,  $\eta(t) = \{\Pi\}$  и  $B(t) = \{K\}$ , или если  $SP_i(t) = 1$ ,  $\eta(t) = \{YK\}$  и  $TS(0)(t) \neq 0$ , то  $n_i(t + 1) = n_i(t) - 1$ .

2. Во всех остальных случаях  $n_i(t + 1) = n_i(t)$ .

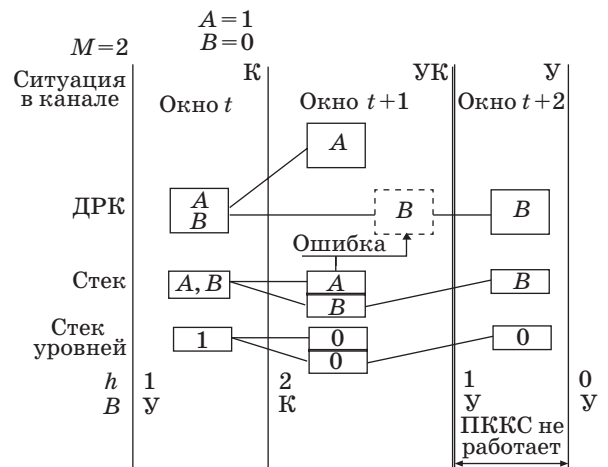
**Инструкции для  $h(t)$ .**

1. Если  $h(t) = 0$ , то сеанс завершен.
2. Если  $h(t) \geq 1$ , то:
  - если  $\eta(t) = \{K\}$ , то  $h(t + 1) = h(t) + 1$ ;
  - если  $\eta(t) = \{Y\}$ , или  $\eta(t) = \{\Pi\}$  и  $B(t) = \{Y\}$ , или  $\eta(t) = \{YK\}$  и  $TS(0)(t) = 0$ , то  $h(t + 1) = h(t) - 1$ ;
  - если  $\eta(t) = \{YH\}$ , то  $h(t + 1) = h(t) - 2$ ;
  - если  $\eta(t) = \{\Pi\}$  и  $B(t) = K$  или  $\eta(t) = \{YK\}$  и  $TS(0)(t) \neq 0$ , то  $h(t + 1) = h(t)$ .

**Инструкции для  $B(t)$ .**

- если  $\eta(t) = \{Y\}$  или  $\eta(t) = \{YH\}$ , то  $B(t + 1) = \{Y\}$ ;
- если  $\eta(t) = \{K\}$  или  $\eta(t) = \{YK\}$  и  $TS(0)(t) \neq 0$ , то  $B(t + 1) = \{K\}$ ;
- если  $TS(0)(t) = 0$ , то  $B(t + 1) = \{Y\}$ .

Пример ДПК двух абонентов, иллюстрирующий алгоритм работы абонента, представлен на



■ Рис. 8. Пример алгоритма работы абонента

рис. 8. В окне  $t + 1$  возникает ложный конфликт кратности 1, и БС формирует ситуацию УК. По содержимому верхней ячейки стека уровней вершин абонент определяет, что окно  $t + 1$  соответствует вершине уровня 0, т. е. концевой вершине, и изменяет значение переменной  $B(t + 1)$  на {У}. Поэтому в дополнительном окне  $t + 2$  процедура ПККС не выполняется, и пакет абонента  $B$  успешно принимается БС. Подобный алгоритм разрешения конфликта, когда ложный конфликт кратности 0 или 1 возникал в вершине ДРК уровня 0, был детально исследован [13].

**Скорость алгоритма**

Для модели абонента с двумя ячейками понятие скорости алгоритма  $R$  было введено Капетанакисом [4] как отношение числа пакетов, переданных в сеансе максимальной кратности  $M$ , к длительности сеанса  $L_{\max}$ :

$$R = \frac{M}{L_{\max}} = \frac{M}{2M - 1}, \tag{1}$$

где  $2M - 1$  — длина сеанса кратности  $M$ . Для предложенного в статье алгоритма, учитывающего ошибки процедуры ПККС, длительность сеанса максимальной кратности можно определить по формуле

$$L_{\max} = 2M - 1 - \frac{1}{2}M(1 - q_1) + \frac{1}{2}Mq_1 = \frac{3M}{2} + Mq_1 - 1. \tag{2}$$

Подставив (2) в (1), получим

$$R = \frac{1}{-\frac{1}{M} + q_1 + \frac{3}{2}}. \tag{3}$$

При  $M \rightarrow \infty$   $R \rightarrow \frac{2}{3 + 2q_1}$ .

По выражению (3) можно рассчитать значение параметра скорости идентификации, определение которого было дано при описании модели системы. Следует отметить, что скорость идентификации зависит только от вероятности ошибки  $q_1$ .

**Средняя задержка алгоритма**

Если для расчета скорости предложенного в статье алгоритма можно воспользоваться методикой, описанной в работах [6, 13], то для расчета средней задержки ее применять нельзя, что связано с особенностями использования процедуры ПККС в алгоритме.

Стандарт ISO/IEC 18000-6 предписывает использовать для разрешения конфликтов блоки-

■ Таблица 1

Интенсивность	Средняя задержка $M = 16$	
	НСА	Предложенный в работе алгоритм при $q = 1,0$
0,05	1,16	1,145
0,1	1,379	1,357
0,15	1,698	1,674
0,2	2,193	2,15
0,25	3,012	2,86
0,3	4,449	3,892
0,35	7,06	5,33
0,4	11,58	7,211

■ Таблица 2

$M$	$q_{\max}$	$M$	$q_{\max}$
8	1,0	64	0,65
16	1,0	128	0,55
32	0,8	256	0,45

рованный немодифицированный стек-алгоритм (НСА) [3, 4].

Представленные далее численные результаты для предложенного в работе алгоритма получены применительно к описанной выше модели системы путем моделирования системы в течение 10 млн окон, что позволило обеспечить точность результатов 0,01. Результаты приведены для случая, когда значения вероятностей ложных конфликтов  $q_0$  и  $q_1$ , возникающих при выполнении процедуры ПККС, одинаковы между собой и равны некоторому значению  $q$ , т. е.  $q_0 = q_1 = q$ .

В табл. 1 приведены значения средней задержки при различных интенсивностях входного потока для системы, состоящей из 16 абонентов, для НСА (расчет по методике [6]) и предложенного в работе алгоритма при  $q = 1$  (процедура ПККС всегда выполняется с ошибками). Результаты моделирования показывают, что при небольшом числе абонентов (8, 16) предложенный алгоритм всегда выигрывает у НСА, в том числе и при постоянных ошибках в процедуре ПККС.

В табл. 2 приведены максимальные значения вероятностей ложных конфликтов  $q_{\max}$ , возникающих при выполнении процедуры ПККС, при которых предложенный алгоритм имеет выигрыш по задержке у НСА при любой интенсивности входного потока и числе абонентов от 8 до 256.

**Заключение**

В данной работе предложен и исследован алгоритм разрешения конфликтов в системах радиочастотной идентификации, основанный на ис-



пользовании идентификаторов меток и процедуры ПЖКС. Для исследования алгоритма система радиочастотной идентификации была описана моделью системы случайного множественного доступа. С использованием данной модели получено выражение для скорости идентификации (число меток, идентифицируемых в единицу времени) и путем моделирования определена зависимость средней задержки идентификации от ин-

тенсивности входного потока при различном числе абонентов в системе и различных вероятностях ложных конфликтов.

Полученные результаты анализа показывают, что даже при существенных ошибках в работе процедуры ПЖКС предложенный алгоритм дает выигрыш по средней задержке идентификации по сравнению с алгоритмом, описанным в стандарте ISO/IEC 18000-6.

## Литература

1. Джуньян В. Л., Шаньгин В. Ф. Электронная идентификация. Бесконтактные электронные идентификаторы и смарт-карты. — М.: АСТ; ИТ-Пресс, 2004. — 695 с.
2. Information Technology — Radio Frequency Identification (RFID) for Item Management. Part 6: Parameters for Air Interface Communications at 860 MHz to 960 MHz. International Standard ISO/IEC 18000-6, 2004. <http://www.youwokeji.com.cn/down/18000-6.pdf> (дата обращения: 17.02.2012).
3. Цыбаков Б. С., Михайлов В. А. Свободный синхронный доступ пакетов в широкополосный канал с обратной связью // Проблемы передачи информации. 1978. Т. 14. № 4. С. 32–59.
4. Capetanakis J. L. Generalized TDMA. The Multi-Accessing Tree Protocol Channels // IEEE Trans. Commun. 1979. Vol. 27. N 10. P. 1476–1483.
5. Цыбаков Б. С. и др. Множественный доступ с разрешением конфликтов с помощью номеров станций // Проблемы передачи информации. 1992. Т. 28. № 3. С. 27–39.
6. Тюрликов А. М., Марковский С. Г. Использование адресов абонентов для организации доступа к высокоскоростному каналу связи // Информационно-управляющие системы. 2003. № 1. С. 32–38.
7. Yu Y., Giannakis G. B. High-throughput random access using successive interference cancellation in a tree algorithm // IEEE Transactions Inform. Theory. 2007. Vol. 53. N 19. P. 4628–4639.
8. Peeters G., Houdt B., Blondia C. A multi access tree algorithm with free access, interference cancellation and single signal memory requirements // Performance Evaluation. 2007. Vol. 64. N 9–12. P. 1041–1052.
9. Houdt B., Peeters G. FCFS tree algorithms with interference cancellation and single signal memory requirements // Proc. Int. Workshop on Multiple Access Commun. 2008. Vol. 1. P. 1–6.
10. Андреев С. Д., Пустовалов Е. В., Тюрликов А. М. Древоподобный алгоритм разрешения конфликта, устойчивый к неполному погашению интерференции // Автоматика и телемеханика. 2009. № 3. С. 78–96.
11. Katheerja Parveen et al. Analysis of bit grouping algorithm for collision resolution in passive RFID tags // Intern. Journal of Engineering Science and Technology. 2010. Vol. 2(9). P. 4192–4204.
12. Don R. Hush, Cliff Wood. Analysis of Tree Algorithms for RFID Arbitration: IEEE Intern. Symp. on Information Theory // IEEE. 1998. P. 107.
13. Тюрликов А. М., Марковский С. Г. Использование адресов абонентов для разрешения конфликтов в канале с шумом // Информационно-управляющие системы. 2006. № 2. С. 27–37.