

УДК 004.7

ИСПОЛЬЗОВАНИЕ ПОСЛЕДОВАТЕЛЬНОГО ПОГАШЕНИЯ ИНТЕРФЕРЕНЦИИ ПРИ ОРГАНИЗАЦИИ СЛУЧАЙНОГО МНОЖЕСТВЕННОГО ДОСТУПА В ЦЕНТРАЛИЗОВАННЫХ СЕТЯХ

А. В. Винель,

канд. техн. наук, старший научный сотрудник

Санкт-Петербургский институт информатики и автоматизации РАН

А. М. Тюрликов,

канд. техн. наук, доцент

Санкт-Петербургский государственный университет аэрокосмического приборостроения

К. А. Федоров,

магистрант

Санкт-Петербургский государственный электротехнический университет «ЛЭТИ»

Предлагаются алгоритмы случайного множественного доступа для централизованных сетей, использующие последовательное погашение интерференции на физическом уровне при разрешении конфликтов. Приводится метод расчета скорости таких алгоритмов и показывается, что применение погашения интерференции в некоторых случаях позволяет повысить эффективность доступа в централизованных сетях.

Ключевые слова — случайный множественный доступ, разрешение коллизий, централизованные сети, последовательное погашение интерференции.

Введение

Случайный множественный доступ (СМД) широко используется для организации доступа группы абонентов к общему каналу связи в большом числе существующих и перспективных систем. В классических трудах по теории СМД предполагалось, что если два или более пакетов передаются по каналу связи одновременно, то происходит так называемый *конфликт*. Конфликт приводит к тому, что невозможен успешный прием ни одного из передаваемых пакетов [1, 2]. Для такого допущения к настоящему моменту разработано большое число алгоритмов случайного доступа и получены их вероятностно-временные характеристики [3]. Известны также модификации этих алгоритмов для случая, когда конфликт происходит только, если одновременно передаются n ($n > 2$) пакетов, в противном случае прием успешен (например, [4]). В последние годы популярность получило новое направление в разработке алгоритмов СМД, основанное на возможности использовать на подуровне управления до-

ступом к среде (Medium Access Control, MAC) *последовательное погашение интерференции* (successive interference cancellation, SIC) на физическом уровне (PHY) [5–7]. Идея этого подхода состоит в том, что в случае возникновения конфликта сигнал, принятый из канала связи, сохраняется, а затем при успешном принятии каких-либо пакетов, участвовавших в исходном конфликте, используется для восстановления остальных пакетов. Вслед за авторами работ [5–7] в целях упрощения изложения мы будем говорить, что в данном случае происходит «вычитание сигналов», подразумевая при этом, что реальные действия не сводятся к простому вычитанию.

В существующих работах по использованию SIC для разрешения конфликтов за основу берется классическая модель СМД [3] и классический древовидный алгоритм разрешения конфликтов [2], который модифицируется для использования подавления интерференции [5–9]. В данной работе рассматривается *модель централизованной сети* [10–13], а в качестве базового алгоритма для выполняемых модификаций используется *алгоритм*

с распределенной очередью Multi-FS-ALOHA [10]. Он обладает хорошими характеристиками и разработан специально для применения в централизованных сетях [10, 11]. Multi-FS-ALOHA основан на более простом, но менее эффективном алгоритме FS-ALOHA [12] и его модификациях [13].

Целью проведенного анализа явилось определение целесообразности использования последовательного погашения интерференции при реализации СМД в централизованных сетях. Известно, что при использовании SIC в стандартных древовидных алгоритмах разрешения конфликтов [2] в рамках идеализированной классической модели СМД предельная интенсивность входного потока запросов, который может быть передан с конечной средней задержкой (т. е. скорость алгоритма), существенно возрастает. Так, например, при наличии единичного объема сигнальной памяти на физическом уровне каждого абонента скорость возрастает с 0,346 до 0,554, а при неограниченном объеме такой памяти — до 0,693 [7]. Данные цифры демонстрируют тот факт, что в рамках теоретической модели использование последовательного погашения интерференции для разрешения конфликтов представляется очень эффективным. К сожалению, для модели централизованной сети перспектива использования SIC с простым алгоритмом для таких сетей — Multi-FS-ALOHA — оказывается не столь многообещающей: скорость алгоритма Multi-FS-ALOHA/SIC практически не отличается от скорости базового алгоритма.

Именно поэтому нами разработан алгоритм Multi-FS-TREE/SIC, комбинирующий подход Mul-

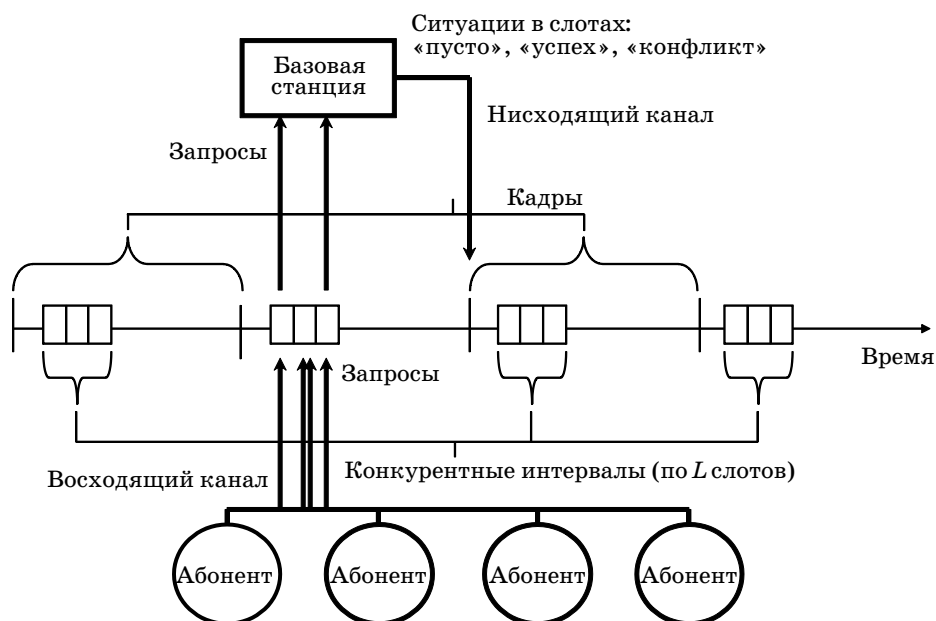
ti-FS-ALOHA для управления структурой кадра с классическим древовидным алгоритмом для разрешения конфликтов [14], базирующимся на процедуре последовательного погашения интерференции [7]. Предлагается численная методика расчета скорости такого алгоритма и показывается, что он превосходит по скорости известные алгоритмы СМД, не использующие SIC, что делает его привлекательным для применения в централизованных сетях передачи данных.

Модель централизованной сети передачи данных

Данная модель (рис. 1) основана на моделях из работ [10–13]. Сформулируем ее основные допущения.

Допущение 1 (бесконечное число абонентов). Рассмотрим систему СМД с одной базовой станцией (БС) и бесконечным числом абонентов, каждый из которых имеет буфер для хранения только одного запроса. Будем называть передачу данных от БС к абонентам *нисходящей*, а от абонентов к БС — *восходящей* передачей. Пусть для нисходящей и восходящей передач используется один и тот же канал связи. Тогда нисходящая передача является ширококвещательной, а при восходящей передаче возникает задача совместного использования этого общего канала абонентами.

Допущение 2 (кадровая структура). Пусть все время передачи по каналу разделено на *кадры* одинаковой длины и структуры. Абоненты передают пакеты в интервале передачи пакетов, предварительно выполняя резервирование времени,



■ Рис. 1. Модель централизованной сети передачи данных

передавая запросы в конкурентном интервале. Правило, в соответствии с которым абоненты передают запросы в конкурентном интервале, будем называть *алгоритмом СМД*. Следуя работе [2], полагаем, что число новых запросов, которые появляются в системе в течение одного кадра, является случайной величиной, распределенной по закону Пуассона с параметром λ : $\Pr\{\text{появление } i \text{ новых запросов}\} = e^{-\lambda} \frac{\lambda^i}{i!}$, где λ — среднее

число запросов, прибывающих в систему за время кадра.

Конкурентный интервал также разбит на равные отрезки времени — *слоты*. Длительность слота равна времени передачи одного запроса. Получив запросы, БС выносит решение о том, как будет распределен интервал передачи пакетов между абонентами, и передает свое решение (расписание) в заголовке следующего кадра. Будем полагать, что число слотов в кадре фиксировано и равно L .

Допущение 3 (бесшумный канал, трючная обратная связь). Канал связи является бесшумным. В каждом слоте возможна одна из трех ситуаций: «успех» — в слоте передавался только один запрос, «пусто» — в слоте не передавалось запросов и «конфликт» — в слоте передавались два и более запросов. БС достоверно определяет ситуации в слотах и безошибочно передает информацию о них всем абонентам в заголовке следующего кадра.

Алгоритмы с распределенной очередью

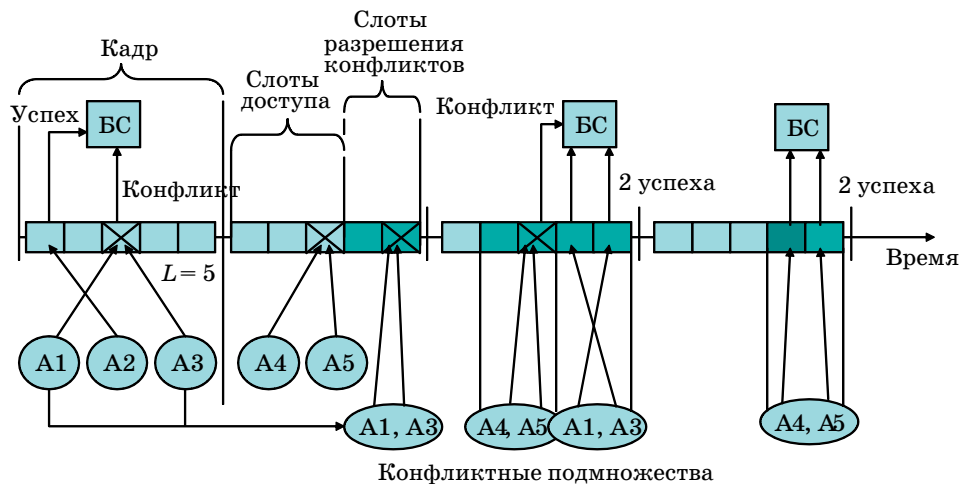
Рассмотрим алгоритмы СМД, в которых множество запросов, попавших в конфликт, разбивается на подмножества. Данные конфликтные подмножества (КП) образуют очередь. Способ

разбиения на подмножества и обслуживания подмножеств определяется конкретным алгоритмом СМД. К алгоритмам с распределенной очередью можно отнести алгоритмы, описанные в работах [10–13], где продемонстрирована их высокая эффективность. В данной работе рассмотрим алгоритм из класса алгоритмов с распределенной очередью Multi-FS-ALOHA [10], который обладает хорошей производительностью независимо от длины конкурентного интервала L и относительной простотой реализации, что делает его привлекательным для использования в перспективных централизованных сетях.

Алгоритм Multi-FS-ALOHA. В алгоритме Multi-FS-ALOHA [10, 11] все L слотов конкурентного интервала разбиваются на два интервала. Размер этих интервалов может изменяться от кадра к кадру. Первый интервал содержит $S(i)$ слотов доступа, второй — $N(i) = L - S(i)$ слотов разрешения конфликтов, где i обозначает номер кадра. Будем считать для удобства, что слоты доступа всегда находятся в начале конкурентного интервала. Первую попытку передачи запроса абонент осуществляет в одном из слотов доступа, который выбирается случайным образом. Слоты разрешения конфликтов предназначены для отправки запросов, которые не были успешно переданы с первого раза (отложенные запросы). При фиксированной величине L Multi-FS-ALOHA имеет только один параметр — максимальный размер интервала разрешения конфликтов, обозначаемый N_{\max} . Значение минимального размера интервала доступа S_{\min} определяется естественным образом: $S_{\min} = L - N_{\max}$.

Алгоритм Multi-FS-ALOHA работает по следующим правилам (рис. 2).

Правило 1 (первая передача — успех). При первой попытке передачи запроса абонент с рав-



■ Рис. 2. Функционирование алгоритма Multi-FS-ALOHA (здесь к началу первого кадра имеется 3 новых запроса (абонента), к началу второго — еще 2)

ной вероятностью выбирает один из $S(i)$ слотов доступа в ближайшем кадре.

Правило 2 (первая передача — конфликт — создание КП). Запросы, которые не были успешно переданы в некотором слоте доступа кадра, образуют КП. Конфликтные подмножества в порядке их возникновения присоединяются к остальным КП распределенной очереди, которые ожидают обслуживания.

Правило 3 (пустая система). Если очередь КП пуста (либо сразу после старта системы, либо после периода низкой активности абонентов), то новые абоненты будут выбирать для передачи случайным образом один из общего числа L слотов в конкурентном интервале кадра. Обнаружение БС конфликтной передачи на одном из этих L слотов будет причиной появления КП (максимальное количество КП, которые могут появиться в таком случае, ограничено числом L). И в следующем кадре i только первые $S(i)$ его слотов будут использованы для новых запросов, а остальные $N(i)$ слотов отводятся для обслуживания КП из очереди.

Правило 4 (обслуживание КП). Слоты разрешения конфликтов используются для обслуживания очереди КП по методу «первый пришел — первый обслужен». Причем два таких слота выделены под передачу запросов из одного КП. Абоненты одного КП для передачи запроса выбирают с одинаковой вероятностью один из двух слотов, выделенных для их КП. Размер интервала разрешения конфликтов в текущем кадре определяется так: $N(i) = \min(2z(i), N_{\max})$, где $z(i)$ — количество КП в начале кадра с номером i . Как только КП будет обслужен, то выделенные ему 2 слота будут назначены другому КП из очереди, если в очереди еще будут КП. Признак, по которому определяется, что в КП не осталось запросов, — это отсутствие среди выделенных для него слотов конфликтов.

Более подробно реализация алгоритма Multi-FS-ALOHA рассмотрена в работах [10, 11].

Алгоритм Multi-FS-ALOHA/SIC. Наличие на физическом уровне возможности погашения интерференции приводит к следующим видоизменениям в допущениях рассмотренной модели. Допустим допущение 3 следующим образом.

Допущение 3 (обратная связь при погашении интерференции). На физическом уровне БС функционирует процедура последовательного погашения интерференции. Объем обратной связи, доступной на MAC-подуровне БС, расширяется с учетом возможностей данной процедуры (состав информации описан далее). Вся информация обратной связи доступна абонентам к концу каждого кадра.

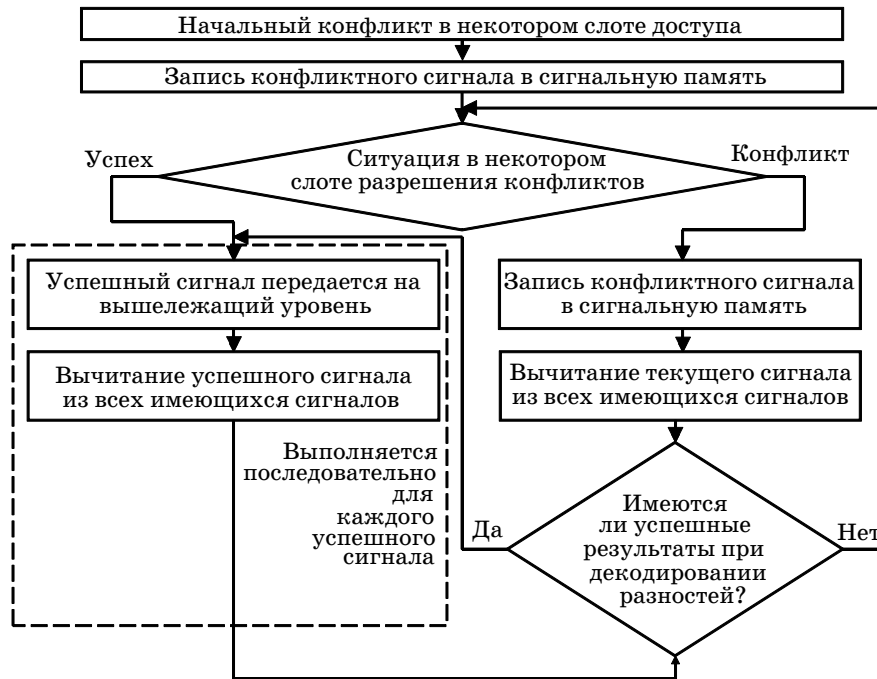
Опишем подробнее, что представляет собой процедура последовательного погашения интер-

ференции. БС в каждом из слотов получает аналоговые сигналы, преобразует их в цифровую форму и предпринимает попытку демодуляции и декодирования принятых сигналов. При этом возможно два исхода декодирования — конфликтный сигнал и успешный сигнал. В случае конфликта оцифрованные отчеты сигнала сохраняются в свободной ячейке *сигнальной памяти на странице данного КП*. Далее возможно два принципиальных варианта реализации SIC: погашение интерференции применяется только, если имел место успех, или же погашение интерференции применяется во всех случаях. Будем считать, что используется второй вариант. Таким образом, используется следующее допущение.

Допущение 4 (погашение при успехе и конфликте). Метод SIC применяется при декодировании в слоте и успешного, и конфликтного сигналов. При получении сигнала всегда предпринимается попытка декодирования разностей данного сигнала и всех остальных сигналов, хранящихся в странице сигнальной памяти, соответствующей данному конфликтному подмножеству. Причем, если вычитается успешный сигнал, то 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. Все успешно восстановленные пакеты данных передаются на вышележащие уровни, а соответствующие им хранимые сигналы удаляются из памяти. Концептуально работа последовательного погашения интерференции показана на рис. 3. Следующее допущение объясняет использование упомянутой выше сигнальной памяти.

Допущение 5 (неограниченный объем сигнальной памяти). На физическом уровне БС имеется неограниченное количество ячеек памяти для хранения принимаемых из канала связи сигналов. Причем память разбита на страницы — каждому КП соответствует своя страница. Допущение о неограниченном объеме сигнальной памяти является практически нереализуемым. Тем не менее, в этой работе мы рассматриваем данный предельный случай, чтобы оценить максимально возможный выигрыш от погашения интерференции. При практической реализации данного подхода в реальных сетях выигрыш будет меньше в силу ограниченности реальной памяти.

Дополнительные возможности физического уровня приводят к необходимости уточнять правила работы алгоритма Multi-FS-ALOHA. Будем называть алгоритм Multi-FS-ALOHA, использующий погашение интерференции, Multi-FS-ALO-



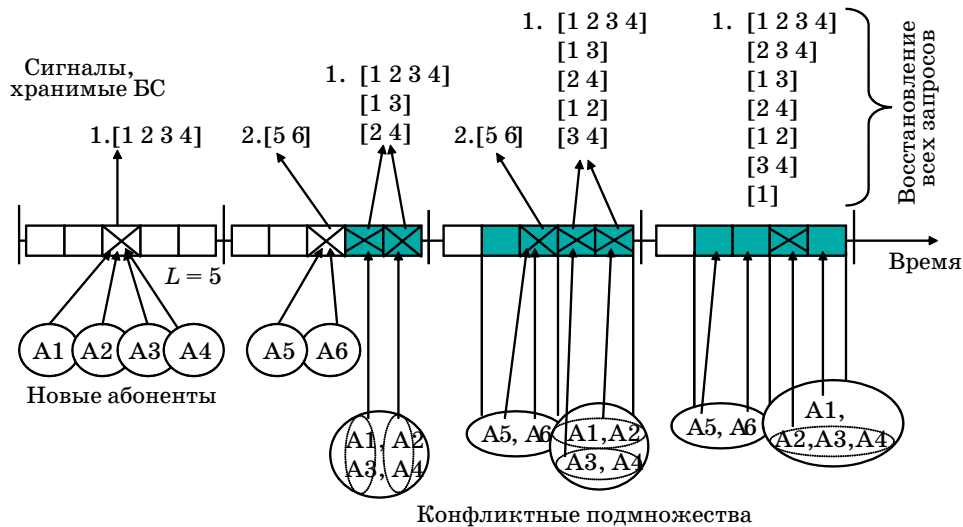
■ Рис. 3. Погашение интерференции при успехе и конфликте для некоторого КП (некоторая страница сигнальной памяти)

НА/SIC. Заметим, что в Multi-FS-ALOHA/SIC должен использоваться следующий признак окончания обслуживания КП — отсутствие конфликтов на двух соответствующих слотах разрешения конфликтов. При практической реализации в некоторых случаях фактический момент окончания обслуживания КП и момент, когда БС узнает об этом, могут не совпадать. Рассмотрим пример. В КП изначально находятся пять запросов. Далее при разрешении конфликтов на двух слотах возникают следующие ситуации (цифрами обозначены номера абонентов): [1 2 3 | 4 5], [1 2 | 3 4 5], [1 2 4 | 3 5], [1 | 2 3 4 5]. После этой последовательности передач появляется возможность восстановить все оставшиеся четыре пакета. Однако БС неизвестна кратность исходного конфликта, поэтому обслуживание данного КП закончится лишь в следующем кадре, где в двух рассматриваемых слотах будет ситуация [пусто, пусто]. Таким образом, разрешение конфликта в данном примере занимает 6 кадров, включая кадр, в котором конфликт изначально возник. В рассматриваемых в данной работе идеализированных условиях бесшумного канала и идеального погашения интерференции признаком окончания обслуживания КП является декодирование всех пакетов из первоначального конфликтного сигнала. Однако необходимо иметь в виду, что для того, чтобы гарантировать работоспособность алгоритма в реальных условиях, необходимо использовать в качестве признака окончания обслуживания

КП отсутствие конфликтов на соответствующих слотах.

Заметим также, что для реализации рассматриваемого подхода БС должна в явном виде в нисходящем канале сообщать MAC-адреса (или любые другие уникальные идентификаторы абонентов), пакеты которых были успешно декодированы с помощью процедуры SIC. Это необходимо, поскольку абонент не может ориентироваться на традиционную информацию об обратной связи, так как «успех» в соответствующем слоте больше не является необходимым условием успешной передачи. Описанные особенности работы Multi-FS-ALOHA/SIC проиллюстрированы на рис. 4.

Нетрудно показать, что использование погашения интерференции в алгоритме Multi-FS-ALOHA/SIC в рамках наших допущений не имеет никакого смысла при разрешении конфликтов кратностей два и три. Действительно, при разрешении конфликта кратности два возможен либо повторный конфликт, либо успешная передача обоих абонентов. При разрешении конфликта кратности три если происходит успешная передача в одном из слотов, то применение процедуры SIC позволит получить уже и так имеющееся у БС содержимое второго слота. Поскольку именно конфликты маленьких кратностей являются наиболее вероятными, то можно ожидать, что Multi-FS-ALOHA/SIC не будет существенно превосходить по характеристикам обычный Multi-FS-ALOHA.



■ **Рис. 4.** Функционирование алгоритма Multi-FS-TREE/SIC (здесь к концу четвертого кадра КИ с четырьмя запросами обслуживается за счет использования SIC)

Возникает вопрос, существует ли алгоритм СМД для централизованных сетей, для которого использование SIC имеет смысл. Ответ на него является положительным. Далее рассматривается алгоритм Multi-FS-TREE/SIC, в котором в качестве алгоритма разрешения конфликтов используется древовидный алгоритм [14] в модификации, использующей погашение интерференции SICТА [5]. Последние результаты для древовидного алгоритма СМД при использовании погашения интерференции применительно к классической модели СМД приведены в работе [9].

Алгоритм Multi-FS-TREE/SIC. Чтобы получить алгоритм Multi-FS-TREE из алгоритма Multi-FS-TREE/SIC, видоизменим правило 4 следующим образом.

Правило 4 (обслуживание КИ). Слоты разрешения конфликтов используются для обслуживания очереди КИ по методу «первый пришел — первый обслужен». Причем *один* такой слот выделен под передачу запросов из одного КИ. Размер интервала разрешения конфликтов в текущем кадре определяется так: $N(i) = \min(z(i), N_{\max})$, где $z(i)$ — как и раньше — количество КИ в начале кадра с номером i . Ограничения на значения параметров для Multi-FS-TREE следующие: $S_{\min} \geq 1$ и $N_{\max} \geq 1$. Как только КИ будет обслужен, то выделенный ему слот будет назначен другому КИ из очереди, если в очереди еще будут КИ. Абоненты одного КИ при передаче запросов используют древовидный алгоритм разрешения конфликтов.

Работу древовидного алгоритма можно описать следующим образом. При возникновении конфликта в некотором слоте кадра под номером t абоненты, вступившие в конфликт, разбиваются

на два подмножества. Например, каждый из участвующих в конфликте абонентов «бросает монету» и выбирает с равной вероятностью одно из них. Абоненты первого подмножества передают свои пакеты в выделенном для них слоте разрешения конфликта кадра $t + 1$, и если этот слот был пуст или в нем была успешная передача, то абоненты второго подмножества передают в кадре $t + 2$. В противном случае, если в кадре $t + 1$ возникает еще один конфликт, первое подмножество снова разбивается на два и повторяет описанные действия. В это время второе подмножество «ожидает» разрешения конфликта между абонентами первого подмножества. Описанную процедуру удобно представлять в виде двоичного дерева, в котором корневая вершина соответствует множеству абонентов, попавших в исходный конфликт. Остальные вершины соответствуют подмножествам абонентов, передающих свои пакеты в каждом кадре. Данные подмножества могут быть и пустыми. Ребра дерева отражают процесс разбиения, т. е. из вершин с двумя и более абонентами отходит 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, следуя работе [9], использовать термины *левый* и *правый* слот дерева разрешения конфликта.

Для работы алгоритма Multi-FS-TREE/SIC необходимо несколько расширить информацию обратной связи (допущение 3), передаваемую БС абонентам к началу каждого кадра (естественно, эта информация должна поступать на подуровень управления доступом к среде от физического

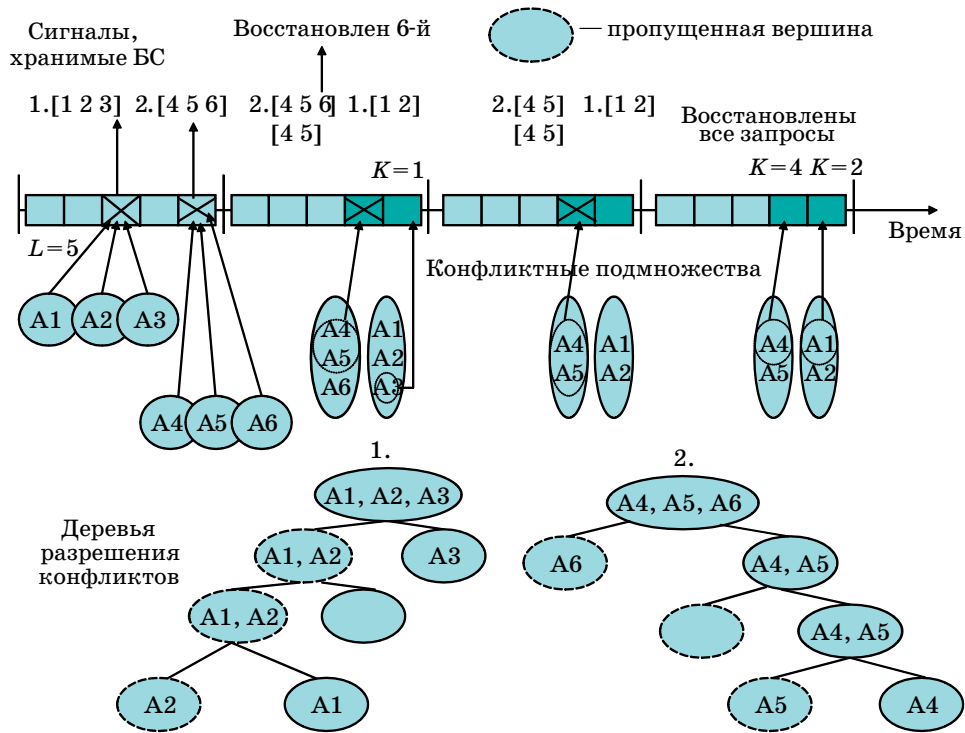


Рис. 5. Функционирование алгоритма Multi-FS-TREE/SIC

уровня): «конфликт», «пусто» и « K -успех», где K — число успешно декодированных запросов, сложенное с числом левых слотов в дереве разрешения конфликтов, помеченных как пустые после выполнения процедуры погашения интерференции ($K \geq 1$). БС может рассчитывать значение K , суммируя количество успешных декодирований в ходе выполнения процедуры SIC между двумя последовательными успешными передачами для некоторого КП.

Работа алгоритма Multi-FS-TREE/SIC показана на рис. 5. В этом примере при обслуживании второго КП: к концу четвертого кадра передается значение $K = 4$, так как успешно получен запрос № 4, дважды восстановлены запросы № 5 и 6 (последний восстановлен еще во втором кадре).

Формальные правила работы древовидного алгоритма с последовательным погашением интерференции SICTA, используемого в Multi-FS-TREE/SIC, приведены в работе [5]. Для упрощения реализации алгоритма SICTA при погашении интерференции можно не реализовывать вычитание двух конфликтных сигналов, а запускать процедуру только при успешной передаче. В этом случае задержка передачи некоторых пакетов может возрасти (в примере на рис. 4 запрос № 6 будет восстановлен только в четвертом, а не во втором кадре), но дерево разрешения конфликтов не изменится.

Метод расчета скорости алгоритмов и численные результаты

Главными характеристиками любого алгоритма СМД являются задержка и скорость передачи. Под *задержкой* будем понимать интервал времени с момента появления запроса у абонента до момента сообщения БС о его успешной передаче. Причем, для упрощения в задержку не будем включать время с момента появления запроса до ожидания начала ближайшего кадра. Среднее значение такой величины для некоторого алгоритма СМД будем называть *средней задержкой* и обозначать D .

Скоростью R алгоритма СМД будем называть верхнюю грань интенсивности входного потока, которая передается алгоритмом с конечной средней задержкой [11]:

$$R = \sup\{\lambda : \bar{D} < \infty\} / L. \quad (1)$$

Метод расчета скорости для широкого класса алгоритмов СМД с распределенной очередью [11] приведен далее применительно к рассмотренным алгоритмам. Опишем работу алгоритмов в терминах теории массового обслуживания. Для начала рассмотрим работу алгоритмов при следующих параметрах: $L = 2$, $N_{\max} = 1$ для алгоритмов, основанных на АЛОНА, и $L = 3$, $N_{\max} = 2$ для алгоритмов, основанных на древовидном подходе. В этом

случае у нас имеется бесконечная очередь КП и один обслуживающий прибор. Система является синхронной, и кадр является единицей времени. Временные интервалы между прибытиями КП являются одинаково распределенными независимыми случайными величинами. То же самое утверждение верно и для временных интервалов, необходимых для обслуживания одного КП. Таким образом, имеем очередь типа $GI/GI/1$, которая стабильна [15], если выполняется неравенство

$$\Lambda < \mu, \quad (2)$$

где Λ есть среднее число конфликтных подмножеств, поступающих в систему за время кадра; среднее время, необходимое для обслуживания одного КП: $\Psi = 1/\mu$.

Скорость алгоритма (1) может быть рассчитана посредством вычисления λ_{\max} , т. е. такого максимального значения λ , при котором выполняется неравенство (1). Применяя аргументацию из работы [16] («правило насыщения»), можно показать, что существует интенсивность входного потока запросов $\lambda < \lambda_{\max}$, при которой очередь КП всегда не пуста, и в то же время система стабильна (т. е. интенсивности входного и выходного потоков равны). Таким образом, при анализе скорости можно полагать, что в ходе работы системы конкурентный интервал всегда разделен на $S = 1$, $N = 2$ слотов для алгоритма Multi-FS-ALOHA/SIC и $S = 1$, $N = 1$ для Multi-FS-TREE/SIC. За длительность кадра в распределенную очередь поступает либо ноль, либо одно КП, т. е.

$$\Lambda = \Pr\{\text{поступление одного КП}\} = 1 - e^{-\lambda} - \lambda e^{-\lambda}. \quad (3)$$

Пусть

$$T(x) = \sum_{k=2}^{\infty} T_k \frac{x^k}{k!} e^{-x}, \quad (4)$$

где T_k — среднее число кадров, необходимых для обслуживания КП, состоящего из k ($k > 1$) запросов.

Можно показать, что Ψ рассчитывается следующим образом:

$$\Psi = \frac{T(\lambda)}{1 - e^{-\lambda} - \lambda e^{-\lambda}}. \quad (5)$$

Из формул (2)–(5) заключаем, что рассмотренная система массового обслуживания стабильна для таких значений λ , при которых выполняется неравенство

$$T(\lambda) < 1. \quad (6)$$

Теперь рассмотрим случай произвольных значений L и N_{\max} , а следовательно, произвольных значений S и N при разбиении кадра. В этом ва-

рианте входной поток КП в S раз выше, чем для простейшего случая, а число обслуживающих устройств равно $n = N/2$ для Multi-FS-ALOHA/SIC и $n = N$ для Multi-FS-TREE/SIC. Отсюда можно видоизменить формулу (2):

$$S\Lambda < n\mu. \quad (7)$$

Таким образом, из выражений (2)–(6) получаем, что скорость R может быть рассчитана как $R(S, N) = \lambda_{\max} / L$, где λ_{\max} — это максимальная интенсивность λ , при которой выполняется условие

$$T(\lambda / S) < n / S. \quad (8)$$

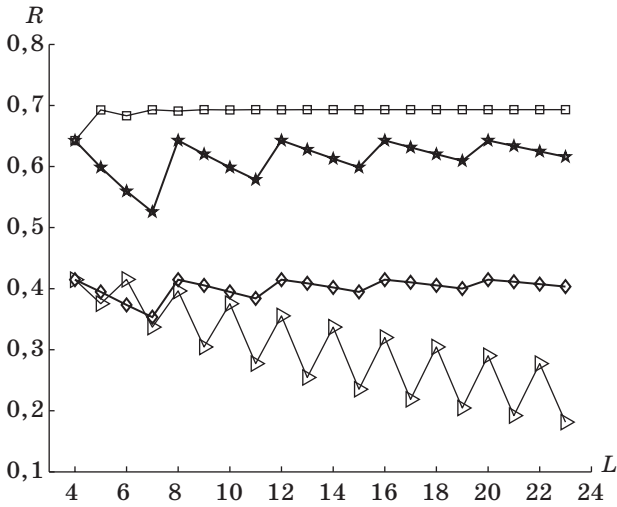
Для различных алгоритмов величины T_k в выражении (4) будут различными. Формулы для их расчета применительно к алгоритму Multi-FS-ALOHA приводятся в работе [10]. Получить выражения для среднего времени, необходимого для обслуживания КП кратности большей трех, для алгоритма Multi-FS-ALOHA/SIC не представляется простой задачей. Поэтому для оценки скорости Multi-FS-ALOHA/SIC применено совмещенное аналитическое и имитационное моделирование (T_k рассчитываются имитационно). Как и ожидалось, скорости алгоритмов Multi-FS-ALOHA и Multi-FS-ALOHA/SIC практически не отличаются. Что касается средней задержки передачи пакетов, то на входных интенсивностях, меньших скорости алгоритма, вероятность конфликтов высоких кратностей также мала, т. е. задержки, обеспечиваемые алгоритмами Multi-FS-ALOHA и Multi-FS-ALOHA/SIC, также отличаются несущественно.

Покажем, как рассчитывать величины T_k для предложенного нами алгоритма Multi-FS-TREE/SIC, следуя подходу из работ [9] и [17].

Рассмотрим классический древовидный алгоритм разрешения конфликтов без погашения интерференции [14]. Пусть v — число вершин в дереве разрешения конфликта по такому алгоритму, а среднее время разрешения конфликта кратности k обозначим $t_k = E[v]$. Число успешных, конфликтных и пустых слотов в дереве обозначим за v_s , v_c и v_e соответственно. Поскольку $v_s + v_c + v_e = v$, воспользуемся основным свойством дерева разрешения конфликтов [17], чтобы получить следующие соотношения:

$$v_s = k; v_c = \frac{v-1}{2}; v_e = \frac{v+1}{2} - k. \quad (9)$$

Будем рассматривать дерево разрешения конфликта алгоритма SICТА как дерево разрешения конфликта классического древовидного алгоритма, в котором время просмотра некоторых вер-



■ Рис. 6. Зависимость скорости алгоритмов Multi-FS-ALOHA и Multi-FS-TREE/SIC от параметра N_{max} при различных длинах L :
 □□□□ — Multi-FS-TREE/SIC ($N_{max} \sim L$);
 ★★ ★★ — Multi-FS-TREE/SIC ($N_{max} \sim L/2$);
 ◇◇◇◇ — Multi-FS-ALOHA ($N_{max} \sim L/2$);
 ▲▲▲▲ — Multi-FS-ALOHA ($N_{max} \sim L$)

шин равно нулю, что обеспечивается функционированием процедуры SIC. Заметим, что при работе алгоритма SICTA содержимое первого слота левого поддерева для любой вершины дерева *всегда* устанавливается с помощью процедуры SIC без просмотра соответствующей вершины. Таким образом, для нахождения среднего числа вершин с ненулевым временем просмотра в дереве разрешения конфликта по SICTA необходимо вычесть из среднего числа вершин в дереве классического алгоритма (t_k) половину успешных, конфликтных (кроме начального слота доступа) и пустых слотов [9], и с учетом соотношений (9), а также учитывая, что необходимо исключить начальный слот доступа, получаем

$$T_k = t_k - \frac{1}{2}E[v_s] - \frac{1}{2}(E[v_c] - 1) - \frac{1}{2}E[v_e] - 1 = \frac{t_k - 1}{2}. \quad (10)$$

В свою очередь, несложно показать, что среднее время t_k для классического древовидного алгоритма можно рассчитать по следующей рекуррентной формуле:

$$t_k = \frac{1 + 2^{1-k} \sum_{i=0}^{k-1} C_k^i t_i}{1 - 2^{1-k}}, \quad k \geq 2; \quad t_0 = t_1 = 1. \quad (11)$$

Численные результаты для рассмотренных алгоритмов приведены на рис. 6.

Заключение

В работе было показано, что использование последовательного погашения интерференции в централизованных сетях с применением эффективного алгоритма случайного доступа Multi-FS-ALOHA не позволяет получить существенного улучшения вероятностно-временных характеристик системы даже при бесконечном объеме доступной сигнальной памяти и безошибочном погашении при успехе и конфликте. По нашему мнению, использование алгоритма Multi-FS-ALOHA/SIC на практике нецелесообразно, поскольку существенное повышение сложности реализации приводит к незначительному повышению производительности. Перспективным алгоритмом для организации СМД в централизованных сетях является исследованный в данной работе алгоритм Multi-FS-TREE/SIC. Направлением дальнейших исследований может являться анализ рассмотренных в статье алгоритмов для условий канала с шумом и неточного погашения интерференции.

Литература

1. Abramson N. The ALOHA system — Another alternative for computer communications: Proc. of Fall Joint Computer Conference. 1970. Vol. 37. P. 281–285.
2. Tsybakov B. S. Survey of USSR Contributions to Random Multiple-Access Communications // IEEE Transactions on Information Theory. Mar. 1985. Vol. IT-31. N. 2. P. 143–165.
3. Bertsekas D., Gallager R. Data Networks, Englewood Cliffs. NJ: Prentice-Hall, 1992.
4. Tsybakov B. S. Packet Multiple Access for Channel With Binary Feedback, Capture, and Multiple Reception // IEEE Transactions on Information Theory. June 2004. Vol. 50. N 6. P. 1073–1085.

5. Yu Y., Giannakis G. B. SICTA: A 0.693 contention tree algorithm using successive interference cancellation: Proc. of INFOCOM-2005 Conf. Miami, FL, Mar. 13–17. 2005. P. 1908–1916.
6. Wang X., Yu Y., Giannakis G. B. A Robust High-Throughput Tree Algorithm Using Successive Interference Cancellation // IEEE Transactions on Communications. Dec. 2007. Vol. 55. N 12. P. 2253–2256.
7. Andreev S., Pustovalov E., Turlikov A. SICTA Modifications with Single Memory Location and Resistant to Cancellation Errors: Proc. of NEW2AN Conf. S.-Petersburg, Russia, Sept. 2008. P. 13–24.
8. Andreev S., Pustovalov E., Turlikov A. Tree algorithms with free access and interference cancellation

- in presence of cancellation errors: Proc. of International Symp. on Wireless Personal Multimedia Communications. Lapland, Finland, 2008. P. 1–5.
9. **Андреев С. Д., Пустовалов Е. В., Тюрликов А. М.** Древоидный алгоритм разрешения конфликта, устойчивый к неполному погашению интерференции // Автоматика и телемеханика. 2009. № 3. С. 78–96.
 10. **Kobliakov V., Turlikov A., Vinel A.** Distributed Queue Random Multiple Access Algorithm for Centralized Data Networks: Proc. of the 10th IEEE International Symposium on Consumer Electronics — ISCE'06. St.-Petersburg, Russia, 2006. P. 290–295.
 11. **Винель А. В., Кобляков В. А., Тюрликов А. М.** Класс алгоритмов случайного множественного доступа с очередью для централизованных сетей передачи данных // Информационные технологии. 2007. № 5. С. 32–41.
 12. **Vazquez-Cortizo D., Garcia J., Blondia C., Van Houdt B.** FIFO by Sets ALOHA (FS-ALOHA): A Collision Resolution Algorithm for the Contention Channel in Wireless ATM Systems // Performance Evaluation. Istanbul, 1999. P. 401–427.
 13. **Vazquez-Cortizo D., Blondia C., Garcia J.** Fs-alo-ha++, a collision resolution algorithm with QoS support for the contention channel in multi-service wireless LAN: Proc. of Globecom-99 Conf. Dec. 1999. P. 2773–2777.
 14. **Tsybakov B. S.** Survey of USSR contributions to random multiple-access communications // IEEE Transactions on Information Theory. 1985. Vol. IT-31. P. 143–165.
 15. **Kleinrock L.** Queuing Systems. Vol. 1: Theory. N. Y.: John Wiley and Sons, 1975.
 16. **Baccelli F., Foss S.** On the saturation rule for the stability of queues // Journal of Applied Probability. 1995. Vol. 32. P. 494–507.
 17. **Евсеев Г. С., Тюрликов А. М.** Взаимосвязь характеристик блокированных стек-алгоритмов случайного множественного доступа // Проблемы передачи информации. 2007. Т. 43. № 4. С. 83–92.

ОАО «Центр технологии судостроения и судоремонта»
 Институт информатики и автоматизации РАН
 Отделение нанотехнологий и информационных технологий РАН
 Российский национальный комитет по индустриальной и прикладной математике

**ЧЕТВЕРТАЯ ВСЕРОССИЙСКАЯ НАУЧНО-ПРАКТИЧЕСКАЯ КОНФЕРЕНЦИЯ
 «ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ. ТЕОРИЯ И ПРАКТИКА» (ИММОД–2009)
 21–23 октября 2009 г.**

Место проведения: Санкт-Петербург, Дворцовая наб., д. 26, Дом ученых им. М. Горького РАН

Генеральный спонсор
 Компания XJ Technologies

Спонсоры
 Отделение нанотехнологий и информационных технологий РАН
 Российский фонд фундаментальных исследований
 Комитет по науке и высшей школе при Администрации Санкт-Петербурга

Задачи конференции
 Развитие и обобщение теории имитационного моделирования и квалиметрии моделей
 Обмен опытом и обсуждение результатов исследований и практических приложений имитационного моделирования
 Обмен опытом применения имитационного моделирования для решения научных и практических задач
 Распространение опыта обучения теории и практике имитационного моделирования

Направления работы конференции
 Теоретические основы и методология имитационного моделирования
 Методы оценивания качества моделей
 Методы и системы распределенного моделирования
 Моделирование глобальных процессов

Средства автоматизации и визуализации имитационного моделирования
 Системная динамика (с обязательным наличием имитационной составляющей)
 Практическое применение моделирования и инструментальных средств автоматизации моделирования, принятие решений по результатам моделирования
 Имитационное моделирование в обучении и образовании

Контрольные сроки
 Материалы докладов в секретариат конференции предоставляются не позднее **17 сентября 2009** года (только в этом случае гарантируется их размещение в сборнике материалов конференции). Заявки на выступление с докладом (в форме извещения о намерении) и стендовую демонстрацию принимаются до **12 октября 2009** года.

Дополнительная информация и справки
 ОАО «Центр технологии судостроения и судоремонта»
 Россия, 198095, Санкт-Петербург,
 Промышленная ул., д. 7
 Плотников Александр Михайлович
 Долматов Михаил Анатольевич
 Эл. почта: immod2009@crist.ru
 Информация на сайтах: www.gpss.ru, www.crist.ru,
 www.spiiras.nw.ru, www.xjtek.com