

ИДЕНТИФИКАЦИЯ RFID-МЕТОК С СЕНСОРНОГО ПОЛЯ С ИСПОЛЬЗОВАНИЕМ БПЛА

Абрамян В.Л.¹

(ФГБУН Институт проблем управления им. В.А. Трапезникова
РАН, Москва)

Технология RFID (Radio Frequency Identification) находит широкое применение в научных и инженерных задачах, включая взаимодействие с беспилотными летательными аппаратами (БПЛА) для идентификации объектов в труднодоступных зонах. Одними из ключевых характеристик таких систем являются вероятность успешного считывания метки и время, затраченное на считывание, которые могут быть количественно описаны с помощью моделей полумарковских процессов. В данной работе рассматриваются два принципиально различных сценария. В первом случае предполагается разреженное размещение меток, исключающее коллизии. Для него предложена аналитическая модель взаимодействия считывателя и метки. Во втором случае метки размещены плотно, что приводит к коллизиям; для оценки параметров работы системы в таких условиях разработана дискретно-событийная имитационная модель. В модели учитываются особенности протокола EPC Gen2, параметры радиоканала, геометрия размещения меток и стратегия чтения данных. Проведено сравнение аналитических и имитационных результатов, а также исследовано влияние плотности меток, объёма данных и высоты полёта на характеристики чтения.

Ключевые слова: RFID, БПЛА, сенсорная метка, симуляция, аналитика.

1. Введение

Радиочастотная идентификация (RFID) — это технология маркировки и автоматической идентификации объектов, при которой данные, хранящиеся в RFID-метках, считываются или записываются с помощью радиосигналов. Базовая RFID-система состоит минимум из двух компонентов: меток (радиоэлектронных устройств, закрепляемых на объектах) и считывателей (активных устройств, инициирующих обмен данными). Одной из ключевых особенностей RFID-систем является использование, как правило, пассивных меток, не имеющих собственного источника питания. Эти устройства активируются сигналом считывателя, что делает их простыми и недорогими, благодаря чему они

¹ Абрамян Вильмен Леонович, м.н.с. (abramian.vl@phystech.edu).

широко применяются в повседневной жизни, например, в розничной торговле (бирки одежды), библиотечном деле (метки на книгах) и других сферах [2]. Помимо традиционных областей RFID также активно используется в транспорте [5, 10, 11]. Потенциал технологии позволяет расширять сферы её использования: RFID-системы находят применение в строительстве, логистике, отслеживании и контроле движения специализированного транспорта, а также его парковке [17, 19].

Одним из потенциальных применений RFID-систем является использование их совместно с беспилотными летательными аппаратами (БПЛА), которые находят широкое применение как в гражданской, так и в оборонной промышленности для решения множества задач [1, 3, 12]. Один из популярных сценариев включает установку считывателя на БПЛА для опроса меток, размещённых на идентифицируемых объектах [7]. Например, в работах [14, 15] рассматривается использование БПЛА для сбора данных о растениях на сельскохозяйственных полях. В этих исследованиях описаны эксперименты по определению оптимальной высоты полета БПЛА, а также других параметров, влияющих на вероятность успешного чтения RFID-меток. Авторы в [15] также представили эксперименты, направленные на изучение различных алгоритмов полета БПЛА с RFID-считывателем для оптимизации скорости пролета над каждой меткой с целью её считывания с заданной вероятностью.

Определение состояния почвы на больших территориях также может быть выполнено с использованием БПЛА, которые опрашивают метки со встроенными датчиками (температуры, влажности и других параметров) [4]. Датчики RFID-меток, установленные на БПЛА, могут использоваться для сбора данных о температуре или влажности воздушных масс [8, 9]. Пространственную карту окружающей среды можно строить с использованием роя сверхмалых БПЛА, оснащённых бортовыми датчиками в RFID-метках, и более крупного дрона с RFID-считывателем [8]. Также исследователи уделяют внимание вопросам беспроводного канала связи в подобных системах [18, 20]

Тем не менее упор в перечисленных работах сделан в первую очередь на проведении реальных и численных экспериментов с находящимися в движении считывателем или меткой без построения аналитической модели подобного взаимодействия, что может осложнить масштабирование полученных в этих работах результатов и поиск потенциально новых применений БПЛА совместно с RFID-системами. В то же время в работах, в которых строятся аналитические модели взаимодействия метки и считывателя (например, [6]), напротив, не приводится экспериментальная верификация полученных моделей. В связи с этим актуальной задачей является построение аналитической модели взаимодействия RFID-метки с движущимся RFID-считывателем и её сравнение с результатом численного эксперимента, реализованного с помощью имитационной модели. При этом выбор в сторону численного (а не реального) эксперимента обусловлен большей гибкостью к модификации программируемой модели по сравнению с реальной, а также более простой реализацией.

В настоящей статье рассматривается задача оценки времени считывания данных с сенсорных RFID-меток с помощью считывателя, установленного на БПЛА. Рассматривается технология UHF RFID, работающая в диапазоне 860–960 МГц и регламентируемая стандартом EPC Class 1 Gen.2 [16]. В данной работе проводится анализ различных сценариев (стратегий) чтения сенсорных меток при разных расстояниях между считывателем и меткой, а также объёме считываемых данных. Проводится оценка среднего времени, необходимого для считывания данных с метки в рамках каждой стратегии. При этом всесторонний анализ энергопотребления БПЛА в процессе полета и считывания меток выходит за рамки данного исследования. В разделе 2 перечисляются используемые в настоящей статье методы и их границы применимости, в том числе описывается зависимость параметров сигнала метки от физических характеристик (высоты полёта, мощности сигнала и пр.), анализируется влияние использования различных стратегий чтения сенсорных меток и времени задержки чтения данных. В разделе 3 описывается аналитическая мо-

дель расчёта среднего времени успешного чтения RFID-метки. Рассматриваются полумарковские процессы как в случае отсутствия повторной передачи сигнала от метки, так и в случаях с повторной передачей. Раздел 4 посвящена сравнению полученных при помощи аналитической модели предсказаний с результатами, получаемыми из имитационной модели. В разделе 5 рассматривается имитационное моделирование системы с коллизиями. В заключении приводятся выводы и дальнейшие потенциальные направления исследования.

2. Применяемые методы

2.1. ОПИСАНИЕ ИССЛЕДУЕМОЙ СИСТЕМЫ

Обозначим высоту расположения RFID-считывателя на БПЛА как h , а расстояние до метки по земле – как x . Под расстоянием по земле подразумевается расстояние от метки до точки на земле, находящейся строго под БПЛА (рис. 1). Предположим, что БПЛА либо замирает в заданной точке, либо движется с постоянной скоростью, при этом антенна считывателя направлена строго вниз, антенна метки — строго вверх.

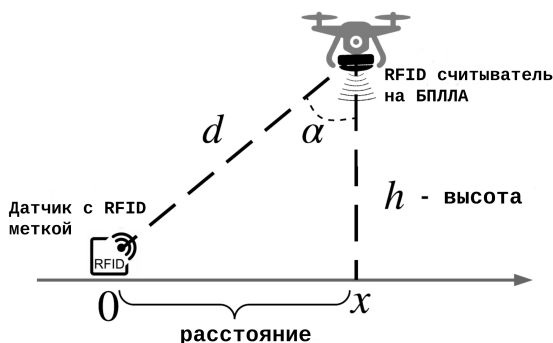


Рис. 1. Геометрия системы, включающей RFID считыватель на БПЛА и сенсорную метку

Аналитические формулы для расчёта мощности излучения, падающего на метку и считыватель, а также потерь в канале, бы-

ли взяты из работы [13]. При этом если полученная меткой мощность сигнала меньше пороговой чувствительности, она не включается и не может ответить считывателю. Численный пример расчета мощности и потерь на пути показан на рис. 2.

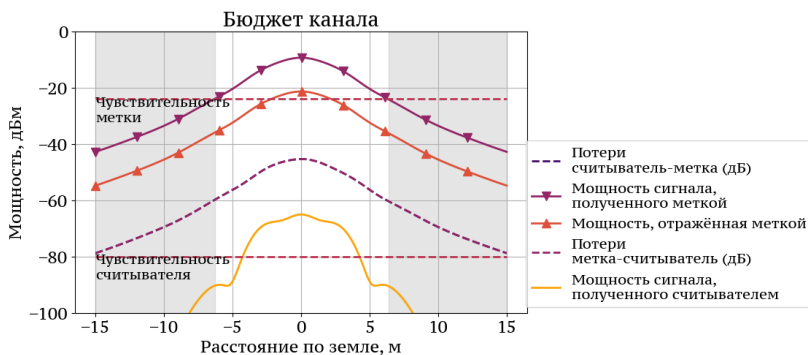


Рис. 2. Оценка бюджета связи при размещении RFID считывателя на БПЛА на разных расстояниях от сенсорной метки на высоте 5 м. Пороговая чувствительность метки составляет -22 дБм, передаваемая мощность считывателя составляет 31 дБм

Потери в канале зависят от расстояния между считывателем и меткой, а также от угла между осью антенны и направлением луча прямой видимости. Таким образом, максимальное расстояние по земле, на котором возможно соединение, зависит от текущей высоты БПЛА, а также от мощности передачи считывателя и чувствительности метки.

Мощность, принимаемая считывателем, подвергается влиянию потерь на пути дважды: в прямом и обратном направлениях, так как метка не имеет собственного источника питания и может использовать только энергию, полученную от считывателя.

Из-за обратного рассеяния (способ, с помощью которого метка передаёт радиосигнал) мощность сигнала, принимаемая считывателем от метки, значительно ниже мощности, принимаемой меткой от считывателя, поэтому ошибки гораздо более ве-

роятны на стороне считывателя при приёме ответа метки. В данной работе предполагается, что если метка имеет достаточную мощность для включения, то она всегда способна правильно декодировать команду считывателя. Поэтому любая ошибка связи возникает на стороне считывателя из-за ненулевой вероятности битовой ошибки (BER).

2.2. СТРАТЕГИИ ЧТЕНИЯ СЕНСОРНЫХ МЕТОК

Рассмотрим случай, при котором метки расположены на достаточно большом расстоянии друг от друга. Тогда можно сделать предположение о том, что раунд инвентаризации (т.е. законченный набор обменов сигналами между считывателем и метками) может содержать только один слот ($Q = 0$) [16], причём в начале нового раунда инвентаризации считыватель выбирает такие флаги сессии, чтобы метка в его области чтения всегда могла принять участие в раунде. Однако из-за ненулевого BER некоторые ответы сенсорной метки могут быть потеряны. Ключевой характеристикой здесь является время, необходимое для идентификации метки и получения данных с датчика.

Если считыватель не получил ответ от метки, он может либо повторно отправить последнюю команду, либо перейти к следующему раунду инвентаризации. Благодаря этому возникает вопрос о предпочтительной стратегии (или сценарии) поведения меток: с повторной передачей или без. Более того, поскольку данные с датчика можно считывать как с помощью одной команды Read, так и с помощью отправки последовательности команд для получения только частей (фрагментов) данных, кажется важным сравнить между собой и эти возможные стратегии. Таким образом кажется последовательным рассмотреть следующие 3 сценария поведения меток:

1. Всякий раз, когда ответ метки получен с ошибкой, считыватель начинает новый раунд.

2. После любой ошибки считыватель повторно отправляет последнюю команду до тех пор, пока не получит ответ или метка не отключится (БПЛА улетел от метки). Для чтения данных считыватель отправляет одну команду Read.

3. Аналогично предыдущему варианту, но считыватель получает данные от датчика с помощью последовательности команд Read, считывая данные блоками по 2 слова (т.е. 32 бита).

Рисунок 3 демонстрирует типичный раунд инвентаризации и обработку ошибок в рамках этих стратегий. Первая стратегия может быть использована для идентификации быстро движущихся меток (она была изучена в [5]).

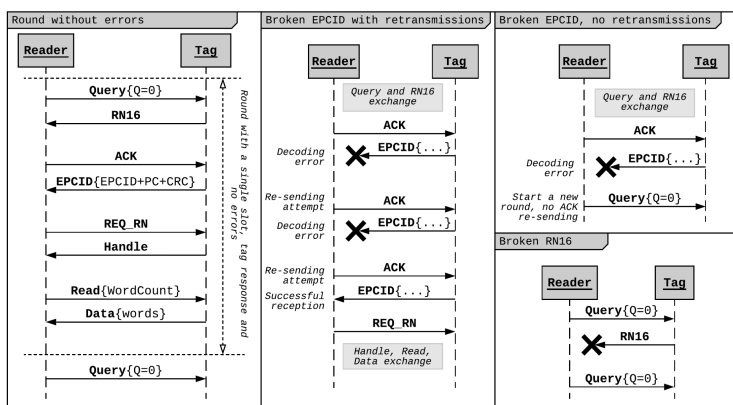


Рис. 3. Раунд инвентаризации (левый столбец); обработка ошибок с повторной отправкой последней команды (средний столбец) или без повторной отправки (вверху справа); ошибки при передаче RN16 обрабатываются одинаково во всех сценариях (внизу справа)

Рисунок 4 показывает разницу между второй и третьей стратегиями. Третья стратегия неудобна тем, что при каждом ответе на команду Read метка передает дополнительные 33 бита (1 бит – заголовок, 16 бит – текущий случайный дескриптор и 16 бит – контрольная сумма).

В качестве примера рассмотрим случай, когда считывателю нужно получить 10 слов. При второй стратегии, если не возникает ошибок, метка передает $10 * 16 + 33 = 193$ бита, а при третьей стратегии – $5 * (2 * 16 + 33) = 325$ бит.

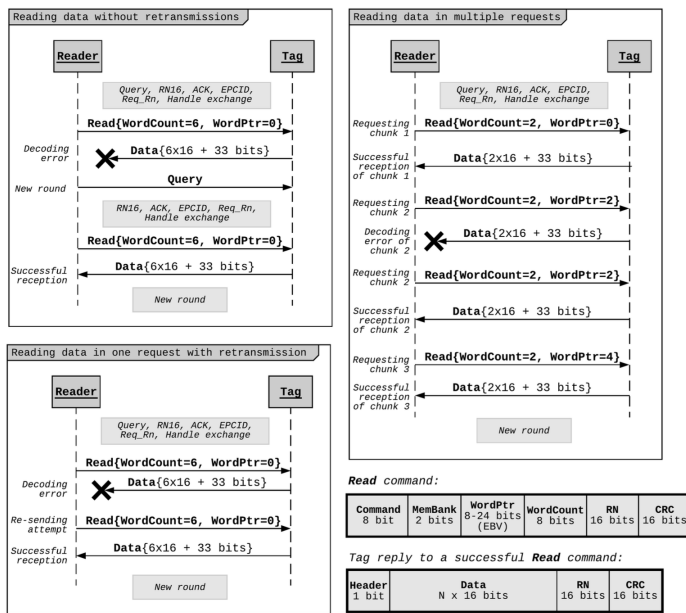


Рис. 4. Стратегии чтения данных: чтение без повторных передач (вверху слева), чтение с повторными передачами при ошибках и фрагментации данных (вверху справа), повторные передачи без фрагментации (внизу слева); Форматы команд чтения и ответов меток (внизу справа)

Пусть BER равен $p_{er} = 10^{-3}$, тогда передача одиночного ответа будет успешной с вероятностью $(1 - p_{er})^{193} \approx 0,824$ при использовании стратегии без фрагментации и $(1 - p_{er})^{65} \approx 0,937$ при использовании фрагментации. В случае одиночной ошибки метка передает 386 бит без фрагментации и 390 бит в противном случае. При большем количестве ошибок в передаче данных стратегия с фрагментацией начинает выигрывать с 455 бит против 579 бит без фрагментации. При этом здесь не учитывается время, необходимое для передачи дополнительных команд Read. Однако тенденция не меняется: чем больше данных требуется передать и чем хуже канал, тем больше времени нужно, если фрагментация не используется.

2.3. ЗАДЕРЖКИ ЧТЕНИЯ ДАННЫХ

Предположим, что считывателю необходимо получить N слов от сенсорной метки, и обозначим длину ответа как $|msg|$, а P_{msg} – вероятность того, что ответ «msg» будет успешно получен. Здесь предположено, что BER не меняется во время передачи ответа, поэтому $P_{msg} = (1 - p_{er})^{|msg|}$. Если используется фрагментация, то ответ с фрагментом данных будем называть «Chunk» (он содержит 2 слова). Если размер данных нечетный, то последний chunk, содержащий одно слово, будет рассмотрен и учтён отдельно. Задержки доступа к данным для всех определенных ранее стратегий могут быть описаны с помощью случайных полумарковских поглощающих процессов, показанных на рис. 5–7. Задержка доступа является случайной величиной, равной времени от начала до поглощения. Обозначим эти случайные величины как $\tau_1(N)$, $\tau_2(N)$ и $\tau_3(N)$ соответственно, где N , как определено выше, — это количество слов, которые нужно прочитать.

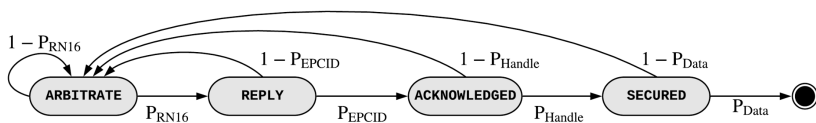


Рис. 5. Случайный процесс оценки задержки чтения данных без повторных передач

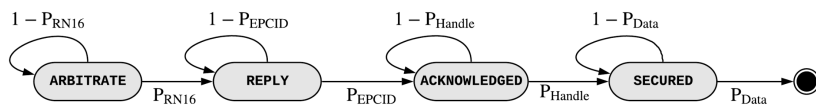


Рис. 6. Случайный процесс оценки задержки чтения данных с повторными передачами и без фрагментации

Поведение RFID-метки моделируется с помощью случайных процессов в соответствии с её рабочими режимами, описанными в стандарте EPC Class 1 Gen. 2 [16]. Состояние метки изменяется после успешной передачи ответа (см. рис. 3 для иллюстрации раунда инвентаризации).

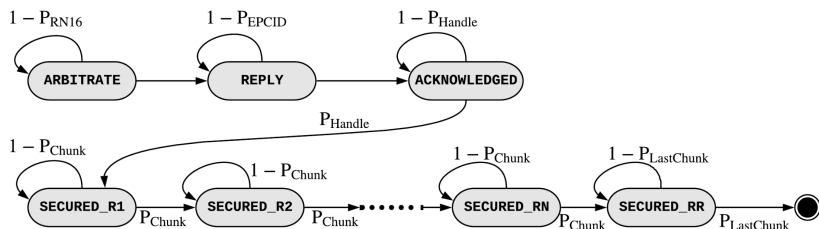


Рис. 7. Случайный процесс оценки задержки чтения данных с повторными передачами при ошибках и фрагментации данных

– Arbitrate: метка получает команду «Query» с $Q = 0$, генерирует случайное слово RN16 и передает его считывателю;

– Reply: метка получает команду Ack с правильным RN16 и передает обратно свой EPCID вместе со вспомогательными данными (PC, CRC);

– Acknowledged: после успешной отправки EPCID считыватель запрашивает новый случайный дескриптор через команду Req_RN, а метка возвращает его в ответе Handle. В отличие от первого ответа, он отправляет контрольную сумму вместе со сгенерированным дескриптором;

– Secured: метка получает команду «Read» и передает свои данные в виде заданного количества слов из банка USER;

– Secured_Ri, Secured_LC: эти состояния похожи на состояние Secured, но используются в моделировании третьего сценария для отслеживания текущего фрагмента (Secured_Ri). Если было запрошено нечетное количество слов, состояние Secured_LC указывает на то, что передается последний фрагмент с одним словом. В противном случае предполагается, что состояние Secured_LC имеет нулевую длительность, и переход в поглощающее состояние выполняется с вероятностью $P_{LastChunk} = 1$.

Предположим, что EPCID составляет 96 бит и считыватель запрашивает N слов из банка памяти USER (данные датчика). Форматы ответов меток определены в [16], а их длина в битах составляет: $L_{RN16} = 16$, $L_{EPC} = 128$ (EPCID, PC, CRC), $L_{Handle} = 32$ (RN, CRC), $L_{Data} = 16N + 33$, $L_{Chunk} = 65$ и $L_{LastChunk} = 49$.

Обозначим время, проведенное в состоянии S , как $T\{S\}$, время, необходимое для передачи команды или ответа «msg», – как T_{msg} , а задержку распространения – как δ . Согласно [16], обозначим время от передачи считывателя до ответа метки как T_1 , а интервал между последним фиктивным битом ответа метки и началом передачи считывателя как T_2 . Тогда время, проведенное процессом в состояниях, можно определить следующим образом:

$$\begin{aligned}
 T\{Arbitrate\} &= T_{Query} + T_1 + T_{RN16} + T_2 + 2\delta, \\
 T\{Reply\} &= T_{Ack} + T_1 + T_{EPC} + T_2 + 2\delta, \\
 (1) \quad T\{Acknowledged\} &= T_{Req_RN} + T_1 + T_{Handle} + T_2 + 2\delta, \\
 T\{Secured\} &= T_{Read} + T_1 + T_{Data} + T_2 + 2\delta, \\
 T\{Secured_Ri\} &= T_{Read} + T_1 + T_{Chunk} + T_2 + 2\delta, \\
 T\{Secured_LC\} &= T_{Read} + T_1 + T_{LastChunk} + T_2 + 2\delta.
 \end{aligned}$$

Длительность ответа можно найти как $T_{resp} = m/BLF(L_p + L_{resp} + 1)$, где L_p – длительность преамбулы; 1 добавлена для учета последнего фиктивного бита; m – порядок кода Миллера, а BLF – частота обратного рассеяния, $BLF = DR/TRcal$. Длительность команд оценить не просто: считыватель использует PIE (pulse interval encoding – кодирование длительностью импульса), поэтому нужно знать точное количество нулей и единиц, поскольку эти символы имеют разную длительность. Для оценки длительности команд в данной работе строятся эти команды, кодируется их содержимое и определяется количество нулей и единиц. Для упрощения вычислений некоторые поля заполняются равным количеством нулей и единиц, например поля контрольной суммы. Также предполагается, что T_1 имеет номинальное значение $T_1 = \max(RTcal, 10T_{pri})$ и T_2 имеет минимальное значение $T_2 = 3T_{pri}$.

3. Аналитический метод расчета времени передачи данных

Как было отмечено ранее, в работе рассмотрены три сценария передачи данных меткой: без пересылки данных в случае

ошибки, с пересылкой данных в случае ошибки и с пересылкой данных и их фрагментацией. Каждый сценарий передачи представляет собой случайный полумарковский процесс с единственным поглощающим состоянием $\xi = (n, t), n = \overline{1, N}, t \geq 0$, где n – номер состояния, а N – число состояний. В данном случае поглощающим состоянием является состояние с номером $n = N$. Попадание процесса в состояние с данным номером является завершением передачи данных меткой. Основной задачей является расчет математического ожидания времени до попадания процесса в поглощающее состояние при заданных вероятностях успешной передачи и среднем времени передачи на каждой фазе. Таким образом, аналитической моделью рассматриваемого процесса будет математическое ожидание времени, необходимого для попадания процесса в поглощающее состояние, как функция от времени каждой фазы и вероятностей успешного перехода между ними для всех трёх описанных ранее сценариев.

3.1. СЦЕНАРИЙ БЕЗ ПОВТОРНОЙ ПЕРЕДАЧИ

На рис. 8 показан граф переходов случайного процесса, описывающего передачу данных от метки к считывателю без повторной отправки. Процесс может находиться в пяти состояниях: первые четыре характеризуют отpravку данных, пятое – поглощающее состояние, характеризующее окончание передачи данных.

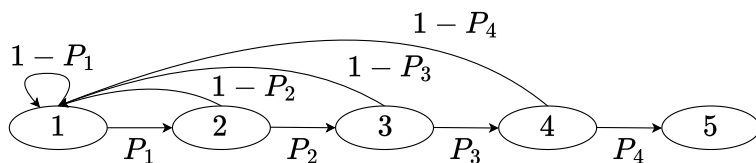


Рис. 8. Граф переходов случайного процесса, описывающего передачу данных метки без повторной передачи

Кратко опишем, каким образом происходят переходы. Начальным состоянием процесса является состояние 1. После чего через время T_1 процесс с вероятностью P_1 переходит в состояние 2 и с вероятностью $1 - P_1$ остается в состоянии 1, после

чего по истечении времени T_1 повторяется процедура перехода. Аналогично устроены переходы из состояний $i = 2, 3, 4$: после ожидания времени T_i процесс с вероятностью P_i переходит в состояние $i + 1$ и с вероятностью $1 - P_i$ возвращается в состояние 1.

Траекторией процесса является последовательность состояний, в которых был процесс до перехода в поглощающее состояние. Например, траекторией является $(1, 2, 3, 4)$, когда процесс последовательно прошел все состояния без возврата в состояние 1. Также примером траектории является $(1, 2, 3, 1, 1, 2, 3, 4)$, где процесс дважды вернулся в состояние 1: первый раз из состояния 3 и второй раз – из состояния 1. Также любой траектории можно сопоставить элемент вида (n_1, n_2, n_3, n_4) , где $n_i \geq 0, i = \overline{1, 4}$, – число возвратов процесса из состояния i в состояние 1. Можно заметить, что одному элементу (n_1, n_2, n_3, n_4) соответствует $\frac{(n_1+n_2+n_3+n_4)!}{n_1!n_2!n_3!n_4!}$ различных неповторяющихся траекторий. Обозначим множество $(n_1, n_2, n_3, n_4), n_i \geq 0, i = \overline{1, 4}$, как Ω .

Среднее время до перехода в поглощающее состояние:

$$(2) \quad T = \sum_{\omega \in \Omega} p_{\omega} T_{\omega},$$

т.е. равно сумме вероятности реализации траектории, умноженной на время до поглощающего состояния. Или, что то же самое:

$$(3) \quad T = \sum_{\omega \in \Omega} p_{\omega} (s_1 T_1 + s_2 T_2 + s_3 T_3 + s_4 T_4),$$

где s_i – число пребываний процесса в состоянии i для траектории ω , $s_i \geq 1, i = 1, 2, 3, 4$, для любой траектории $\omega \in \Omega$.

Для любой траектории число нахождений в состоянии i можно представить в виде суммы $s_i = m_i + n_i$, где m_i – число переходов в $(i + 1)$ -е состояние, а n_i – число переходов в состояние 1. Тогда между числами, характеризующими любую траекторию, можно найти следующие закономерности:

$$(4) \quad \begin{aligned} s_1 &= \sum_{i=1}^4 n_i + 1, & s_2 &= \sum_{i=2}^4 n_i + 1, \\ s_3 &= \sum_{i=3}^4 n_i + 1, & s_4 &= n_4 + 1; \end{aligned}$$

прибавление единицы к каждому значению s_i означает удачный переход в поглощающее состояние. Таким образом, уравнение (3) можно переписать в виде:

$$(5) \quad T = \sum_{\omega \in \Omega} p_{\omega} \left[\left(\sum_{i=1}^4 n_i + 1 \right) T_1 + \left(\sum_{i=2}^4 n_i + 1 \right) T_2 + \right. \\ \left. + \left(\sum_{i=3}^4 n_i + 1 \right) T_3 + (n_4 + 1) T_4 \right],$$

а время до поглощения для траектории ω равно соответственно:

$$(6) \quad T_{\omega} = T_{(n_1, n_2, n_3, n_4)} = \left(\sum_{i=1}^4 n_i + 1 \right) T_1 + \\ + \left(\sum_{i=2}^4 n_i + 1 \right) T_2 + \left(\sum_{i=3}^4 n_i + 1 \right) T_3 + (n_4 + 1) T_4.$$

Вероятность p_{ω} реализации траектории определяется как

$$(7) \quad p_{\omega} = \prod_{i=1}^4 p_i^{m_i} (1 - p_i)^{n_i},$$

или, с учетом уравнения (4):

$$(8) \quad p_{\omega} = p_{(n_1, n_2, n_3, n_4)} = \\ = p_1^{n_2 + n_3 + n_4 + 1} (1 - p_1)^{n_1} \cdot p_2^{n_3 + n_4 + 1} (1 - p_2)^{n_2} \cdot \\ \cdot p_3^{n_4 + 1} (1 - p_3)^{n_3} \cdot p_4 (1 - p_4)^{n_4}.$$

При переходе от суммирования по множеству всевозможных траекторий к суммированию по $n_i, i = \overline{1, 4}$, стоит учесть, что одному элементу суммы (n_1, n_2, n_3, n_4) соответствует $C_{(n_1, n_2, n_3, n_4)} = \frac{(n_1 + n_2 + n_3 + n_4)!}{n_1! n_2! n_3! n_4!}$ различных траекторий. Поэтому при переходе к сумме по $n_i, i = \overline{1, 4}$, получаем окончательную формулу для вычисления математического ожидания времени перехода в поглощающее состояние:

$$(9) \quad T(p_i, T_i) = \sum_{n_1, n_2, n_3, n_4} C_{(n_1, n_2, n_3, n_4)} p_{(n_1, n_2, n_3, n_4)} T_{(n_1, n_2, n_3, n_4)}.$$

Поскольку $n_i \geq 0$, а верхней границы у них, вообще говоря, нет, сумма в формуле (9) является бесконечной. Тем не менее, поскольку $p_i \leq 1$, $p_{(n_1, n_2, n_3, n_4)}$, будучи произведением показательных функций от n_i очень быстро убывает, а потому ряд в правой

части (9) является сходящимся. Подставив в него конкретные значения p_i и T_i и просуммировав получившийся ряд, можно получить математическое ожидание времени перехода в поглощающее состояние.

3.2. СЦЕНАРИЙ С ПОВТОРНОЙ ПЕРЕДАЧЕЙ ДАННЫХ

Как было отмечено ранее, возможна модификация чтения метки, при которой повторная передача осуществляется с момента неправильно переданного фрагмента. С точки зрения переходов случайного процесса это означает, что при неудачной отправке данных переход из состояния $i, i = \overline{1, 4}$, процесс остается в этом же состоянии (см. рис. 9).

Обозначим $p(n_1, n_2, n_3, n_4)$ – вероятность того, что процесс находился n_i раз в i -м состоянии; она вычисляется по следующей формуле:

$$(10) \quad p(n_1, n_2, n_3, n_4) = \prod_{i=1}^4 (1 - p_i)^{n_i} p_i.$$

Тогда математическое ожидание времени перехода в поглощающее состояние можно посчитать как

$$\begin{aligned} (11) \quad T &= \sum_{n_1, n_2, n_3, n_4} p(n_1, n_2, n_3, n_4) \sum_{i=1}^4 (n_i + 1) T_i = \\ &= \sum_{n_1, n_2, n_3, n_4} \prod_{i=1}^4 (1 - p_i)^{n_i} p_i \sum_{i=1}^4 (n_i + 1) T_i = \\ &= \sum_{n_1=0}^{\infty} (1 - p_1)^{n_1} p_1 (n_1 + 1) + \sum_{n_2=0}^{\infty} (1 - p_2)^{n_2} p_2 (n_2 + 1) + \\ &+ \sum_{n_3=0}^{\infty} (1 - p_3)^{n_3} p_3 (n_3 + 1) + \sum_{n_4=0}^{\infty} (1 - p_4)^{n_4} p_4 (n_4 + 1) = \\ &= \overline{T_1} + \overline{T_2} + \overline{T_3} + \overline{T_4}, \end{aligned}$$

т.е. математическое ожидание времени перехода в поглощающее состояние равно сумме математических ожиданий времен нахождения в непоглощающих состояниях.

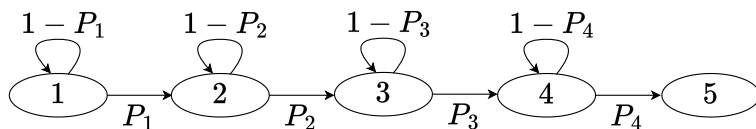


Рис. 9. Граф переходов случайного процесса, описывающего передачу данных с повторной передачей

3.3. СЦЕНАРИЙ С ПОВТОРНОЙ ПЕРЕДАЧЕЙ И ФРАГМЕНТАЦИЕЙ ДАННЫХ

Математическое ожидание времени перехода в поглощающее состояние в случайном процессе, который характеризует передачу с разбиванием данных на части (рис. 10), вычисляется таким же образом, как и для случая с повторной отправкой данных, но с учетом всех состояний, где происходит передача данных. Формула для вычисления имеет следующий вид:

$$(12) \quad T = \overline{T}_1 + \overline{T}_2 + \overline{T}_3 + M\overline{T}_c,$$

где M – число частей, на которые разбиваются данные, а T_c – время передачи части данных.

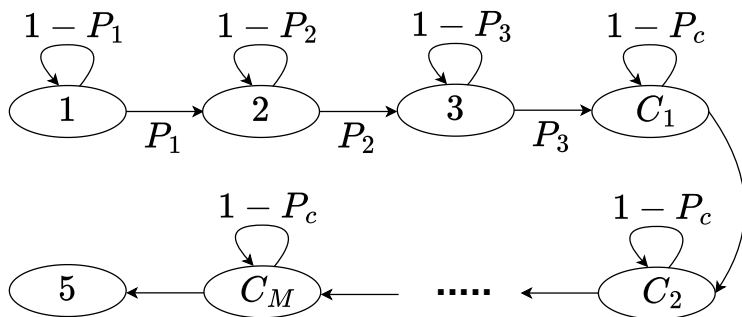


Рис. 10. Граф переходов случайного процесса, описывающего передачу данных с повторной передачей и фрагментацией данных

4. Результаты

4.1. ПРОИЗВОДИТЕЛЬНОСТЬ RFID СИСТЕМЫ

RFID-считыватель размещается на борту БПЛА и, как правило, использует тот же источник питания. Важно, чтобы считыватель мог надёжно взаимодействовать с сенсорными метками, не подлетая слишком близко к ним. Это особенно актуально при полёте над объектами, к которым невозможно приблизиться на малой высоте.

В отличие от традиционных беспроводных систем, RFID-считыватель должен постоянно излучать несущую волну, которая используется меткой для питания и обратного рассеивания своих данных. Поэтому длительность времени, когда считыватель включен, напрямую влияет на общее количество ресурсов, затрачиваемых на обмен данными. При значительных расстояниях между метками возможен режим, при котором считыватель активен только при пролёте над предполагаемой зоной обнаружения.

В этом разделе приводится оценка таких зон — высоты полёта БПЛА и расстояния до метки по земле — при которых возможно корректное считывание при заданном уровне BER в канале. Далее анализируется продолжительность обмена данными в зависимости от стратегии чтения и объёма считываемой информации. Предполагается, что БПЛА движется с постоянной скоростью. Все численные параметры приведены в таблице 1.

Таблица 1. Параметры, используемые в численном эксперименте для оценки производительности системы RFID

Параметр	Значение
Усиление антенны считывателя	7 дБм
Усиление антенны метки	2 дБм
Чувствительность метки	–22 дБм
Поляризационные потери	–3 дБ
Потери обратного рассеивания	–10 дБ
Шум на считывателе	–80 дБм

4.2. ОЦЕНКА ВРЕМЕНИ ЧТЕНИЯ

Задержки доступа к данным определяются как случайные величины $\tau_1(N)$, $\tau_2(N)$ и $\tau_3(N)$, описывающие время до поглощения в полумарковских случайных процессах ξ_1 (см. рис. 5), ξ_2 (рис. 6) и ξ_3 (рис. 7) соответственно. Численная оценка задержек доступа к данным в зависимости от BER для различных значений N дает результаты, показанные на рис. 11–14.

Для проверки аналитически полученных в предыдущей главе результатов были разработаны имитационные модели. Средняя погрешность между аналитическим расчетом и имитационным моделированием не превысила 0,5%.

Результаты для первого сценария показаны на рис. 11. Этот сценарий соответствует ситуации, когда в процессе обмена данными метка теряет сигнал от считывателя и отключается (так как не имеет собственного источника питания). Для второго сценария это же время существенно меньше, что видно из рис. 12. Это объясняется тем, что в случае ошибки отправки ответа метка не теряет сигнал от считывателя и остается в текущем состоянии. Благодаря этому она не тратит лишнее время на прохождение всего пути до поглощающего состояния с самого начала. Легко заметить, что в обоих случаях с ростом количества переданных слов увеличивается время, необходимое для считывания метки.

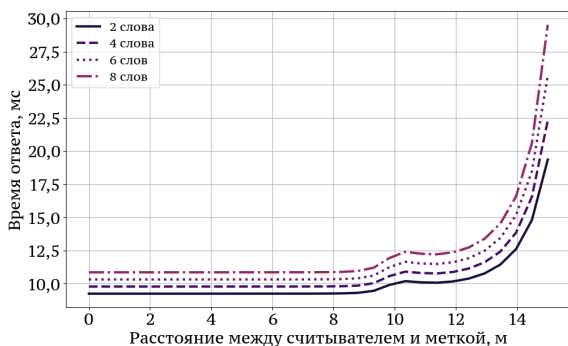


Рис. 11. Время чтения для первого сценария

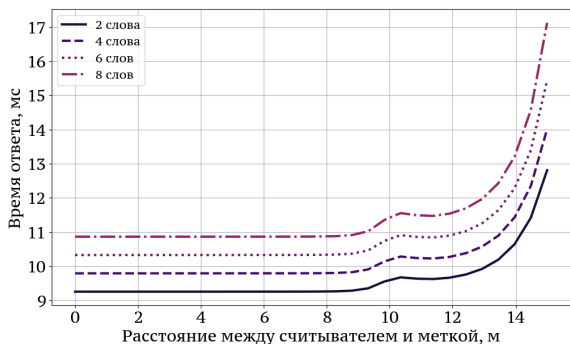


Рис. 12. Время чтения для второго сценария

В случае передачи большого объема данных (более 16 слов) время считывания метки может увеличиться на порядок (рис. 13). Это связано с тем, что при высоких значениях BER слишком длинный ответ метки, содержащий все необходимые считывателю данные, скорее всего, будет потерян. В связи с этим при значениях расстояния более 10 м (BER больше 0,004) выгоднее разбить большой пакет данных на несколько частей, что соответствует сценарию 3 (рис. 14). В этом случае потеря одного из фрагментов не приведет к серьезным потерям времени.

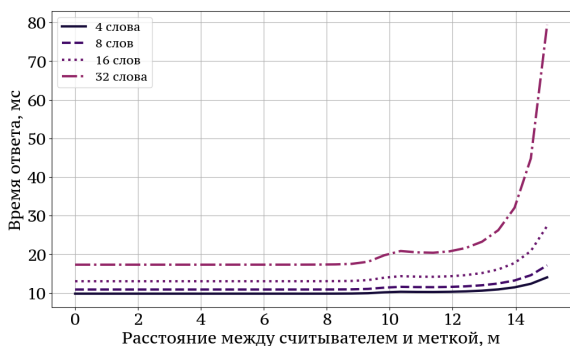


Рис. 13. Время чтения для второго сценария при чтении большого количества данных

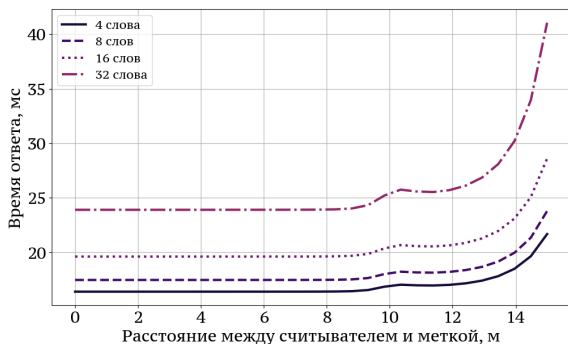


Рис. 14. Время чтения для третьего сценария при чтении большого количества данных

5. Оценка времени считывания меток при наличии интерференций

Данный раздел посвящен разработке и исследованию имитационной модели, учитывающей взаимодействие подвижного RFID-считывателя с метками в случае интерференций (коллизий). Используемая модель построена по принципу дискретно-событийной имитации, в которой события инициируют переходы между состояниями системы. Событиями являются, к примеру: включение считывателя, генерация метки в зоне чтения, движение считывателя, создание транзакции и прочее. Модель реализована при помощи объектно-ориентированного программирования на языке Python.

В основе модели лежит взаимодействие между тремя объектами: считывателем, меткой и каналом. Каждый из этих объектов реализован как публичный класс (Reader, Tag и Channel соответственно). Класс Reader содержит в качестве внутренних параметров, в частности, мощность генерируемого сигнала, T_{ar_i} , высоту и скорость движения, и генерирует команды согласно протоколу EPC Gen2 (Query, QueryRep, Ack и прочие). Аналогично, класс Tag также характеризуется большим количеством параметров (положение на земле, усиление антенны, чувствительность

приёмника, возможность включения/выключения по питанию). Наконец, класс Channel отвечает за моделирование распространения сигнала (расчёт мощности, BER, SNR согласно работе [13]) с учётом расстояния и эффекта Доплера.

На каждом шаге для считывателя, перемещающегося с заранее заданной скоростью, пересчитываются относительные положения меток и определяется, какие из них находятся в зоне считывания, происходит перерасчёт BER и реализуется детальная логика EPC Gen2: просчитываются раунды инвентаризации, слоты внутри раундов, обработка коллизий, управление Q , учёт флагов сессий. При этом для меток поддерживается протоколирование состояний: для каждой метки создаётся Python-объект журнала, в котором хранится каждый факт чтения, время каждого чтения, количество раундов, в котором метка успела принять участие, принимаемая мощность во время обмена данными со считывателем.

В качестве результатов имитационная модель возвращает:

- вероятность успешной идентификации;
- вероятность чтения пользовательского банка памяти, основанную на доле меток, успешно считанных хотя бы один раз;
- среднее время идентификации, вычисляемое как среднее по временным промежуткам от включения метки до первого чтения;
- количество коллизий, число слотов без ответов.

Важной особенностью созданной модели является поддержка многопроцессорных расчётов. Это позволяет запускать несколько параллельных симуляций, что в разы сокращает время получения результатов.

На рис. 15 представлено среднее время, необходимого для чтения одной метки, в зависимости от высоты полёта БПЛА. Можно заметить, что в случае большого количества меток среднее время чтения одной метки значительно увеличивается (более чем в 5 раз для высоты полёта в 1 м). Однако с увеличением высоты полёта относительный вклад в среднее время чтения одной метки начинает уменьшаться.

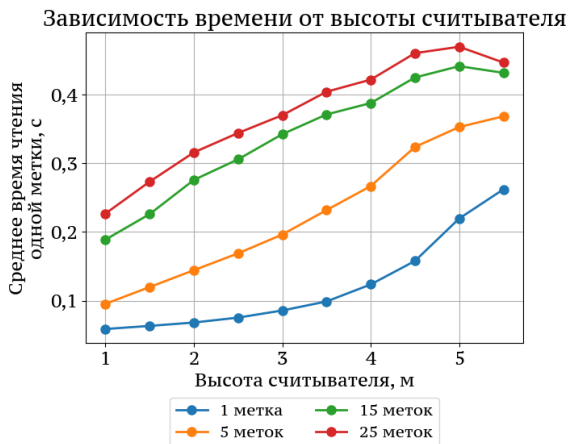


Рис. 15. Зависимость среднего времени, необходимого для чтения одной сенсорной метки, от высоты полёта в случае различного количества меток, находящихся одновременно в зоне чтения

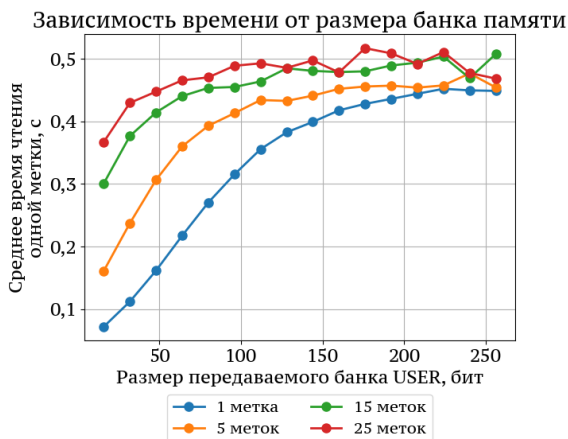


Рис. 16. Зависимость среднего времени, необходимого для чтения одной сенсорной метки, от размера передаваемого банка памяти в случае различного количества меток, находящихся одновременно в радиусе чтения

Аналогично уменьшение вклада от увеличения количества меток на среднее время чтения заметно и в случае увеличения среднего размера передаваемого банка памяти. Как видно из рис. 16, при увеличении объёма передаваемых данных коллизии перестают быть главным фактором, влияющим на среднее время чтения, из-за чего разница между всеми четырьмя линиями существенно сокращается, когда размер передаваемого банка превышает 200 бит.

Таким образом, можно сделать вывод, что приближение отсутствия коллизий является близким к реальности даже в случае наличия в зоне чтения считывателя более чем одной метки, при большой высоте полёта и большом объёме передаваемых данных.

6. Заключение

В данной статье исследуется время считывания RFID-меток с помощью считывателя, установленного на БПЛА. Для оценки состояния канала использовалась модель с белым гауссовым шумом. Были рассмотрены три сценария передачи данных: без повторной передачи фрагментов, с повторной передачей, а также с применением повторной передачи и фрагментацией. Для каждого из них разработаны соответствующие математические и имитационные модели.

Особое внимание уделено двум характерным случаям размещения меток. В первом случае метки расположены на значительном расстоянии друг от друга, благодаря чему в каждом раунде инвентаризации участвует не более одной метки и вероятность коллизий отсутствует. Для этого случая представлена аналитическая модель, показывающая зависимость времени идентификации от расстояния между считывателем и меткой.

Во втором случае метки размещены более плотно, что приводит к коллизиям в канале связи. Для анализа такой ситуации разработана дискретно-событийная имитационная модель, учитывающая высоту полёта, скорость движения БПЛА, параметры протокола RFID и другие характеристики системы. Проведённый анализ позволил оценить влияние плотности размещения меток

на время идентификации.

Литература

1. CHOI H.W., KIM H.J., KIM S.K. *An Overview of Drone Applications in the Construction Industry* // Drones. – 2023. – Vol. 7, No. 8. – Art. No. 515.
2. FINKENZELLER K. *RFID Handbook*. – New York: John Wiley and Sons, 2003.
3. GOPE P., MILLWOOD O., SAXENA N. *A Provably Secure Authentication Scheme for RFID-Enabled UAV Applications* // Computer Communications. – 2021. – Vol. 166. – P. 19–25.
4. GORTSCHACHER L.J., GROSINGER J. *UHF RFID Sensor System Using Tag Signal Patterns: Prototype System* // IEEE Antennas and Wireless Propagation Letters. – 2019. – Vol. 18, No. 10. – P. 2209–2213.
5. LARIONOV A.A., IVANOV R.E., VISHNEVSKY V.M. *UHF RFID in Automatic Vehicle Identification: Analysis and Simulation* // IEEE J. Radio Freq. Identif. – 2017. – Vol. 1, Iss. 1. – P. 3–12. – DOI: 10.1109/JRFID.2017.2751592.
6. LAZARO A., GIRBAU D., VILARINO R. *Effects of interferences in UHF RFID systems* // Progress In Electromagnetics Research. – 2009. – Vol. 98. – P. 425–443.
7. LI C. et al. *ReLoc 2.0: UHF-RFID Relative Localization for Drone-Based Inventory Management* // IEEE Trans. on Instrumentation and Measurement. – 2021. – Vol. 70. – P. 1–13. – Art. No. 8003313. – DOI: 10.1109/TIM.2021.3069377.
8. LONGHI M., MARROCCO G. *Ubiquitous Flying Sensor Antennas: Radiofrequency Identification Meets Micro Drones* // IEEE Journal of Radio Frequency Identification. – 2017. – Vol. 1, No. 4. – P. 291–299.
9. LONGHI M., CASATI G., LATINI D. *RFIDrone: Preliminary Experiments and Electromagnetic Models* // Proc. of the 2016 URSI Int. Symposium on Electromagnetic Theory (EMTS-2016). – 2016. – P. 450–453.

10. LUBNA et al. *IoT-Enabled Vacant Parking Slot Detection System Using Inkjet-Printed RFID Tags* // IEEE Sensors Journal. – 1 April 2023. – Vol. 23, No. 7. – P. 7828–7835.
11. MA Y. et al. *RAPP: A Radio Tomography Localization Method Characterized by Performance Parameterization in Rapid-Moving RFID System* // IEEE Trans. on Vehicular Technology. – 2023. – Vol. 72, No. 1. – P. 1265–1278.
12. MOHSAN S.A.H., OTHMAN N.Q.H., LI Y. et al. *Unmanned Aerial Vehicles (UAVs): Practical Aspects, Applications, Open Challenges, Security Issues, and Future Trends* // Intelligent Service Robotics. – 2023. – Vol. 16, No. 1. – P. 109–137.
13. NIKITIN P.V., RAO K.V.S. *Performance limitations of passive UHF RFID systems* // IEEE Antennas and Propagation Symposium. – 2006. – P. 1011–1014.
14. QUINO J., MAJA J.M., ROBBINS J. et al. *RFID and Drones: The Next Generation of Plant Inventory* // Agri. Engineering. – 2021. – Vol. 3, No. 2. – P. 168–181.
15. QUINO J., MAJA J.M., ROBBINS J. et al. *The Relationship between Drone Speed and the Number of Flights in RFID Tag Reading for Plant Inventory* // Drones. – 2022. – Vol. 6, No. 1. – DOI: 10.3390/drones6010002.
16. *EPCTM Radio-Frequency Identity Protocols Generation-2 UHF RFID Standard. Specification for RFID Air Interface Protocol for Communications at 860 MHz – 960 MHz, rel. 2.1.* – Wellington: EPCGlobal, 2015.
17. SHARMA D.K., MAHTO R.V., HARPER CH. et al. *Role of RFID Technologies in Transportation Projects: A Review* // Int. Journal of Technology Intelligence and Planning. – 2020. – Vol. 12, No. 4. – P. 349–377.
18. TAJIN M.A.S., JACOVIC M., DION G. et al. *UHF RFID Channel Emulation Testbed for Wireless IoT Systems* // IEEE Access. – 2021. – Vol. 9. – P. 68523–68534. – DOI: 10.1109/ACCESS.2021.3077845.
19. XU H., YIN X., ZHU F. et al. *RF-Ray: Sensing Objects in the Package via RFID Systems* // IEEE Systems Journal. –

- March 2023. – Vol. 17, No. 1. – P. 558–568. – DOI: 10.1109/JSYST.2022.3196462.
20. XUE C., LI T., LI Y. et al. *Radio-Frequency Identification for Drones With Nonstandard Waveforms Using Deep Learning* // IEEE Transactions on Instrumentation and Measurement. – 2023. – Vol. 72. – P. 1–13. – Art. No. 5503713. – DOI: 10.1109/TIM.2023.3306822.

IDENTIFICATION OF RFID TAGS FROM A SENSOR FIELD USING UAVS

Vilmen Abramian, V.A. Trapeznikov Institute of Control Sciences of RAS, Moscow, Junior Researcher (abramian.vl@phystech.edu).

Abstract: Radio Frequency Identification (RFID) technology is widely used in scientific and engineering applications, including integration with Unmanned Aerial Vehicles (UAVs) for object identification in hard-to-reach environments. Two key performance characteristics of such systems are the probability of successful tag reading and the reading time, both of which can be quantitatively described using semi-Markov process models. This paper considers two fundamentally different scenarios. In the first scenario, tags are sparsely distributed, which eliminates collisions. An analytical model is proposed to describe the interaction between the reader and a single tag. In the second scenario, densely placed tags lead to collisions. To evaluate system performance in this case, a discrete-event simulation model is developed. The model takes into account the specifics of the EPC Gen2 protocol, radio channel parameters, spatial configuration of tags, and data reading strategies. The paper compares analytical and simulation results and investigates how tag density, data volume, and UAV altitude affect reading performance metrics.

Keywords: RFID, UAV, sensor tag, simulation, analytics.

УДК 001.57

ББК 30ф

Статья представлена к публикации
членом редакционной коллегии А.А. Печниковым.

Поступила в редакцию 25.03.2025.

Дата опубликования 30.09.2025.