

Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

ПРОТОКОЛЫ МНОЖЕСТВЕННОГО ДОСТУПА НА ОСНОВЕ ГРАФОВЫХ КОДОВ ДЛЯ ПРИМЕНЕНИЯ В СЕТЯХ СВЯЗИ АВТОНОМНЫХ ТРАНСПОРТНЫХ СРЕДСТВ

Автор: Галкин Егор Георгиевич _____

Направление подготовки: 01.03.02 Прикладная
математика и информатика

Квалификация: Бакалавр

Руководитель ВКР: Бочарова И.Е., доцент., к.т.н. _____

Санкт-Петербург, 2020 г.

Обучающийся Галкин Егор Георгиевич

Группа М3435 Факультет ИТиП

Направленность (профиль), специализация

Математические модели и алгоритмы в разработке программного обеспечения

ВКР принята « ____ » _____ 20 ____ г.

Оригинальность ВКР ____ %

ВКР выполнена с оценкой _____

Дата защиты «» г.

Секретарь ГЭК Павлова О.Н. _____

Листов хранения _____

Демонстрационных материалов/Чертежей хранения _____

Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

УТВЕРЖДАЮ

Руководитель ОП

проф., д.т.н. Парфенов В.Г. _____

« ____ » _____ 20 ____ г.

ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ

Обучающийся Галкин Егор Георгиевич

Группа М3435 **Факультет** ИТиП

Квалификация: Бакалавр

Направление подготовки (специальность): 01.03.02 Прикладная математика и информатика

Направленность (профиль): Математические модели и алгоритмы в разработке программного обеспечения

Тема ВКР: Протоколы множественного доступа на основе графовых кодов для применения в сетях связи автономных транспортных средств

Руководитель Бочарова И.Е., доцент., к.т.н., доцент

2 Срок сдачи студентом законченной работы до: «» г.

3 Техническое задание и исходные данные к работе

Разработать протокол связи автономных транспортных средств с использованием графовых кодов для формирования протокольных последовательностей. Для восстановления потерь пакетов, возникающих в результате коллизий, использовать корректирующие коды.

4 Содержание выпускной квалификационной работы (перечень подлежащих разработке вопросов)

- а) Описание предметной области. Терминов. Постановка задачи. Обзор существующих решений.
- б) Выбор корректирующего кода, его тестирование.
- в) Генерация протокольных последовательностей с использованием графовых кодов.
- г) Оценка вероятности потери пакета, моделирование протокола.

5 Перечень графического материала (с указанием обязательного материала)

Графические материалы и чертежи работой не предусмотрены

6 Исходные материалы и пособия

- а) *Low Delay Inter-Packet Coding in Vehicular Networks / I. Bocharova [u dp.]// Future Internet. — 2019. — Т. 11, No 10. — С. 212.*

7 Дата выдачи задания «» г.

Руководитель ВКР _____

Задание принял к исполнению _____ «» г.

Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

АННОТАЦИЯ
ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ

Обучающийся: Галкин Егор Георгиевич

Наименование темы ВКР: Протоколы множественного доступа на основе графовых кодов для применения в сетях связи автономных транспортных средств

Наименование организации, где выполнена ВКР: Университет ИТМО

ХАРАКТЕРИСТИКА ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ

1 Цель исследования: Применение в сетях V2V графовых кодов для формирования протокольных последовательностей. Выбор неортогональных последовательностей приводит к коллизиям, поэтому предлагается кодировать последовательность пакетов каждого пользователя избыточным кодом. Это позволит восстановить пакеты, потерянные в результате коллизии.

2 Задачи, решаемые в ВКР:

- а) Реализация высокоскоростного сверточного кода с хорошей исправляющей способностью;
- б) Генерация протокольных последовательностей и разрешение возникающих конфликтов с помощью корректирующего кода;
- в) Оценка эффективности полученной системы;

3 Число источников, использованных при составлении обзора: 7

4 Полное число источников, использованных в работе: 8

5 В том числе источников по годам:

Отечественных			Иностранных		
Последние 5 лет	От 5 до 10 лет	Более 10 лет	Последние 5 лет	От 5 до 10 лет	Более 10 лет
1	0	1	1	2	3

6 Использование информационных ресурсов Internet: нет

7 Использование современных пакетов компьютерных программ и технологий:

Пакеты компьютерных программ и технологий	Раздел работы
MATLAB	3,4

8 Краткая характеристика полученных результатов

Разработана и смоделирована модель системы связи между автономными транспортными средствами, использующая корректирующие коды для восстановления пакетов, потерянных в результате коллизий. Проанализированы ее характеристики.

9 Гранты, полученные при выполнении работы

При выполнении работы грантов получено не было.

10 Наличие публикаций и выступлений на конференциях по теме выпускной работы

При выполнении работы статей опубликовано не было.

Обучающийся Галкин Е.Г. _____

Руководитель Бочарова И.Е. _____

« _____ » _____ 20__ г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	6
1. Обзор сетей связи V2V и существующих протоколов	8
1.1. Сети V2V	8
1.2. Стратегии доступа в канал	9
1.3. Алгоритмы на основе CSMA	9
1.4. Алгоритмы на основе TDMA	10
1.5. Недостатки существующих подходов	11
1.6. Попытки применения кодов в сетях VANET	11
1.7. Постановка задачи	11
1.8. Актуальность работы	11
Выводы по главе 1	12
2. Корректирующие коды и их применение в сетях связи	13
2.1. Основные понятия и определения	13
2.2. Коды на основе графов и гиперграфов	14
2.3. Методы декодирования корректирующих кодов	16
Выводы по главе 2	17
3. Сверточные коды для восстановления потерянных пакетов в сетях V2V	18
3.1. Задача восстановления потерянных пакетов в сетях	18
3.2. Кодирование сверточным кодом	19
3.3. Оконное декодирование сверточного кода	21
3.4. Оценка корректирующей способности	22
3.5. Тестирование	22
Выводы по главе 3	24
4. Протокольные последовательности, согласованные с кодами для исправления стираний	25
4.1. Определение и использование	25
4.2. Протокол разрешения коллизий	25
4.3. Выбор протокольных последовательностей	27
4.4. Математические оценки на число активных пользователей в канале	31
4.5. Моделирование	32
Выводы по главе 4	33

ЗАКЛЮЧЕНИЕ	34
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	35

ВВЕДЕНИЕ

Безопасность дорожного движения является одной из самых важных проблем современных городов. Последние годы сети автономных транспортных средств вызывают все больший интерес. Основная цель их внедрения — это улучшение безопасности дорожного движения. Сеть транспортных средств, носящая название *vehicular ad hoc network (VANET)*, это вид децентрализованной сети, в которой машины коммуницируют между собой. Каждое транспортное средство, принимающее участие в такой сети, оснащено *onboard unit (OBU)*, а также модулем *GPS*, которые позволяют осуществлять взаимодействие с другими участниками сети. В этой работе рассматривается взаимодействие *vehicle-to-vehicle (V2V)*, то есть без участия базовой станции. Для организации общения между транспортными средствами используется *Medium Access Control (MAC)*. Протоколы, используемые в сетях *V2V*, заданы текущим стандартом *IEEE-802.11p*. Но так как они были унаследованы от стандарта *IEEE-802.11*, их использование в сетях *V2V* ограничено. В сети возникает конфликт и потеря пакетов, если несколько пользователей передают данные по каналу в один момент. Для связи *V2V* сейчас существуют две стратегии доступа в канал: *CSMA* (случайный множественный доступ с прослушиванием канала) и *STDMA* (множественный доступ с разделением по времени). Конфликтов в этих системах либо пытаются избежать разделением времени, либо используют обратную связь, то есть делают по запросу повторную передачу пакета. Однако, в системах связи автономных транспортных средств из-за быстрого старения информации большие задержки недопустимы. В настоящей работе рассматривается система с множественным доступом в канал и последующим восстановлением потерянных пакетов. Для обеспечения возможности восстановления пакетов используется корректирующий код. Пользователь передает пакеты в соответствии со случайно выбранной протокольной последовательностью, задающей поведение в канале. Протокольные последовательности допускают определенное число конфликтов между пользователями. В отличие от предыдущих работ [6], рассматриваются сверточные коды с малой задержкой и сложностью декодирования. Множество протокольных последовательностей строится на основе графового кода и анализируется соответствие его характеристик выбранному корректирующему коду. В первой

главе данной работы будет приведен обзор сетей связи V2V и существующих протоколов связи в них.

Во второй главе дан краткий обзор в области корректирующих кодов и их применения в сетях связи.

В третьей главе будет приведено описание сверточного кода, использующегося в работе, его свойства и результаты работы в двоичном канале со стираниями.

В четвертой главе будут рассмотрены различные протокольные последовательности, описан протокол взаимодействия на их основе, а так же взаимодействие с кодами для исправления стираний.

В заключении приводятся результаты моделирования, приведены оценки на число пользователей в полученной системе.

ГЛАВА 1. ОБЗОР СЕТЕЙ СВЯЗИ V2V И СУЩЕСТВУЮЩИХ ПРОТОКОЛОВ

В данной главе рассматривается введение в предметную область решаемой задачи, приведен обзор существующих подходов, указаны их недостатки, даны основные определения. Также в данной главе будет произведена постановка задачи оптимизации и обоснована актуальность данной работы.

1.1. Сети V2V

Безопасность дорожного движения, при использовании V2V коммуникаций это относительно новая и перспективная сфера в области интеллектуальных транспортных систем. Взаимодействие V2V подразумевает передачу географических данных, направление движения, информации об изменении трафика. Все эти данные передаются постоянно. Использование такого взаимодействия требует высокой надежности доставки пакетов. Основные требования к таким сетям это:

- а) Отсутствие Roadside Unit,
- б) Использование канала без обратной связи,
- в) Пользователи не могут договариваться друг с другом, но все могут декодировать информацию, передаваемую по каналу,
- г) Низкая задержка доставки,
- д) Высокая вероятность успешной доставки сообщения за заданное время,
- е) Любой пользователь с подходящим оборудованием должен иметь возможность стать участником сети.

Решения подходящие для V2V были введены в IEEE 802.11p. Требования, указанные выше, ограничивают возможность использования существующих протоколов взаимодействия в сетях V2V. К примеру, CSMA, использующийся в этом стандарте, не гарантирует доступ в канал за конечное время. Также существующие методы, не позволяют избежать коллизий и обеспечить всем пользователям доступ в канал. Они лишь стараются минимизировать число коллизий и предоставить доступ в канал всем желающим. В контексте ненагруженных дорог или судоходства, эти решения справляются хорошо, но в случае высоконагруженных дорог, они показывают результаты значительно хуже.

1.2. Стратегии доступа в канал

Почти все хорошо изученные стратегии случайного доступа в канал, использующиеся в взаимодействии device-to-device, подразумевают наличие базовой станции, контролирующей взаимодействие.

В V2V, обычно, взаимодействие происходит без участия базовой станции. Поэтому основными претендентами для использования являются децентрализованные алгоритмы случайного доступа CSMA и STDMA.

Другим важным качеством сетей V2V является ограничение на задержку доставки. Именно это требование, например, ограничивает возможность использовать алгоритм CSMA, так как он может приводить к неограниченной задержке. Также ограничение на задержку доставки не дает применять алгоритмы основанные на повторной доставке потерянных пакетов, поэтому только каналы без обратной связи могут применяться в децентрализованных V2V сетях.

1.3. Алгоритмы на основе CSMA

Как уже упоминалось ранее, в стандарте IEEE 802.11p в качестве стандартного MAC метода для V2V взаимодействия используется алгоритм CSMA. Приведем его описание, а так же причины, по которым он не подходит для V2V.

В стандартной реализации алгоритма, узел, желающий передать данные, сначала слушает канал, и если определенное время он был свободен, то пользователь передает данные. Если же узел видит, что канал занят, он ждёт какое-то количество времени прежде чем повторить повторную попытку доступа в канал. В такой модели может быть ситуация, когда у нескольких узлов совпал период прослушивания канал и после того, как они убедились, что он свободен, происходит передача пакетов от нескольких пользователей одновременно и случается коллизия. Более того, могут быть ситуации, когда у узла будет очень высокая задержка доступа в канал, которая притом никак неограничена. То есть, может возникнуть ситуация, когда узел вообще не сможет получить доступ в канал. Указанные ранее недостатки особенно заметны при высокой загруженности сети, когда вероятность встретить одну из таких ситуаций максимальна. В стандарте IEEE 802.11p недостатки этого метода предлагается решать, используя traffic smoothing. Но это решение используется в основном в централизованных сетях, что нельзя сказать про сети V2V. Также traffic

smoothing не решает проблему неограниченной задержки доступа в канал, а лишь уменьшает вероятность возникновения таковой.

Существуют также некоторые модификации этого метода. Одна из них, которая также применяется в стандартке IEEE 802.11p, называется CSMA/Collision Avoidance (CA). Основная идея этой модификации заключается в том, что коллизиям подвергаются не сами пакеты данных, а так называемые jam сигналы. Так же как в оригинальном протоколе узел слушает канал, но перед тем как совершить отправку, он посылает по нему jam сигнал. Если за время некоторого ожидания, ни один из других узлов не послал такой же сигнал, то пользователь передает свои пакеты, если же были посланы другие jam сигналы, наш узел останавливает передачу на отрезок времени случайной длины и затем повторяет попытку. Но такой подход обеспечивает избегание коллизий только для unicast пакетов, что неприменимо в сетях V2V.

Существуют так же и другие модификации, например на основе приоритизации пакетов, в которых чем больше у передаваемого пакета приоритет, тем меньший промежуток времени узлу нужно наблюдать свободный канал перед его отправкой. Такая модификация также включена в приведенный ранее стандарт.

1.4. Алгоритмы на основе TDMA

Алгоритмы на основе TDMA разбивают время доступа в канал на временные отрезки и присваивают каждому узлу некоторые из них, используя распределенные схемы планирования доступа. Такие алгоритмы гарантировано обеспечивают доступ в канал связи за конечное время.

Одной из разработок как раз предложенной для сетей VANET, является MAC протокол названный Vehicular MAC (VeMAC) [7]. В VeMAC временные отрезки разделены на 2 непересекающихся множества. В таком случае машинам, едущим в разных направлениях, присваиваются временные отрезки из разных множеств, это сделано, чтобы уменьшить число коллизий, вызванных движением узла. Но так же как и в базовом алгоритме, один временной отрезок может быть присвоен двум или более узлам и тогда будут происходить коллизии. Другим известным алгоритмом на основе TDMA является STDMA, который в отличие от VeMAC, нашел свое применение и используется в Автоматическая идентификационная система, применяемой в судоходстве.

1.5. Недостатки существующих подходов

Суммируя все вышесказанное, мы можем выделить основные недостатки существующих решений. Такие алгоритмы как: IEEE 802.11p CSMA [8], VeMAC [7], STDMA [3, 8], в большинстве своем не подходят по характеристикам сетям V2V. Они не гарантируют низкой задержки доставки пакета. Также их недостатком является потеря пакетов, участвующих в коллизиях, и если в других сетях эту проблему предлагается решать повторной отправкой потерянных пакетов, то в сетях V2V такое решение неприменимо.

1.6. Попытки применения кодов в сетях VANET

Старейшей исследовательской работой по схожей теме является [6]. В ней впервые было предложено применение кодов для восстановления потерянных при передаче данных. Главным недостатком этой работы является применением в ней очень длинных блоковых кодов, что приводило к большой задержке. Работа выступает хорошим подспорьем для исследования применения сверточных кодов в сетях V2V, так как канал в этих сетях можно трактовать, как канал с коллизиями без обратной связи.

Относительно, новой работой по смежной теме является [5]. В ней было рассмотрено применение сверточных кодов и протокольных последовательностей для взаимодействия vehicle-to-roadside (V2V). Результаты этой работы показали эффективность применения таких кодов и послужили стимулом к написанию этой работы.

1.7. Постановка задачи

В рамках этой работы необходимо реализовать модель системы связи между автономными транспортными средствами. Для этого в первую очередь нам необходимо выбрать сверточный код, с помощью которого возможно будет осуществить восстановление потерянных пакетов. Далее необходимо выбрать протокольные последовательности, на основе которых будет осуществляться доступ в канал, посчитать вероятности конфликтов для общего случая. После необходимо соединить эти решения и смоделировать выход в канал различным числом пользователей.

1.8. Актуальность работы

В данный момент существующие стандарты не предоставляют подходящих решений для V2V взаимодействия в сетях VANET. От протоколов требу-

ется маленькая задержка при передаче, а также высокая надежность. Проводятся разные исследования в этой области и одно из направлений это применение кодов для реализации протокола.

Выводы по главе 1

В данной главе было проведено введение в предметную область, рассмотрены существующие решения, приведены их недостатки. Также были рассмотрены работы и исследования в области. Дана постановка задачи и обоснование актуальности темы.

ГЛАВА 2. КОРРЕКТИРУЮЩИЕ КОДЫ И ИХ ПРИМЕНЕНИЕ В СЕТЯХ СВЯЗИ

В данной главе приводятся основные понятия из Теории кодирования, используемые при выполнении работы, даны определения методов декодирования.

2.1. Основные понятия и определения

В данном разделе будут приведены основные термины, которые неоднократно будут использоваться в тексте этой работы.

Линейное пространство $V(F)$ над полем F — это упорядоченная четверка $(V, F, +, \cdot)$, где

- а) V — непустое множество векторов;
- б) F — поле, элементы которого называются скалярами;
- в) Определена операция сложения векторов $V \times V \rightarrow V$.
- г) Определена операция умножения векторов на скаляры $F \times V \rightarrow V$;

Также заданные операции удовлетворяют аксиомам линейного пространства.

Линейный (n, k) -код — линейное подпространство C размерности k векторного пространства F_q^n , где F_q — конечное поле из q элементов.

Порождающая матрица кода — базис подпространства, образующего код. Обозначается как G .

Проверочная матрица кода — базис ортогонального к линейному коду пространства. Обозначается как H .

Вес Хэмминга последовательности — количество в ней ненулевых символов.

Минимальное расстояние линейного кода — минимальный вес Хэмминга ненулевых кодовых слов. Обозначается как d_{min} .

Синдром — синдромом для вектора y называется $s = yH^T$. То, что синдром не равен нулю, указывает на наличие ошибок в принятой последовательности y . Используется, например, при синдромном декодировании.

Блочный код — корректирующий код, который по входным блокам длины k порождает выходные блоки длины n . Кодеры блочных кодов не имеют памяти.

Сверточный код — в отличие от блочных кодов, в сверточных кодах блоки данных, по которым вычисляются в каждый момент времени кодовые

символы, перекрываются с предыдущими. Кодер содержит k двоичных регистров сдвига длин v_1, v_2, \dots, v_k . Входами регистров сдвига являются информационные символы. Выходы ячеек регистров соединены с многовходовыми сумматорами по модулю 2, таких сумматоров n , по числу выходных символов, формируемых кодером на каждом такте работы. [2]

Скорость кода — отношение k/n .

Длина кодового ограничения сверточного кода — суммарная длина регистров сверточного кода.

Память сверточного кода — максимальная длина регистров кода, обозначается как m .

Свободное расстояние сверточного кода — минимальное расстояние сверточного кода, обозначается как d_{free} .

Двоичный канал со стираниями — модель канала связи, использующаяся в теории кодирования. В данной модели отправитель передает бит по сети, а получатель с определенной вероятностью получает либо отправленный бит, либо сообщение о том, что бит не был получен ("стирание"). Модель данного канала изображена на Рис.1:

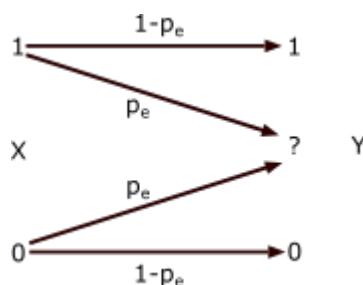


Рисунок 1 – Схема BEC

2.2. Коды на основе графов и гиперграфов

В данном разделе мы рассмотрим представление кодов в виде графов, введем некоторые сопутствующие термины, а так же введем определение кода с низкой плотностью проверок на четность.

Граф — граф G задается множеством вершин V и множеством ребер E , где каждое ребро соединяет в точности два узла..

Гиперграф — обобщение графа, в котором каждым ребром может соединяться не только две вершины, но и любые подмножества вершин.

t -дольный граф — пусть множество узлов V разделено на t непересекающихся подмножеств V_k , $k = 0, 1, \dots, t-1$. Такой граф называется t -дольным

если в нем не существует ребер соединяющих два узла из одного и того же множества V_k , $k = 0, 1, \dots, t - 1$.

Путь — путем из вершины v_i в вершину v_j называется чередующаяся последовательность вершин и ребер, ведущая от v_i к v_j , в которой никакое ребро не встречается дважды.

Цикл — циклом называется путь, в котором начальная и конечная вершины совпадают.

Длина цикла — число ребер в этом цикле.

Простой цикл — цикл называется простым, если он не проходит через одну вершину более одного раза.

Обхват — длина наименьшего цикла в графе.

Граф Таннера — графом Таннера линейного кода, заданного проверочной матрицей $H = \{h_{ij}\}$, $i = 1, \dots, r$, $j = 1, \dots, n$, называется двудольный граф, одно множество узлов которого соответствует множеству проверок (проверочные узлы), другое — множеству кодовых символов (символьные узлы) [2]. Символьный узел c_i связан ребром с некоторым проверочным узлом s_j , если i -й кодовый символ входит в j -ю проверку, иными словами если элемент h_{ij} не равен нулю. Пример графа Таннера изображен на Рис.2:

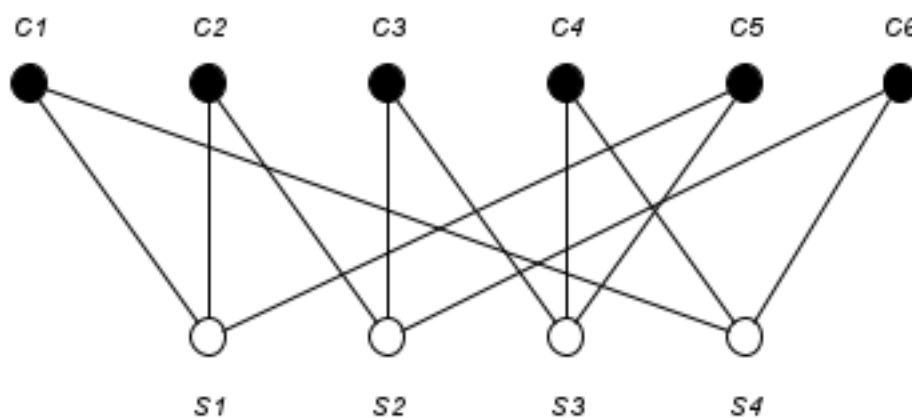


Рисунок 2 – Граф Таннера

Данный граф Таннера соответствует матрице:

$$H = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \end{pmatrix}$$

МППЧ-коды — класс линейных блочных кодов, особенностью которых является малое число единиц в проверочной матрице по сравнению с числом нулей. Главное преимущество таких кодов их производительность и линейная сложность декодирования. [2]

регулярный МППЧ-код — двоичный линейный (n, k) -код, заданный проверочной матрицей H , называется (J, K) -регулярным кодом с малой плотностью проверок на честность, если каждый столбец матрицы H содержит ровно J единиц, а каждая строка — K единиц. [4]

2.3. Методы декодирования корректирующих кодов

В данном разделе будут рассмотрены методы декодирования. А так же их модификации в используемом в данной работе Двоичном канале со стираниями.

Для начала рассмотрим два основных подхода к декодированию полученных сообщений: *Декодирование по максимуму апостериорной вероятности* и *Декодирование по максимуму правдоподобия*.

Введем необходимые для определения двух этих методов условия. Пусть у нас есть $\mathbf{x}, \mathbf{y} \in A = \{0, \dots, q-1\}^n$ — входная и выходная последовательности канала связи, заданного условными вероятностями. Предположим, что канал стационарен и является каналом без памяти, т.е.

$$\mathbf{p}(\mathbf{y}|\mathbf{x}) = \prod_{i=1}^n p(y_i|x_i), \quad (1)$$

и переходные вероятности $p(y_i = y|x_i = x)$ не зависят от i . Дадим тогда определение методам декодирования.

Декодированием по максимуму апостериорной вероятности кода $C = \{\mathbf{c}_m\}, m = 1, \dots, M$ над алфавитом A называют принятие решения в пользу слова с номером

$$\hat{m} = \arg \max_m \mathbf{p}(\mathbf{c}_m|\mathbf{y}). \quad (2)$$

Декодированием по максимуму правдоподобия кода $C = \{\mathbf{c}_m\}, m = 1, \dots, M$ над алфавитом A называют принятие решения в пользу слова с номером

$$\hat{m} = \arg \max_m \mathbf{p}(\mathbf{y}|\mathbf{c}_m). \quad (3)$$

Если максимум вероятности достигается для нескольких значений m , то производится случайный выбор решения из этого множества.

Также конкретные реализации этих методов зависят от используемого канала связи. В этой работе рассматривается канал со стираниями. В нем декодирование по максимуму правдоподобия может быть выполнено достаточно просто и сводится к решению системы линейных уравнений. В таком случае сложность декодирования будет $O(v^3)$, где v обозначает число стираний.

Выводы по главе 2

В данной главы были даны основные термины из Теории кодирования, которые будут использоваться в будущих главах. Были приведены методы декодирования, объявлен, используемый в работе быстрый метод декодирования.

ГЛАВА 3. СВЕРТОЧНЫЕ КОДЫ ДЛЯ ВОССТАНОВЛЕНИЯ ПОТЕРЯННЫХ ПАКЕТОВ В СЕТЯХ V2V

В данной главе приводится описание работы кодера и декодера в канале со стираниями. Указаны характеристики, используемых кодов. Даны оценки на максимальное число исправлений. Смоделированы зависимость числа потерянных данных от вероятности стирания.

3.1. Задача восстановления потерянных пакетов в сетях

Сперва стоит показать, что проблема восстановления потерянных пакетов в сетях эквивалентна проблеме декодирования в канале со стираниями. В работе предполагается, что единственным источником потери пакетов является их коллизия. Мы можем делать такое предположение, так как корректирующие коды физического уровня, с большой долей вероятности находят ошибки в передаваемых данных. Так же мы предполагаем, что пользователи сети, могут получать и декодировать все сообщения, передающиеся по ней. В таком случае, как и в ВЕС показанном на Рис.1, мы будем получать либо сами данные, либо информацию о потере пакетов, которую мы можем интерпретировать, как стирание, а следовательно и применить для восстановления пакетов методы декодирования.

Стоит также дать определение пакета, так как в контексте данной работы, да и основополагающей работе по передаче данных в канале с коллизиями без обратной связи [6], оно отличается от стандартного определения пакета в сети.

Пакет — в контексте этой работы, под пакетом подразумевается битовая последовательность, являющаяся частью сообщения, передаваемого пользователем. Размер пакета, зависит от характеристик используемого кода. В некоторых случаях мы можем разделять пакет на части размером до одного бита для передачи.

Из-за жестких требований по задержке и сложности, мы выполняем ”оконное декодирование”. Само по себе это достаточно широкое определение, в контексте этой задачи оно означает, что мы выполняем декодирование по максимуму правдоподобия некотором окне.

В нашем случае, для кодирования и декодирования мы будем использовать сверточный код со скоростью $(n-1)/n$. При декодировании по максимуму правдоподобия этот код позволит не только гарантированно исправить $d_{free} - 1$ стирание, но и позволит исправить значительно большее их количество, при

условии, что они будут достаточно сильно разнесены по кодовому слову. Поясним кодирование и декодирование на примере. Приведем порождающую и проверочную матрицу используемого кода.

$$H = \begin{pmatrix} 1 + D + D^2 & 1 + D^2 & 1 + D & 1 \end{pmatrix} \quad (4)$$

И, как мы знаем, из проверочной матрицы, заданной в систематическом виде: $H = (P^T I)$, легко получить порождающую матрицу, она будет иметь вид: $G = (IP)$ [2], тогда наша порождающая матрица будет иметь вид.

$$G = \begin{pmatrix} 1 & 0 & 0 & 1 + D + D^2 \\ 0 & 1 & 0 & 1 + D^2 \\ 0 & 0 & 1 & 1 + D \end{pmatrix} \quad (5)$$

Приведенный выше код обладает следующими характеристиками: скорость кода равна $R = 3/4$, кодовое ограничение равно 5, память $m = 2$, свободное расстояние $d_{free} = 3$.

3.2. Кодирование сверточным кодом

В данной работе было протестировано два подхода к кодированию: стандартный сверточный кодер и рекуррентный кодер. Рассмотрим оба из них.

Для стандартного подхода к построению кодера нам необходимо разложить нашу порождающую матрицу по степеням переменной D . В итоге мы получим такое представление.

$$G = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \times D + \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \times D^2 \quad (6)$$

Построим по полученному разложению схему кодера. Она приведена на Рис.3. Нам понадобится в сумме всего 5 ячеек памяти. Задержка кодирования у нас будет равна размеру блока на выходе, то есть 4.

По полученной схеме не составит труда реализовать кодер для заданного сверточного кода, что, собственно и было сделано.

В ходе работы также была попытка реализовать рекурсивный кодер, использующий проверочную матрицу кода. Опишем идею его работы. Преобра-

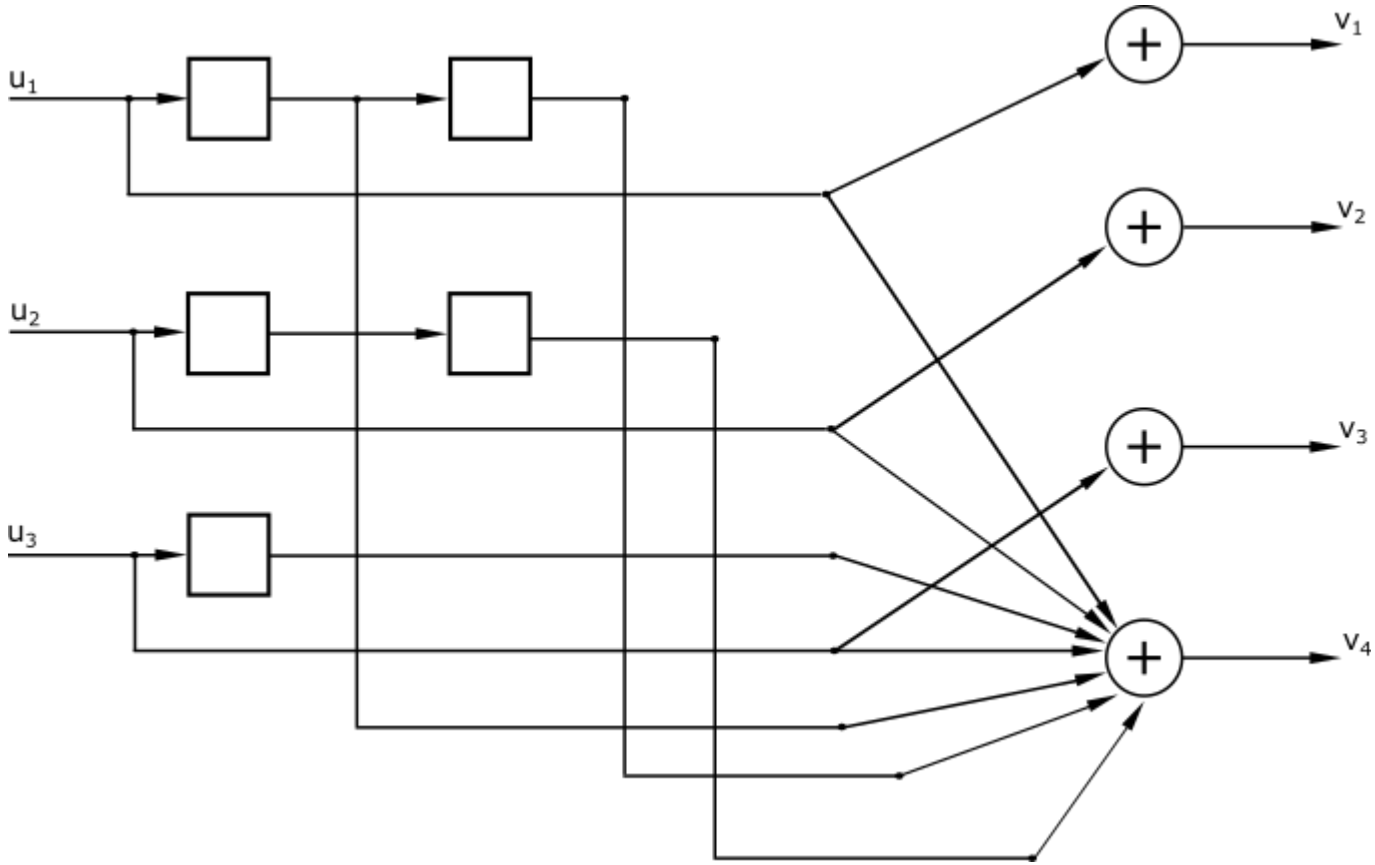


Рисунок 3 – Схема кодера

зовав нашу проверочную матрицу и переписав ее через разложение по степеням D , получим искомую подматрицу

$$H = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (7)$$

Тогда кодовое слово может быть записано, как $\mathbf{u}_1, v_1, \mathbf{u}_2, v_2, \dots$, где $\mathbf{u}_i = (u_{i1}, u_{i2}, u_{i3})$ — блок сообщения, состоящий из информационных символов и v_i — соответствующий проверочный символ. Обозначим как $s_i = (s_{i1}, s_{i2}, s_{i3}) = u_i * \tilde{H}^T$ — частичный синдром информационного блока. Тогда кодирование заданного сверточного кода можно реализовать по рекуррентной формуле.

$$\begin{aligned} v_1 &= s_{1,1}, v_2 = v_1 + s_{1,2} + s_{2,1}, \\ v_i &= v_{i-2} + v_{i-1} + s_{i-2,3} + s_{i-1,2} + s_{i,1}, i = 3, 4, \dots \end{aligned} \quad (8)$$

Данный метод был реализован, но в некоторых случаях, проявлялись различного рода неточности кодирования, поэтому было решено использовать стандартный кодировщик.

3.3. Оконное декодирование сверточного кода

Теперь нам необходимо реализовать декодирование сверточного кода. Как уже упоминалось ранее в этой работе, для канала со стираниями декодирование сводится к решению системы линейных уравнений [5]. Рассмотрим постановку задачи.

Рассмотрим ВЕС с вероятностью стирания равной ϵ . Пусть H — это проверочная матрица размера $r_B \times n_B$ двоичного $(n_B, n_B - r_B)$ блочного кода, с минимальным расстоянием d_{min} . Декодирование по максимуму правдоподобия исправляет любые комбинации v ошибок, если $v \leq d_{min} - 1$. Если же $d_{min} \leq v \leq r_B$, то однозначное декодирование не может быть гарантировано.

Пусть $\mathbf{y} = (y_1, y_2, \dots, y_{n_B})$ — это полученный вектор, где $y_i \in 0, 1, \phi$, и символ ϕ представляет стирание на позиции. Обозначим как $\mathbf{e} = (e_1, e_2, \dots, e_{n_B})$ двоичный вектор, значения которого в позиции i равно единице, если $y_i = \phi$ и нулю иначе. Пусть тогда $I(e)$ будет набором ненулевых позиций вектора e , $|I(e)| = v(e)$, и вектор $\mathbf{z} = (z_1, z_2, \dots, z_v)$ будет вектором неизвестных, которые нам и нужно найти.

Рассмотрим систему линейных уравнений $\mathbf{y}H_{I(e)}^T = \mathbf{0}$. Она может быть сведена к:

$$\mathbf{z}H_{I(e)}^T = s(e), \quad (9)$$

где $s(e)$ — это посчитанный по нестертым позициям синдром. Тогда декодирование по максимуму правдоподобия в ВЕС сводится к решению (9).

Используемый в работе алгоритм основан на механизме скользящего окна. На каждом шаге декодирования наш алгоритм использует матрицу размера $(W - m) \times W$ определяющую усеченный нулями (zero-tail terminated) сверточный код, где $W = L + m + 1$ — это ширина окна. На каждом шаге алгоритма мы пытаемся исправить стирания в текущем окне.

Для нашего кода эта матрица имеет вид:

$$H_{ZT} = \begin{pmatrix} 1100 & 1010 & 1111 & & \\ & 1100 & 1010 & 1111 & \\ & & 1100 & 1010 & 1111 \end{pmatrix} \quad (10)$$

После по этой матрице, был построен декодер с параметрами $W = 5$, $m = 2$, $L = 2$.

3.4. Оценка корректирующей способности

Приведенный код, может гарантированно исправлять $d_{free} - 1$ стирание. Для нашего кода $d_{free} = 3$. Так как мы использовали усечение кода, то у полученного в его результате блочного кода d_{min} могло быть меньше, чем у порождающего его сверточного кода. Однако при выборе размера окна $L \geq 2$, минимальное расстояние усеченного кода, равняется свободному расстоянию исходного сверточного, то есть при усечении мы не потеряли в корректирующей способности.

При декодировании сверточного кода по максимуму правдоподобия, мы можем исправить большое число образцов стираний с числом стираний больше либо равным d_{free} . Типично исправляются стирания достаточно разнесенные по длине кодового слова. При оконном декодировании, для каждого окна гарантируется d_{min} , но могут исправляться образцы стираний с числом стираний не превышающим число избыточных символов усеченного кода.

3.5. Тестирование

Протестируем класс наших кодов и посмотрим как в зависимости от вероятности стирания и размера окна, будет изменяться вероятность битовой ошибки для переданных данных. Сразу отметим, что для кодов с разной скоростью, диапазон допустимых вероятностей стираний отличается так, как пропускная способность канала со стираниями равна $1 - \epsilon$, где ϵ — вероятность стирания, и если $\epsilon \geq 1 - R$, то код будет работать со скоростью больше пропускной способности. Был смоделирован ВЕС. Приведем результаты для кода, рассмотренного выше в этой главе. :

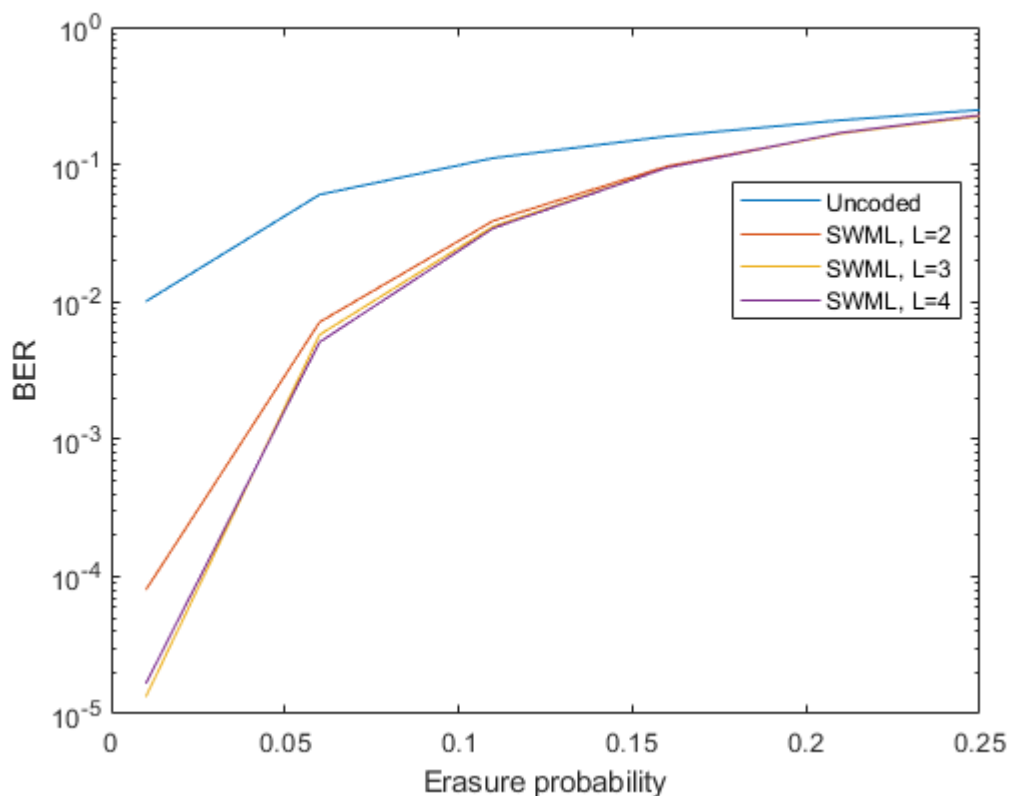


Рисунок 4 – Зависимость ошибки на бит от вероятности стираний, $R = 3/4$

Как следует из графиков, приведенных на Рис.4, кодирование дает нам значительный выигрыш, особенно на небольших вероятностях стирания. Посмотрим теперь результаты для других кодов этого класса. Рассмотрим очень высокоскоростной код с $R = 7/8$:

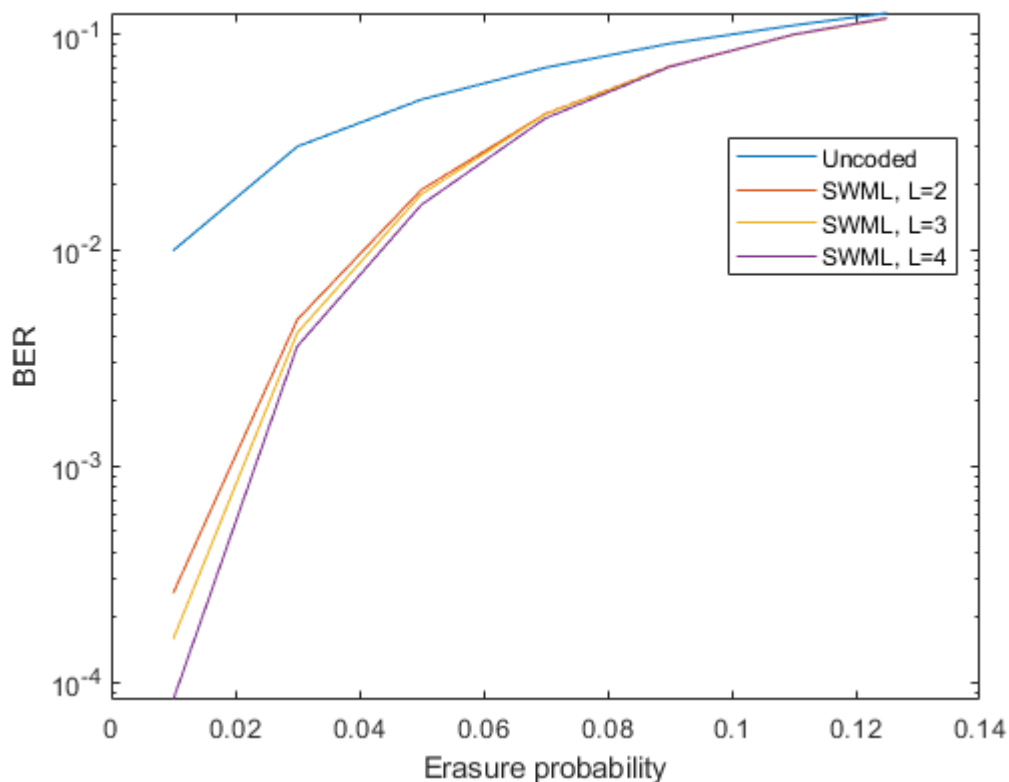


Рисунок 5 – Зависимость ошибки на бит от вероятности стираний, $R = 7/8$

Как следует из графиков, приведенных на Рис. 5, кодирование, так же как и для $(4,3)$, кода дает значительный выигрыш. Для этого кода, отчетливо видно, что лучшие результаты дает окно $L = 4$. Это обусловлено высокой скоростью кода, так при меньшем размере окна, при оконном декодировании мы получим меньше информации, а следовательно и хуже исправим стирания.

Выводы по главе 3

В этой главе был описан класс кодов, подходящий для использования в сетях V2V. Приведен пример построения кодера и декодера для этого класса кодов. Проведено тестирование кода в двоичном канале со стираниями, построена зависимость ошибки на бит от вероятности стираний для разных кодов и размеров окна.

ГЛАВА 4. ПРОТОКОЛЬНЫЕ ПОСЛЕДОВАТЕЛЬНОСТИ, СОГЛАСОВАННЫЕ С КОДАМИ ДЛЯ ИСПРАВЛЕНИЯ СТИРАНИЙ

В данной главе будет дано определение протокольной последовательности, объяснено почему для их построения были выбраны графовые коды. Проведено моделирование доступа в канал с использованием протокольных последовательностей. Дана оценка вероятности потери пакета в сети. Смоделировано совместное использование протокольных последовательностей и корректирующего кода.

4.1. Определение и использование

Протокольные последовательности определяют поведение пользователя в канале. При желании послать сообщение пользователь будет действовать в соответствии с выбранной им протокольной последовательностью. Поэтому важно, чтобы последовательности при наименьшей возможной своей длине обеспечивали малое заданное число коллизий согласованное с корректирующей способностью кода. Опишем протокол их использования.

Рассмотрим ситуацию в канале. M пользователей генерируют свои пакеты. В нашей системе пользователь посылает сообщение с заданной вероятностью $p_i, i = 1, \dots, M$, с вероятностью $(1 - p_i)$ пользователь "молчит". Если пользователь имеет сообщение для передачи, он случайно выбирает протокольную последовательность из списка доступных протокольных последовательностей.

После того, как выбрана протокольная последовательность пользователь передает пакеты на "единицах" этой последовательности и молчит на ее "нулях".

Протокольные последовательности можно выбирать ортогональными, чтобы точно избежать конфликтов, но таких последовательностей слишком мало. В настоящей работе рассматривается класс протокольных последовательностей с малым числом пересечений (коллизий), предназначенный для совместного использования с корректирующими кодами.

4.2. Протокол разрешения коллизий

Как уже отмечалось в предыдущем разделе, мы не можем выбрать большое число ортогональных протокольных последовательностей. Существующие протоколы в V2V сетях используют обратную связь для разрешения конфликтов. Но это требует наличия канала обратной связи и вносит большую за-

держку. Выберем протокольные последовательности, которые допускают коллизии, но при этом будем восстанавливать полученные из-за них стирания, с помощью нашего сверточного кода. Был выбран высокоскоростной код, так как он имеет малую алгоритмическую сложность декодирования, значит это не сильно повлияет на производительность системы. Также наш код, в случае длинных последовательностей, позволяет исправить большее образцов стираний с числом стираний больше либо равным d_{free} .

На Рис. 6 приведен пример доступа в канал двумя пользователями и возникновения в результате этого коллизии:

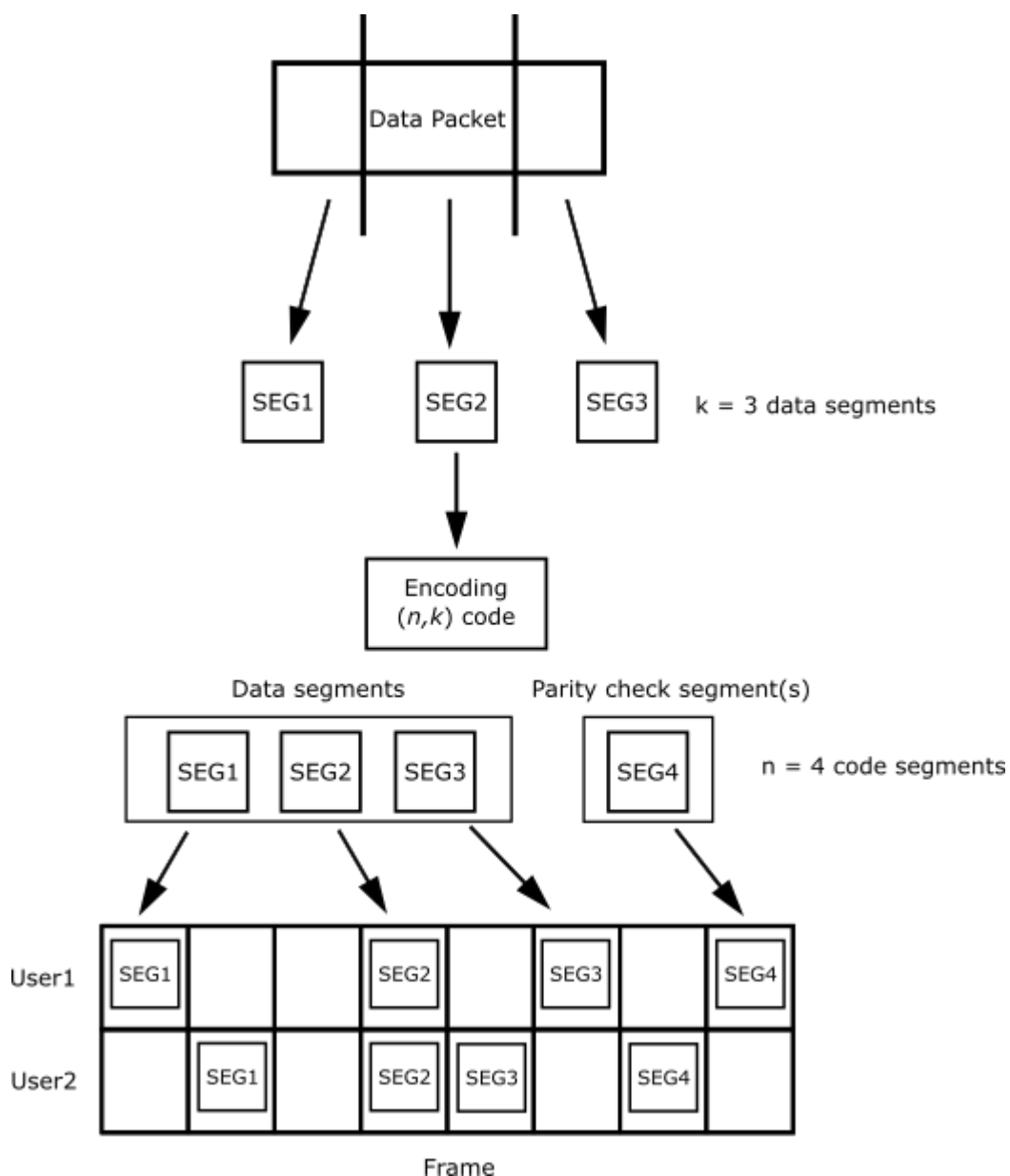


Рисунок 6 – Пример доступа в канал

Как мы видим, пользователи передают закодированные данные на конкретных подкадрах, для каждого из пользователей они определяются выбранной им протокольной последовательностью. Первый пользователь использует протокольную последовательность $[10010101]$, а второй $[01011010]$. В нашем примере возникает коллизия на 4 подкадре, она приведет к стиранию пакета, которое может быть исправлено декодере сверточного кода.

4.3. Выбор протокольных последовательностей

Для задания набора протокольных последовательностей, мы используем регулярный МППЧ-код. Выбор в качестве протокольных последовательностей строк его проверочной матрицы, может быть хорошим решением, так как его характеристики позволяют нам точно знать, сколько потерь пакетов соответствует одновременному выбору соответствующих групп протокольных последовательностей. Нам остается подобрать код с удовлетворяющими нас характеристиками. Так как размер блока, используемого нами для кодирования и декодирования, сверточного кода равен 4, нам необходимо, чтобы в каждой из строк было как минимум 4 единицы. Таким образом мы можем рассматривать класс МППЧ-кодов с $K = 4$. Были рассмотрены варианты ММППЧ-кодов из [1]. В качестве примера был выбран код с обхватом равным 6. В общем случае число единиц в одном столбце мы можем варьировать, но мы рассматриваем $J = 2$. В таком случае для каждой пары протокольных последовательностей при передаче одного пакета у нас будет максимум по 2 коллизии, что нас вполне устраивает, учитывая корректирующую способность нашего кода. Стоит сказать, что для упрощения обозначений в каждой строке проверочной матрицы указаны только позиции и значения степеней мономов ее ненулевых элементов. Таким образом пара (i, j) обозначает, что в строке на i -ой позиции стоит моном D^j . Приведем пример используемого нами МППЧ-кода, задав его проверочную матрицу.

$$H_{conv}(D) = \begin{pmatrix} (1, 0) & (2, 0) & (3, 0) & (4, 0) \\ (1, 1) & (5, 0) & (6, 0) & (7, 0) \\ (2, 4) & (5, 0) & (8, 0) & (9, 35) \\ (3, 29) & (6, 15) & (8, 0) & (10, 0) \\ (4, 0) & (7, 0) & (9, 0) & (10, 26) \end{pmatrix} \quad (11)$$

Теперь, для того, чтобы получить протокольные последовательности, необходимо разложить матрицу $H_{conv}(D)$ по степеням D , и тогда циклические сдвиги матричного вида, полученного нами разложения и будут являться протокольными последовательностями. Граф Таннера этого кода имеет обхват равный 6.

Чтобы получить конечные протокольные последовательности проведем усечение. Для заданной матрицы усечение проводится на длине $M = 78$. Тогда мы получим матрицу протокольных последовательностей размера 390×780 . Тогда получается, что максимальное число пользователей в канале равняется 390.

Проведем моделирование для нашего примера и посмотрим на число конфликтов, а так же на число пользователей в канале. Опишем протокол поведения пользователя. Каждый пользователь с заданной вероятностью хочет передать сообщение по каналу. Случайным образом он выбирает одну из P протокольных последовательностей, которая будет определять его дальнейшее поведение в канале. Каждый пользователь на подкадре кадра может выйти или не выйти в канал, но если выходит и получает протокольную последовательность, то начинает работать не с ее начала, а с соответствующего подкадра. Смоделируем такую систему и посмотрим результаты.

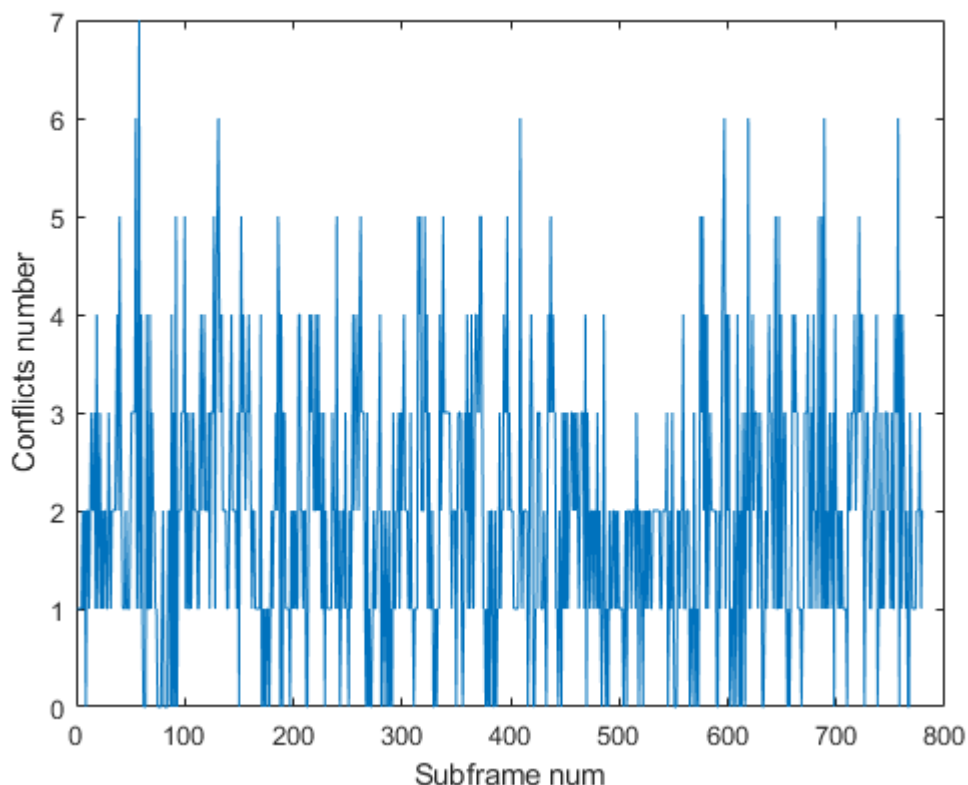


Рисунок 7 – Число конфликтов в канале

Как мы видим, для нашего примера, максимально число коллизий в подкадре составило 7. Такой показатель обусловлен тем, что пользователь выбирает протоковую последовательность полностью случайно, и из-за этого случаются повторения. Если же ограничить выбор, только свободными протоковыми последовательностями, а мы это можем сделать, так как предполагается, что во время работы канала, информация о них доступна всем, то мы получим такие результаты.

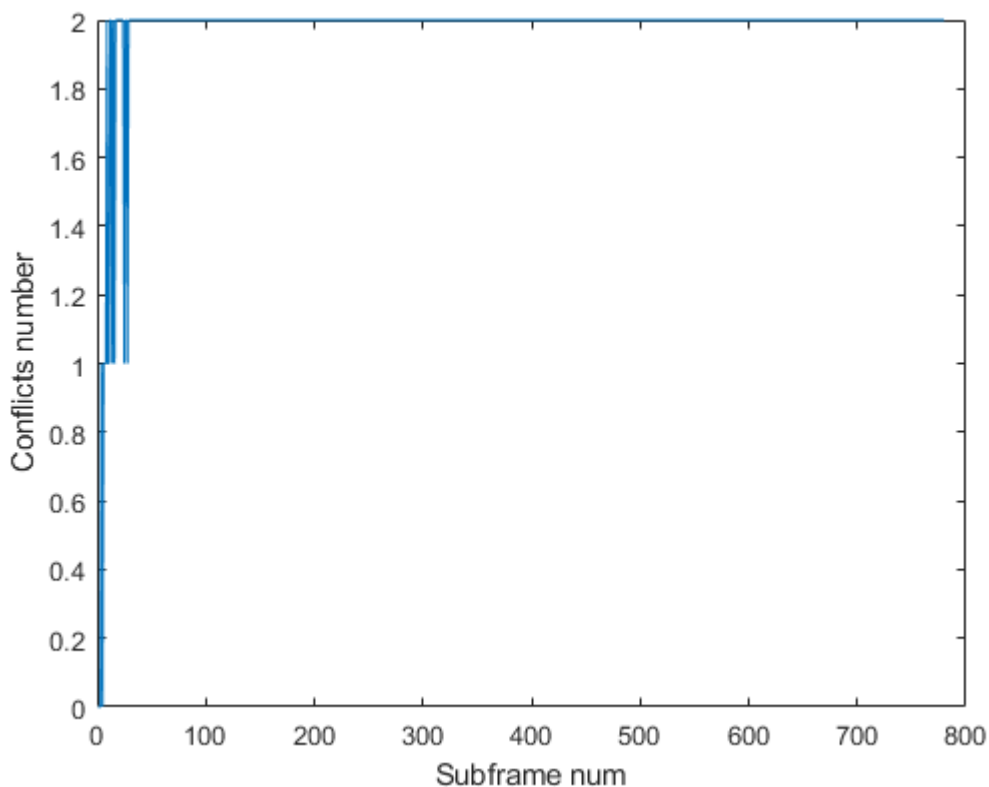


Рисунок 8 – Число конфликтов в канале (уникальные последовательности)

График показывает нам, что количество конфликтов в момент времени, останавливается на 2, что как раз соответствует числу единиц в столбце проверочной матрицы нашего графового кода. Но даже в случае с полностью случайным выбором протокольных последовательностей среднее число конфликтов стремится к 2. Посмотрим теперь на рост числа пользователей в канале. Напомним, что все они выходили на связь равновероятно.

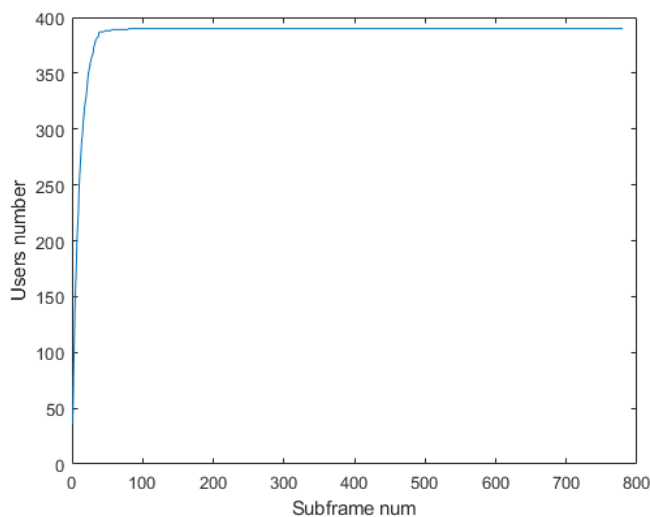


Рисунок 9 – Число активных пользователей в канале

4.4. Математические оценки на число активных пользователей в канале

Приведенное в предыдущем пункте моделирование было основано на конкретном примере, мы же хотим дать оценки в общем случае. Опишем нашу систему, пусть у нас есть код, задающий множество наших протокольных последовательностей. Обозначим через M число таких протокольных последовательностей. Вероятность передачи данных по каналу для одного пользователя равняется p . Чтобы упростить вычисления, введем следующие предположения:

- а) Мы анализируем ситуацию, когда один пользователь постоянно работает (он уже выбрал известную всем протокольную последовательность), а вновь входящие в сеть пользователи создают конфликты при передаче. Оценим вероятность потери пакета для этого пользователя. Будем считать, что для всех пользователей ситуация симметричная, то есть вероятность потери пакета не зависит от номера протокольной последовательности.
- б) Пользователи слушают канал и знают, какие протоколы уже активны. Выбор одинаковых протоколов возможен только в момент первого выхода в канал более одного пользователя. Такие конфликты пока не учитываются, они будут рассмотрены отдельно.
- в) Считаем, что число протокольных последовательностей M много больше числа одновременно работающих (активных) пользователей k .

Без потери общности считаем, что первый, постоянно работающий, пользователь использует протокольную последовательность номер 1. Тогда вероятность потери пакета для этого пользователя при условии, что еще k пользователей активны равна:

$$P_e(k) = P_a(k) \binom{M}{k}^{-1} N^{-1} \sum_{(i_1, \dots, i_k)} w(x_1 \cap (x_{i_1} \cup x_{i_2} \dots \cup x_{i_k})) \quad (12)$$

Поясним выражения используемые в этой формуле. $P_a(k)$ — это вероятность того, что k пользователей (протокольных последовательностей) активны. Она равна:

$$P_a(k) = \binom{M}{k} p^k (1 - p)^{M-k}$$

N — это вес одного кадра протокольной последовательности, в нашем случае он будет совпадать с длиной используемого МППЧ-кода. (i_1, \dots, i_k) — это номера протокольных последовательностей, выбранных активными пользователями (предполагаем, что они все разные).

$w(\dots)$ — это число конфликтов, которое подсчитывается как вес Хэмминга последовательности равной логическому И от последовательности x_1 и логического ИЛИ от последовательностей остальных активных пользователей.

Приведем пример. Пусть $k = 3$ и $M = 5$ (общее число пользователей 6, включая постоянно работающего первого пользователя).

$$P_e(3) = P_a(3) * \binom{5}{3}^{-1} * N^{-1} * (w(x_1 \cap (x_2 \cup x_3 \cup x_4)) + w(x_1 \cap (x_1 \cup x_3 \cup x_5)) + w(x_1 \cap (x_2 \cup x_4 \cup x_5)) + w(x_1 \cap (x_3 \cup x_4 \cup x_5)))$$

Другими словами, если в последовательностях активных пользователей возникает конфликт, то он рассматривается как один конфликтующий пакет даже, если совпадение произошло для большего числа пользователей.

Ситуацию одновременного выхода на связь нескольких пользователей, и при этом выбора ими одной и той же последовательности, будем решать так: конфликтующие пользователи прекращают данный сеанс, снова выходят в следующем кадре и снова разыгрывают между собой последовательности и так до успеха. При большом числе протокольных последовательностей вероятность конфликтов и, тем более, повторных конфликтов такого вида мала. Либо мы можем в таком случае одному из этих j пользователей выдавать выбранную последовательность, а остальных $j - 1$ отправлять на повторные попытки. В таком случае такие конфликты, гарантированно будут решаться за конечное время.

4.5. Моделирование

Соединим вместе использование протокольных последовательностей и корректирующий сверточный код. При моделировании для каждого пользователя вероятность попытки доступа в канал, при каждом тесте она изменялась, для большей достоверности результатов. Каждое передаваемое сообщение состояло из 100 блоков. Пользователи выбирали протокольные последователь-

ности случайным образом, но повторения не допускались. Для каждой конфигурации вероятностей выхода и распределения последовательностей проводилось по 100 тестов. В качестве источника протокольных последовательностей рассматривался код (11). Рассматривался число пользователей от 2 до 100 с шагом 5. Обычно считается, что в сети V2V, в зоне, обрабатываемой протоколом, находится несколько десятков машин. Приведем результаты моделирования:

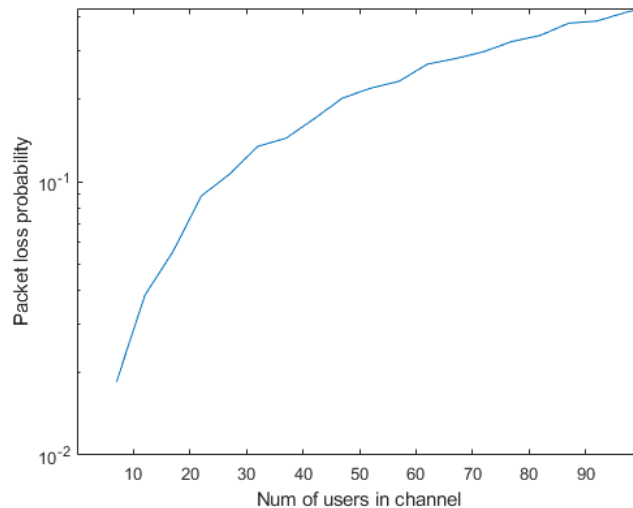


Рисунок 10 – Вероятность потери пакета в зависимости от числа пользователей

Как мы видим, когда число пользователей превышает 50, происходит резкий скачок вероятности потери пакета. Это обусловлено тем, что число протокольных последовательностей, порожденных нашим кодом, равно 390, и после 50 активных пользователей, уже нельзя сказать, что $k \ll M$, поэтому мы и видим, такое увеличение вероятности потери пакета.

Выводы по главе 4

В данной главе мы привели способ построения множества протокольных последовательностей. Смоделировали доступ в канал и оценили вероятность потери пакета как функцию от числа активных пользователей с использованием и без использования сверточного кодирования. Показано, что вероятность потери пакета при использовании сверточного кодирования уменьшается.

ЗАКЛЮЧЕНИЕ

В данной работе был успешно реализован протокол доступа на основе графового кода, удовлетворяющий требованиям на низкую задержку при передаче. Также использование высокоскоростного кода для исправлений стираний, показало хорошую корректирующую способность. Также была дана оценка на вероятность потери пакетов в этой системе, в зависимости от числа активных пользователей...После было смоделировано совместное использование протокольных последовательностей и корректирующего кода.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 *Бочарова И. Е., Кудряшов Б. Д., Сатюков Р. В.* Сверточные и блочные коды с малой плотностью проверок на основе графов // Проблемы передачи информации. — 2009. — Т. 45, № 4. — С. 69–90.
- 2 *Кудряшов Б.* Основы теории кодирования: учеб. пособие // СПб.: БХВ-Петербург. — 2016. — Т. 400.
- 3 In-depth analysis and evaluation of self-organizing TDMA / T. Gaugel [et al.] // 2013 IEEE vehicular networking conference. — IEEE. 2013. — P. 79–86.
- 4 *Leiner B. M.* LDPC Codes—a brief Tutorial // Apr. — 2005. — Vol. 8. — P. 1–9.
- 5 Low Delay Inter-Packet Coding in Vehicular Networks / I. Bocharova [et al.] // Future Internet. — 2019. — Vol. 11, no. 10. — P. 212.
- 6 *Massey J., Mathys P.* The collision channel without feedback // IEEE Transactions on Information Theory. — 1985. — Vol. 31, no. 2. — P. 192–204.
- 7 *Omar H. A., Zhuang W., Li L.* VeMAC: A TDMA-based MAC protocol for reliable broadcast in VANETs // IEEE transactions on mobile computing. — 2012. — Vol. 12, no. 9. — P. 1724–1736.
- 8 On the ability of the 802.11 p MAC method and STDMA to support real-time vehicle-to-vehicle communication / K. Bilstrup [et al.] // EURASIP Journal on Wireless Communications and Networking. — 2009. — Vol. 2009, no. 1. — P. 902414.