УНИВЕРСИТЕТ ИТМО

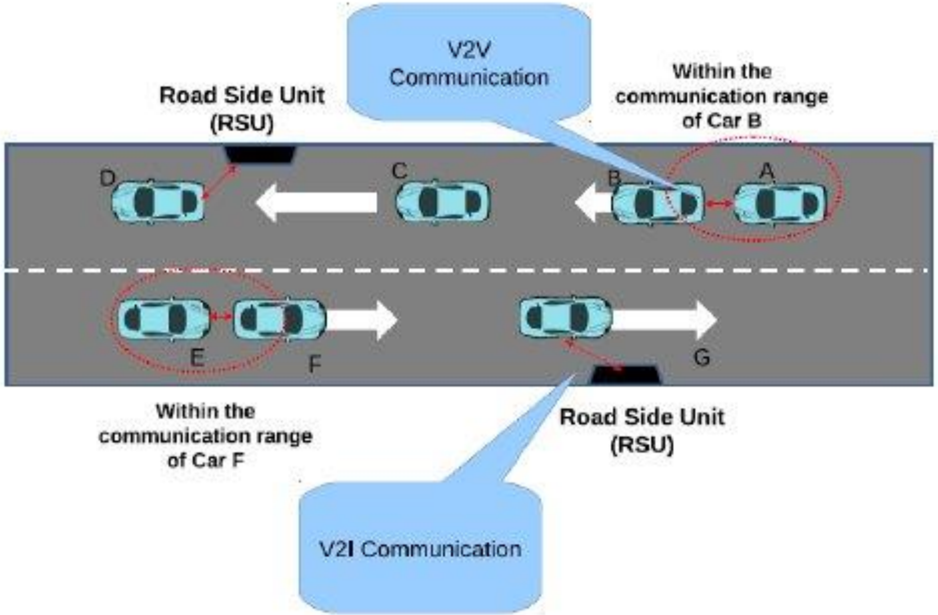
Бакалаврская работа на тему:

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

Галкин Егор Георгиевич, M3435

Научный руководитель: Бочарова И.Е., к.т.н, доцент ФИТиП

Объект исследования



Vehicular ad-Hoc networks (VANETs)—An overview and challenges

Формулировка проблемы

Взаимодействие V2V очень чувствительно к задержке доставки пакетов, а также их потере. Одной из главных проблем является появление коллизий в канале. Существующие алгоритмы пытаются минимизировать число коллизий и/или используют обратную связь для их разрешения. Вопрос восстановления потерянных пакетов с использованием корректирующих кодов является предметом изучения данной работы.

Актуальность работы

В данный момент существующие стандарты не предоставляют подходящих решений для V2V взаимодействия в сетях VANET. От протоколов требуется маленькая задержка при передаче, а также высокая надежность. Проводятся разные исследования в этой области и одно из направлений исследования - это применение кодов для реализации протокола.

Цель

Разработка протокола множественного доступа для сетей V2V, использующего протокольные последовательности, построенные на основе графовых кодов. Разрешение коллизий с применением корректирующих кодов для исправления стираний. Оценка вероятности потери пакета, при использовании протокола.

Задачи

1. Реализация кодера и декодера сверточного кода, используемого для исправления стираний.
2. Генерация протокольных последовательностей, минимизирующих число коллизий.
3. Оценка вероятности потери пакета с заданным набором протокольных последовательностей.
4. Моделирование доступа в канал и анализ результатов.

Исправление стираний

Почему вообще мы можем исправлять коллизии с помощью сверточного кода?

При возникновении коллизии в канале, все участвующие в коллизии пакеты стираются. Существуют решения такие как Slotted-Aloha, CSMA, STDMA, но в большинстве своем они лишь пытаются предотвратить коллизии. Мы же будем рассматривать стирание при коллизии, как стирания в двоичном канале со стираниями. А в таком случае мы можем попытаться их восстановить.

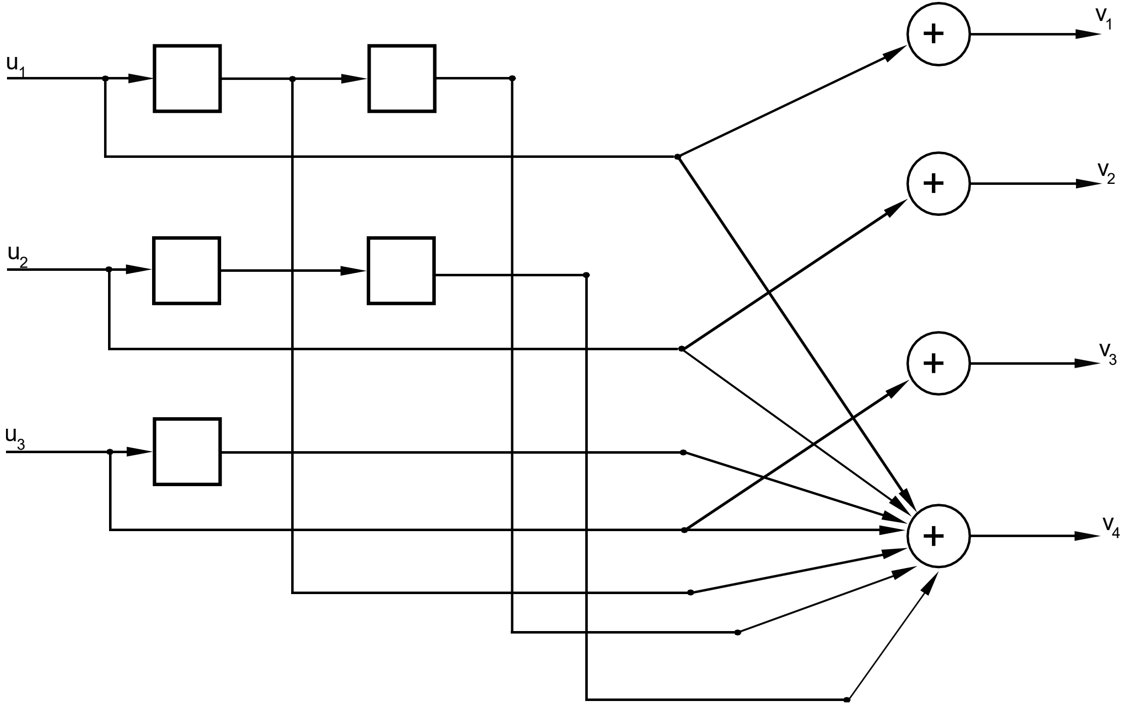
Выбор кода

Сначала выберем код для исправления стираний. Был рассмотрен класс кодов с $R = \frac{(n-1)}{n}$. С учетом высоких требований к задержке и сложности декодирования был выбран код с порождающей матрицей.

$$\mathrm{G} = \begin{pmatrix} 1 & 0 & 0 & 1 + D + D^2 \\ 0 & 1 & 0 & 1 + D^2 \\ 0 & 0 & 1 & 1 + D \end{pmatrix}$$

Наш код обладает следующими характеристиками: $k = 3, n = 4, R = \frac{3}{4}, d_{free} = 3, m = 2.$ На примере этого кода, покажем алгоритмы кодирования и декодирования.

Схема кодера



Декодирование

Декодирование в двоичном канале со стираниями сводится к решению системы линейных уравнений. Используя оконное декодирование, мы получаем сложность алгоритма линейную по длине последовательности, так как сложность декодирования в окне равна $O(v^3)$, где v – число стираний в окне, т.е. константе по длине кода. Для декодирования будет использоваться проверочная матрица.

$$H = \begin{pmatrix} 1 + D + D^2 & 1 + D^2 & 1 + D & 1 \end{pmatrix}$$

Разложив эту матрицу по степеням D, получим полубесконечную матрицу, из которой мы возьмем только определенную подматрицу, использующуюся в процессе декодирования.

Декодирование

Для нашего кода она имеет вид:

$$H_{ZT} = \begin{pmatrix} 1100 & 1010 & 1111 & & \\ & 1100 & 1010 & 1111 & \\ & & 1100 & 1010 & 1111 \end{pmatrix}$$

Далее по ней, с помощью оконного декодирования строим систему и декодируем по максимум правдоподобия. Размер окна определяется по формуле:

$$W = L + m + 1$$

L – параметр окна. В нашем примере L = 2.

Декодирование

........0 0 0 0 0 0 0 0 0 1 0 -1 -1 1 0 0 1 -1 0 1 0 0 0 1 0 0 0 1........

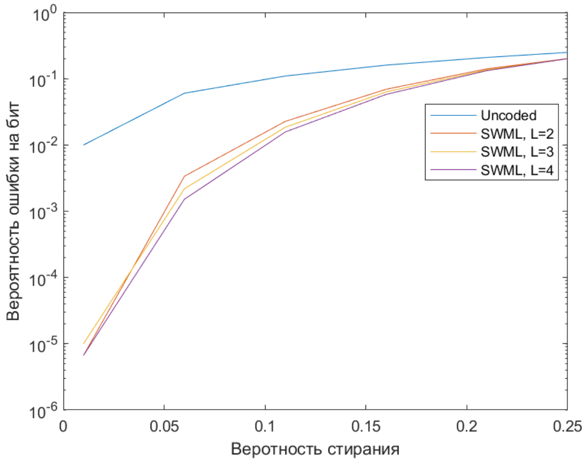
.........0 0 0 0 0 0 0 0 0 1 0 1 -1 1 0 0 1 -1 0 1 0 0 0 1 0 0 0 1.........

.........0 0 0 0 0 0 0 0 0 1 0 1 1 1 0 0 1 -1 0 1 0 0 0 1 0 0 0 1.........

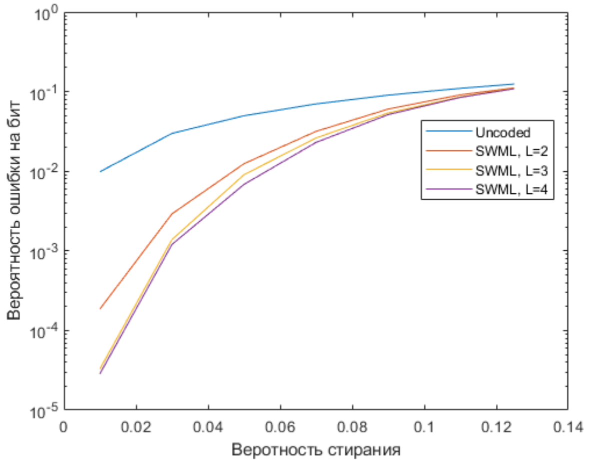
.........0 0 0 0 0 0 0 0 0 1 0 1 1 1 0 0 1 0 0 1 0 0 0 1 0 0 0 1..........

Пример оконного декодирования, L = 2

Тестирование



Вероятность ошибки на бит, $R = \frac{3}{4}$



Вероятность ошибки на бит, $R = \frac{7}{8}$

Схема доступа в канал

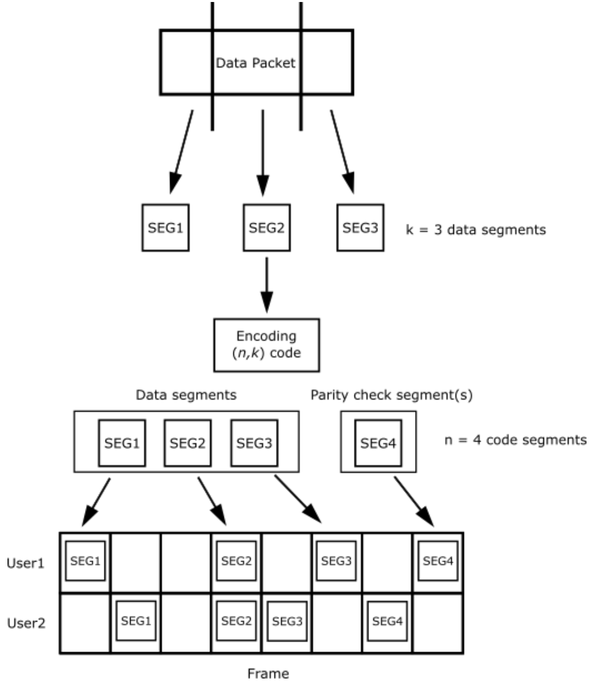
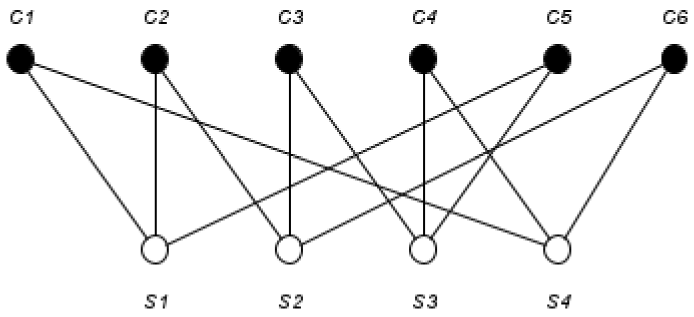


Схема доступа в канал

Построение протокольных последовательностей

В качестве источника протокольных последовательностей используем графовый МППЧ-код. Графовый код так называется потому, что его проверочная матрица – это матрица инцидентности некоторого графа. Его особенность в том, что каждый столбец проверочной матрицы содержит только 2 единицы. Его также можно представить с помощью графа Таннера



Пример графа Таннера

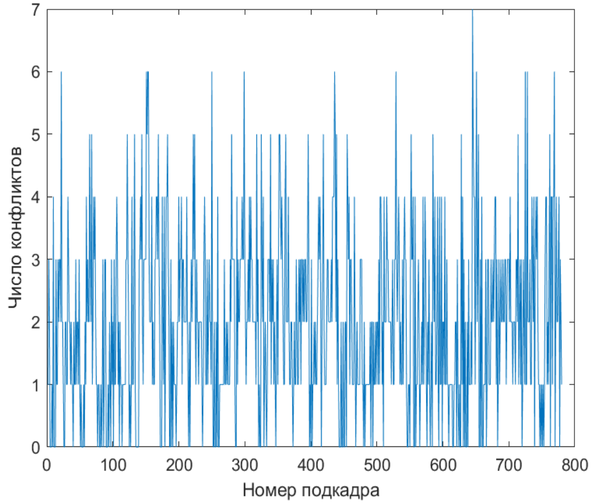
Источник протокольных последовательностей

В качестве источника последовательностей был выбран (2,4) регулярный МППЧ-код.

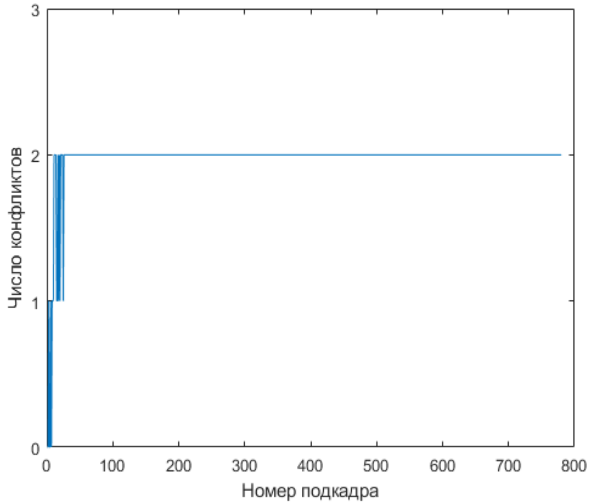
$$H_{conv} = \begin{pmatrix} (1,0) & (2,0) & (3,0) & (4,0) \\ (1,1) & (5,0) & (6,0) & (7,0) \\ (2,4) & (5,0) & (8,0) & (9,35) \\ (3,29) & (6,15) & (8,0) & (10,0) \\ (4,0) & (7,0) & (9,0) & (10,26) \end{pmatrix}$$

Проведя его усечение, получим набор последовательностей, в качестве которого будут выступать строки из проверочной матрицы.

Тестирование протокольных последовательностей



Число конфликтов на подкадре



Число конфликтов на подкадре
(уникальные последовательности)

Математическая оценка вероятности потери пакета

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

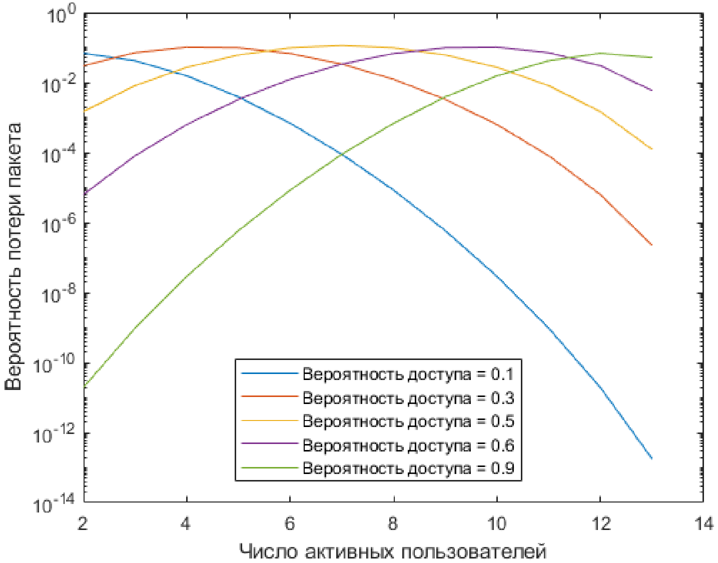
$P_a(k)$ — вероятность того, что k пользователей одновременно выйдут в канал.

M — число протокольных последовательностей.

N — длина протокольной последовательности.

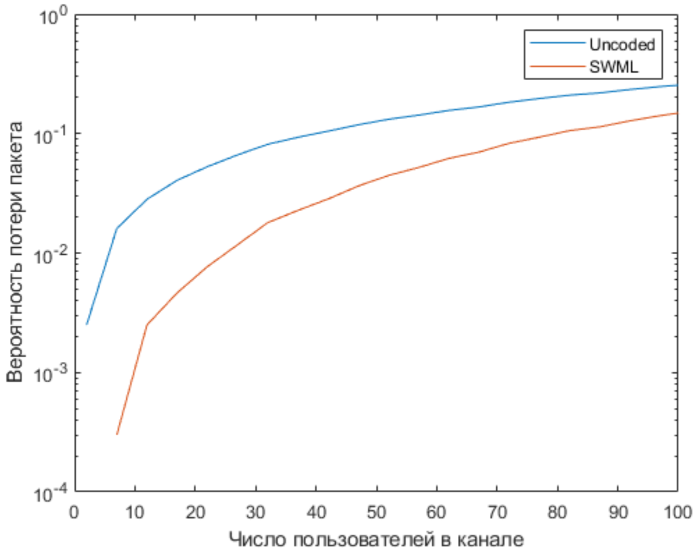
$w(..)$ — число конфликтов.

Математическая оценка вероятности потери пакета



Зависимость вероятности потери пакета
от числа активных пользователей.
(2, 3)-регулярный МППЧ-код.

Моделирование



Вероятность потери пакета при моделировании.
(2, 4)-регулярный МППЧ-код.

Сравнение

Без потери общности, мы считаем, что алгоритмы, использующие разделение по времени, эквивалентны использованию ортогональных протокольных последовательностей. Проведем сравнение на максимальном рассматриваемом нами при моделировании числе пользователей. В таком случае вероятность потери пакета равняется 10^{-1}.

Пусть X — число кадров в течение которых мы передаем данные. Тогда число пакетов, которые мы можем передать, используя ортогональные последовательности равняется:

$$2X$$

Число же пакетов, которые мы можем передать, используя наш протокол, равняется:

$$JX = 4X$$

Сравнение

В нашем протоколе мы передаем избыточные пакеты. Их число можно посчитать как.

$$RP = FN \times PPF \times (1 - R)$$

FN — число кадров, на которых передавались данные

PPF — число пакетов, которые можно передать за один кадр

R — скорость кода

Сравнение

Посмотрим какой выигрыш мы в итоге получим:

$$\frac{4 \times X\ - (X \times 4 \times \left(1\ - \frac{3}{4}\right))) \times (1\ - 10^{-1})}{2 \times X} = \frac{3 \times X \times 0.9}{2 \times X} = 1.35$$

Сравнение

Наш метод является эффективным вплоть до вероятности потери пакета равной 0.3. Такая вероятность потери пакета превосходит значения типичные для практических сценариев связи

Результаты

- Был разработан протокол взаимодействия в сети с возможностью исправления стираний.
- Даны оценки на вероятность потери пакетов в зависимости от числа пользователей.
- Проведено моделирование и сравнение.

Направления дальнейшего исследования

1. Рассмотрение для исправлений других классов кодов, например коды Рида-Соломона.
2. Учет потери пакетов на физическом уровне и их дальнейшее исправление.
3. Более детальное сравнение и возможное испытание в реальной среде.