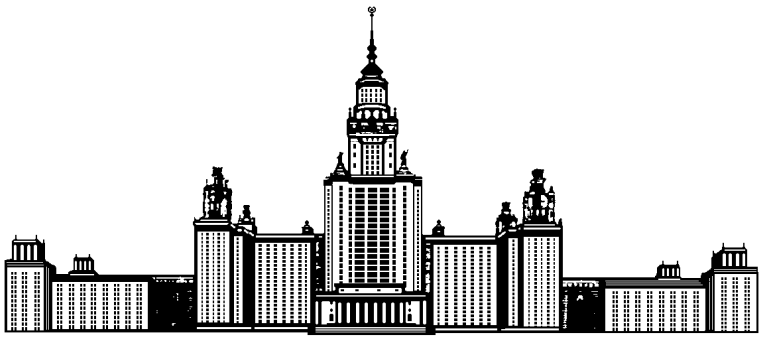
Использование идей фонтанного кодирования при передаче малого числа битовых пакетов

Иванов Егор Романович, 417 группа
Научный руководитель: к.ф.-м.н. Гуров Сергей Исаевич

ММП ВМК МГУ

8 ноября 2024 г.



Содержание

Глобальная задача

Общая задача

Требуется передать по стирающему каналу[a] связи информацию в виде K битовых векторов.

[a] BEC (Binary Erasure channel)

- Почему нельзя просто перезапросить информацию?
  Ответ: ее может уже не быть на сервере (потоковое телевещание)
- Применимы ли алгебраические коды (например, Рида-Соломона)? Что если размер одного битового вектора велик? или *очень* велик?
  Ответ: алгебраические коды эффективны в случае «не слишком больших» значений своих параметров и сильно зависимы от априорной оценки на потери в канале.

Новая аксиоматика

- Минимальная единица информации – **пакет** – длинный битовый вектор.
- Над пакетами определена операция суммы по модулю 2 ($\oplus$) как соответствующая покомпонентная.
- Обращаться к битовой структуре пакета **нельзя**:
  - для отправителя и получателя это означает, что нельзя использовать что-то, кроме операции $\oplus$
  - для канала это означает, что пакет либо доходит целиком, либо не доходит вовсе

Замечание

Примем следующее допущение: цена утери пакета достаточно низкая. Иными словами, ошибки декодирования, конечно, нежелательны, но могут происходить «не слишком часто». Пример: потоковое вещание.

Такая постановка позволяет перейти от алгебраических подходов к стохастическим.

Линейный фонтан

Идея

Для формирования нового кодирующего символа s_{new} каждый из пакетов x независимо от остальных включается в сумму с вероятностью $\frac{1}{2}$.

Получатель знает, какие пакеты были сложены и формирует матрицу $\mathbf{G}$.[1] Новый полученный пакет – новое уравнение в системе. Возможность полного декодирования эквивалентна обратимости матрицы системы.

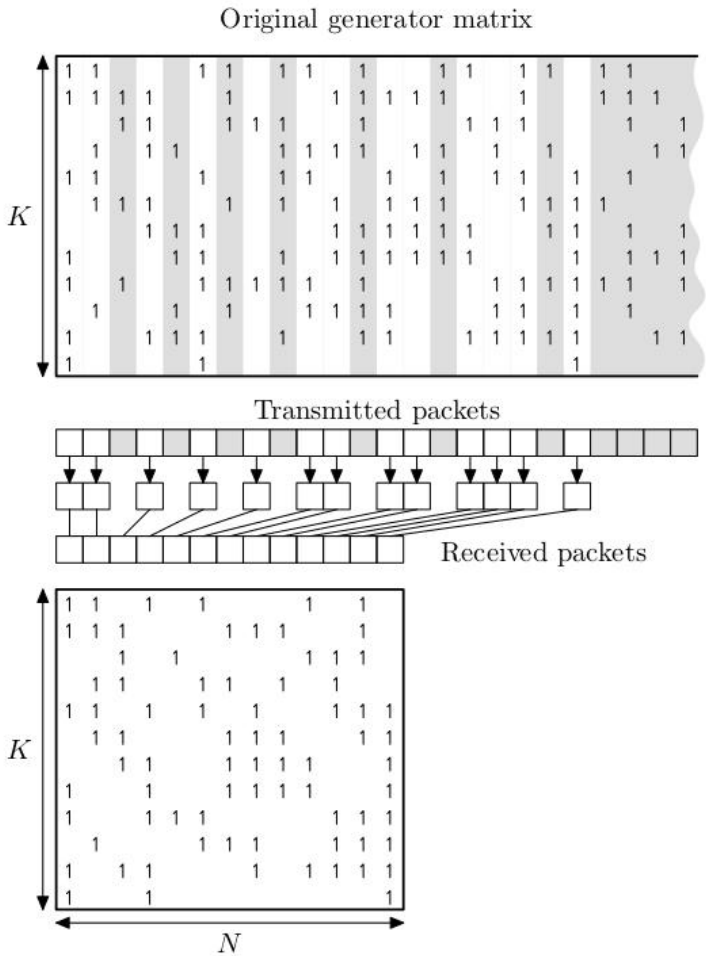
$$\mathbf{Gx} = \mathbf{s},\ \mathbf{G} \in \mathbb{Z}_2^{K \times K},\ \mathbf{x} = (x_1, \ldots, x_K),\ \mathbf{s} = (s_1, \ldots, s_K)$$

$$\mathrm{P}\left\{\exists\, \mathbf{G}^{-1}\right\} = \left(1 - \frac{1}{2}\right)\left(1 - \frac{1}{4}\right)\ldots\left(1 - \frac{1}{2^K}\right) \approx 0.289,\ \forall\, K \geq 10$$

символа Конечно, мы бы хотели 0.999, но полученное число

[1] MacKay, “Fountain codes”

Пояснение



На рисунке слева на белом фоне переданные через канал пакеты, на сером – стертые.
С помощью переданных пакетов (received packets) формируется матрица.

Отправляем больше

Помним о том, что мы все еще не пользовались избыточностью. Отправим $N = K + E$ пакетов. Тогда вероятность ненахождения обратимой подматрицы равномерно ограничена по всем K:

$$\delta(E) \leq \frac{1}{2^E}$$

Недостатки

Обратимую подматрицу нужно

- Обратимую подматрицу нужно найти и обратить
- Метод плохо масштабируем в силу кубической асимптотики

Зачем нам матрицы?

Предложение 1

Представим задачу в виде двудольного графа: в одной доле пакеты, в другой – кодирующие символы.
Отношение связности – это отношение включения в сумму: то есть символ s_i и пакет x_j смежны тогда и только тогда, когда $s_i = x_j \oplus \dots$.

Определение

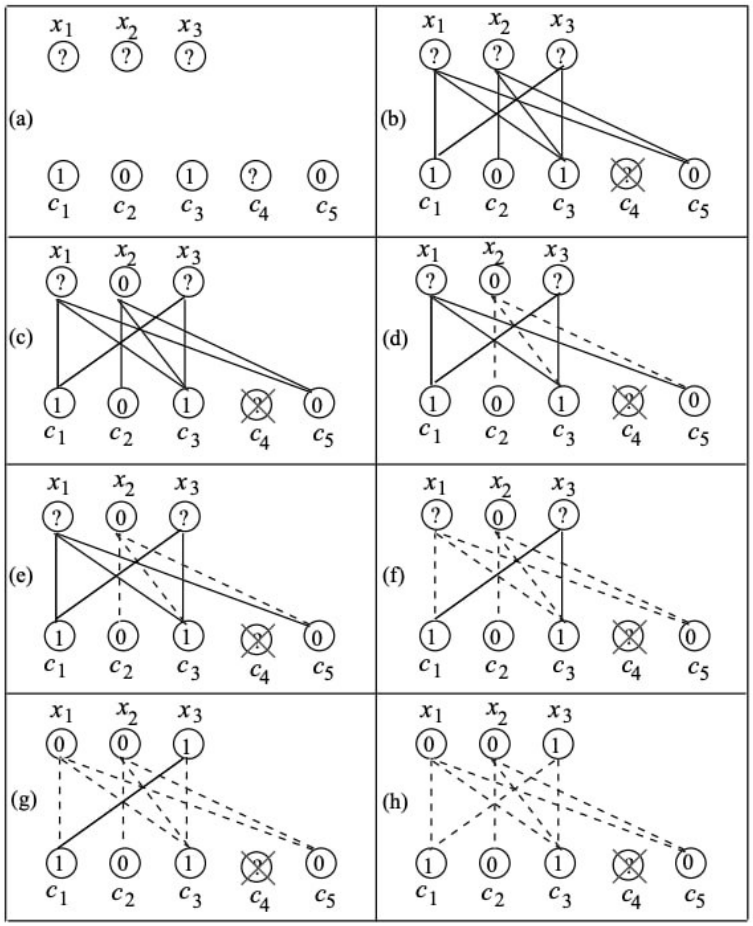
Граф, описанный в предложение 1, называется *графом Таннера*[a].

[a] Tanner, "A recursive approach to low complexity codes"

Предложение 2

Последовательно декодируем то, что можем декодировать, и переходим к задаче меньшей размерности.

Лучше показать пример



a) символы c_1, c_2, c_3, c_5 прошли через канал

b) визуализируем информацию о связности

c) пакет x_2 дошел «как есть» в виде символа c_2, восстановим его

d) пакет x_2 присутствует участвовал в следующих символах:
$c_3 = x_1 \oplus x_2 \oplus x_3$ и $c_5 = x_1 \oplus x_2$

e) прибавим к обеим частям уравнений выше x_2: $c_3 \oplus x_2 = x_1 \oplus x_3$, $c_5 = x_1$

f)-h) по тому же алгоритму решаем задачу меньшей размерности

ЛТ-процесс (LT-process, belief propagation)

Предложение

Подберем такой алгоритм генерации символов, чтобы ЛТ-процесс завершался успешно с достаточно большой вероятностью.

Схема генерации выглядит так: у нас есть заданное распределение степеней символов

$$\boldsymbol{\rho} = (\rho_1, \ldots, \rho_K),\ \rho_i \geq 0,\ \sum_i \rho_i = 1$$

При формировании нового символа из этого распределения сэмплируется величина d и для суммирования равновероятно выбираются d из K пакетов:

$$d \sim \boldsymbol{\rho},\ s_{new} = \sum_{i=1}^{d} x_{j_i},\ j_i \in \{1, \ldots, K\},\ s \neq p \Rightarrow j_s \neq j_p$$

Идеальное солитонное распределение

Определение

Идеальным солитонным распределением ρ называется следующий набор чисел:

$$\rho_i = \begin{cases} \frac{1}{K}, & i = 1 \\ \frac{1}{i(i-1)}, & i = 2, \ldots, K \end{cases}$$

Наводящее соображение: на каждом этапе у нас появляется в точности один 1 символ.

Недостатки

- Не работает
- Слишком строгое ограничение на процесс

Робастное солитонное распределение

Определение

Робастным солитонным распределением μ с параметрами $c > 0$ и $0 < \delta < 1$ называется набор чисел, получаемых следующим образом: положим $R = c \ln(K/\delta)\sqrt{K}$ и

$$\tau_i = \begin{cases} \frac{R}{i \cdot K}, \ i = 1, \ldots, \frac{K}{R} - 1 \\ \frac{R \ln(R/\delta)}{K}, \ i = \frac{K}{R} \\ 0, \ i = \frac{K}{R} + 1, \ldots, k \end{cases}$$

Тогда μ_i – нормированное $\rho_i + \tau_i$.

Наводящее соображение: изменение размера очереди рассматривается как случайное блуждание (стартуем с запасом и используем его, чтобы не сломать раньше времени).

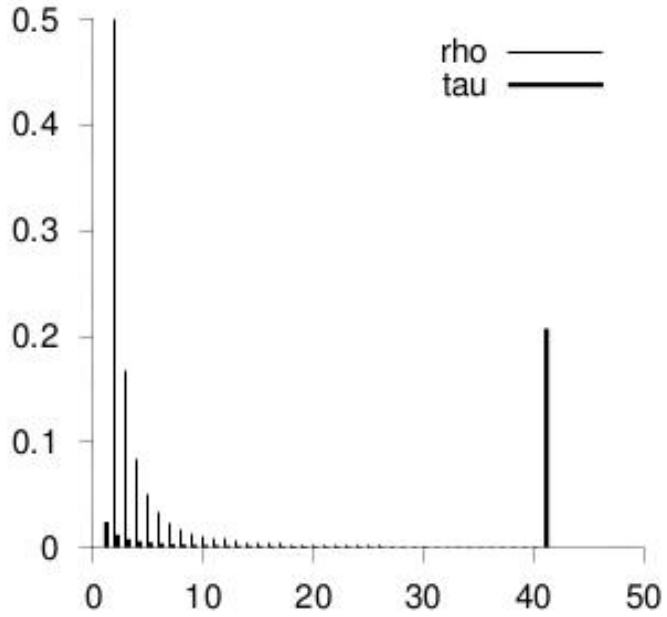


Рис.: Пример робастного солитонного распределения, $K = 10^4, c = 0.2, \delta = 0.05$

Теорема (Лаби, 2002)

При получении $K \cdot \sum_i (\rho_i + \tau_i)$ символов по распределению $\mu_{c,\delta}$ вероятность полного декодирования составляет хотя бы $1 - \delta$. [a]

[a] Luby, "LT codes"

Постановка задачи

Идея фонтанного кодирования не содержит внутри себя обращения к **порядку** отправки пакетов, поэтому коды работают даже если канал передачи информации крайне ненадежен. На практике же канал заведомо «хороший», то есть теряет небольшой процент пакетов. Используя идеи фонтанного кодирования, будем пытаться сделать хорошего канала **почти идеальный**.

- Процент потерь в канале достаточно мал ($\leq 5 - 7\%$)
- Количество пакетов, высылаемых при одних и тех же параметрах, невелико (пропускная способность канала зависит от времени)
- Коды систематические – сначала высылаем все «как есть»

Стратегии исследования

Опишем общие стратегии решения. У нас есть 2 степени свободы: алгоритм декодирования $\mathcal{D}$ и распределение $\boldsymbol{\rho}(\cdot)$.

Стратегия 1

Использование простого алгоритма декодирования (допускающего аналитический вывод формул) и решение задачи условной оптимизации.

Стратегия 2

Использование сложного алгоритма декодирования (слабо поддающегося анализу) и хорошо изученных (Бернулли, Пуассона) или просто устроенных распределений.

Стратегия 1. Пример простого алгоритма

Пусть имеются пакеты x_i, $i = 1, \ldots, 5$. Отправитель пересылает через канал их по очереди и до получателя доходят пакеты с номерами $1, 2, 4$. Далее отправитель пересылает 2 проверочных пакета:

$$s_1 = x_1 \oplus x_2 \oplus x_3$$

$$s_2 = x_1 \oplus x_3 \oplus x_5$$

- При получении s_1 декодирование будет проведено – восстановим x_3: все пакеты, кроме x_3 – полученные информационные.
- При получении s_2 декодирование проведено **не** будет: среди просуммированных пакетов есть два не из списка полученных информационных – x_3 и x_5.

Данный пример иллюстрирует возможную нерациональность простого алгоритма: очевидно, что к моменту получения s_2 пакет x_3 уже восстановлен и s_5 на самом деле восстановить можно.

Аналитические формулы

$$\mathrm{P}\{\text{пакет с номером } K \text{ не будет восстановлен}\} =$$

$$= \tau \sum_{l=0}^{K-1} \binom{K-1}{l} \sum_{C=0}^{M} \binom{M}{C} \tau^{K-1+M-l-C} (1-\tau)^{l+C} \left[1 - \sum_{d=1}^{l+1} \rho_d \frac{\binom{l}{d-1}}{\binom{K}{d}} \right]^C$$

$$=: \tau \cdot \mathcal{L}(\boldsymbol{\rho}, K, M)$$

Получили задачу условной оптимизации:

$$\begin{cases} \mathcal{L}(\boldsymbol{\rho}, K, M) \to \min \\ \rho_i \geq 0 \\ \sum\limits_i \rho_i = 1 \end{cases}$$

Результаты экспериментов. Метрики

K	M	$\tau\mathcal{L}$, %	Parity Packet Baseline, %
20	2	1.23	1.32
30	2	1.54	1.76
40	2	1.78	2.09
20	3	0.93	1.32
30	3	1.23	1.76
40	3	1.47	2.09
20	4	0.74	1.32
30	4	1.01	1.76
40	4	1.25	2.09

Таблица: Результаты экспериментов. $\tau = 0.03$

Результаты экспериментов

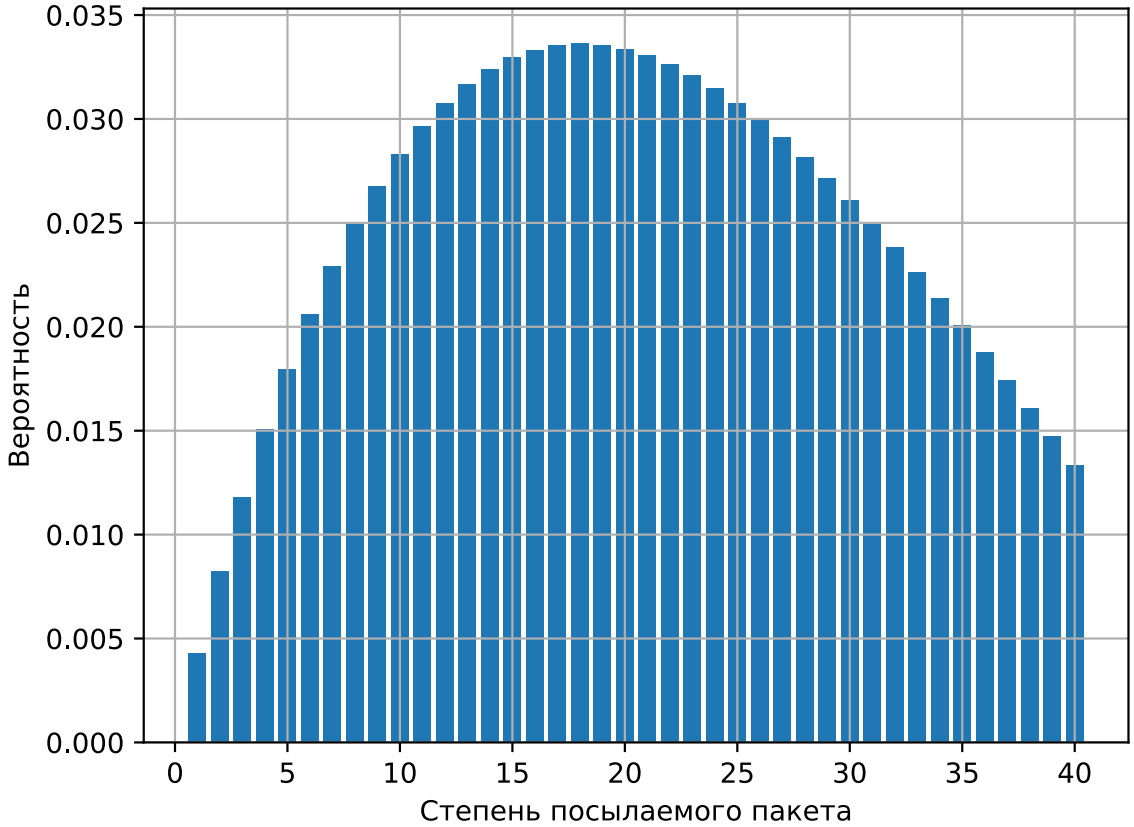


Рис.: $K = 40,\ M = 4$

Стратегия 2. Выбор распределений

Наблюдение

Хорошо работающие распределения идейно выглядят следующим образом: чаще всего генерируются символы одной и той же степени, которые постепенно расходуются для поддержания размера очереди.

Таким условиям удовлетворяют биномиальное и Пуассоновские распределения, а также, например:

$$\rho_i = \begin{cases} 1, & i = d \\ 0, & i \neq d \end{cases}$$

Рассмотрим их подробнее.

Биномиальное распределение

K	N	$p_{\text{опт}}$	p_{BEC}	τ
20	24	0.571	0.05	0.01
20	24	0.776	0.03	0.003
50	60	0.286	0.05	0.008
50	60	0.327	0.03	0.002

Таблица: Результаты экспериментов. Избыточность 20%

Пуассоновское распределение

K	N	$\lambda_{\text{опт}}$	p_{BEC}	τ
20	24	10.6	0.05	0.01
20	24	11.0	0.03	0.004
50	60	15.3	0.05	0.007
50	60	18.6	0.03	0.001

Таблица: Результаты экспериментов. Избыточность 20%

Направления дальнейших исследований

- Исследование дисперсии числа восстановленных пакетов
- Поиск способов точного вычисления вероятностей восстановления
- Изучение решений оптимизационной задачи из стратегии 1