

Оглавление

1	Введение	3
2	Анализ принципов кодирования/декодирования	3
2.1	Оптимальное (эффективное) кодирование	5
3	Помехоустойчивое кодирование	6
4	Блочный код	7
	Список использованных источников	10

1. Введение

Рассмотрим общую схему передачи данных по каналу связи:

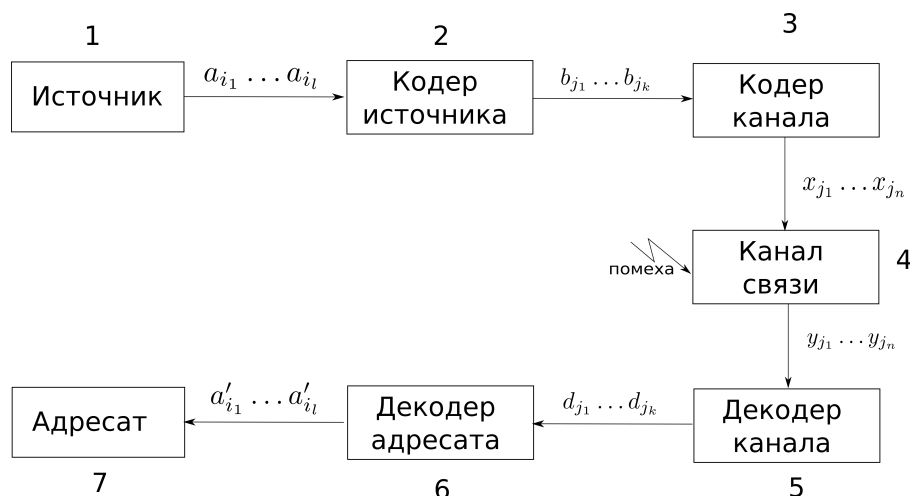


Рисунок 1 – Общая схема передачи данных по каналу связи

Задача передачи данных - добиться того, чтобы с вероятностью как можно ближе к 1, произошло событие $(a_{i_1} \dots a_{i_l}) = (a'_{i_1} \dots a'_{i_l})$.

Кодер источника (2) предназначен для того, чтобы так преобразовать информацию, порожденную источником, чтобы среднее количество передаваемых символом было минимальным (т.е. решить задачу оптимального (эффективного) кодирования). Блок (6) (декодер адресата) осуществляет обратное преобразование.

Блоки 3 и 5 (кодер и декодер канала) решают противоположную задачу: внесение такой избыточности в передаваемое сообщение которое позволит обнаружить и исправить большинство ошибок, которые могут возникнуть в канале связи (природа этих ошибок определяется свойствами среды, в которой существует канал связи) - это задача помехоустойчивого кодирования.

В рамках данной курсовой работы, рассматриваются блоки 3 и 5.

2. Анализ принципов кодирования/декодирования

Процесс кодирования это преобразование сообщения из одного алфавита в другой. Процесс декодирования же это обратное преобразование.

Введем определения согласно [1]:

Рассмотрим 2 алфавита: $A = \{a_1, \dots, a_m\}$ и $B = \{b_1, \dots, b_D\}$ ($|B| = D$, $|A| = m$)

Введем обозначение: $B^* = \cup_{i=1}^{\infty} B^i$ (т.е. B^* - множество всех возможных слов (любой конечной длины) состоящих из букв алфавита B)

Определение 2.1 Алфавитное D -ичное кодирование (кодирующее отображение, код) - отображение $\varphi : A \rightarrow B^*$

При этом $\varphi(a_i) \in B^*$ - называется кодovým словом

Определение 2.2 Кодирование называется однозначным, если отображение φ инъективно.

Определение 2.3 Кодирование называется префиксным, если никакое кодовое слово не является префиксом другого кодового слова.

Кодирование называется суффиксным, если никакое кодовое слово не является суффиксом другого кодового слова.

Теорема 2.1 Если алфавитное кодирование является префиксным или суффиксным, то оно однозначное. Обратное не верно.

Один из способов графического представления кода - корневое размеченное дерево, в котором каждое ребро помечено каким-либо символом из алфавита B , из каждой вершины выходит не более D ребер и каждому кодовому слову однозначно соответствует (существует биекция) лист такой, что последовательность меток ребер пути от корня до листа составляет кодовое слово.

Например рассмотрим следующий код: $A \rightarrow B$

$A \rightarrow 0$

$B \rightarrow 10$

$C \rightarrow 11$

Ему будет соответствовать следующее кодовое дерево:

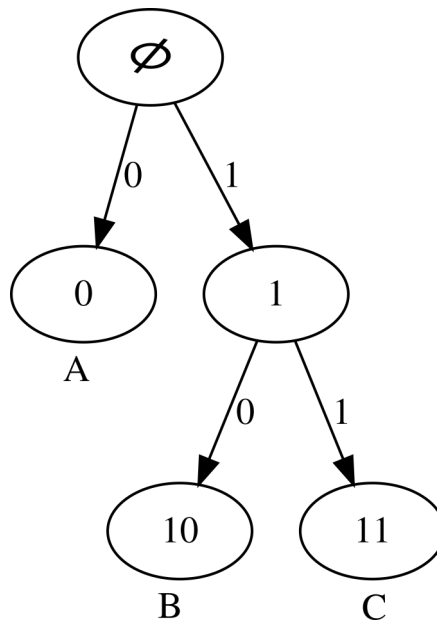


Рисунок 2

Теорема 2.2 Неравенство-Крафта

- 1) Если $\varphi: A \rightarrow B^*$ ($|A| = m$) - D -ичное префиксное алфавитное кодирование с длинами кодовых слов $\text{len}(\varphi(a_i)) = l_i$, $1 \leq i \leq m$, то справедливо неравенство

$$\sum_{i=1}^m D^{-l_i} \leq 1 \quad (1)$$

- 2) Если натуральные числа D, m, l_1, \dots, l_m удовлетворяют неравенству 1, то существует D -ичное префиксное алфавитное кодирование $\varphi: A \rightarrow B^*$ с длинами кодовых слов $\text{len}(\varphi(a_i)) = l_i$, $1 \leq i \leq m$.

Данная теорема позволяет не вдаваясь в подробности какой преобраз в какой образ отображается, проверить может ли вообще существовать однозначный префиксный код при данном наборе длин кодовых слов при данной мощности алфавита B . Если неравенство нарушается, то такого однозначного кода точно не существует.

2.1. Оптимальное (эффективное) кодирование

Цель оптимального кодирования - так преобразовать текст, чтобы средняя длина передаваемого сообщения была минимальной.

Пусть $\vec{p} = (p(a_1), \dots, p(a_m))$, где $p(a_i)$ - вероятность появления буквы $a_i \in A$. Основная идея оптимального кодирования - сопоставлять буквам с наибольшей вероятностью появления кодовые слова наименьшей длины.

Определение 2.4 Пусть заданы алфавит источника $A = \{a_1, \dots, a_m\}$, кодовый алфавит $B = \{b_1, \dots, b_D\}$ и распределение $\vec{p} = (p_1, \dots, p_m)$. Тогда средней длиной кодового слова $\varphi: A \rightarrow B^*$ называется величина:

$$l^\varphi = \sum_{i=1}^m p_i l_i \quad (2)$$

Определение 2.5 Алфавитного кодирование $\varphi: A \rightarrow B$ называется оптимальным, если φ однозначно декодируемо и при этом средняя длина l^φ минимальна.

Теорема 2.3

$$\frac{H(\vec{p})}{\log_2 D} \leq l^\varphi \leq 1 + \frac{H(\vec{p})}{\log_2 D} \quad (3)$$

, где $H(\vec{p}) = H(p(a_1), \dots, p(a_m)) = H(A) = \sum_{i=1}^m p_i \log_2 \frac{1}{p_i} = - \sum_{i=1}^m p_i \log_2 p_i$ - энтропия вероятностной схемы A

$$A = \begin{pmatrix} a_1 & a_2 & \dots & a_m \\ p(a_1) & p(a_2) & \dots & p(a_m) \end{pmatrix}$$

3. Помехоустойчивое кодирование

Как было написано ранее задача оптимального кодирования - максимально убрать избыточность кода, чтобы длина передаваемого сообщения была минимальной. Но при передаче в канале связи могут возникать помехи, а для того чтобы обнаружить и исправить ошибку возникшую в процессе передачи по каналу связи необходимо наоборот внести избыточность в код. В этом заключается суть помехоустойчивого кодирования: внести в код избыточность таким образом, чтобы при возникновении искажений, можно было исправить или по крайней мере обнаружить ошибки возникшие при передаче. При этом

желательно, чтобы скорость передачи по каналу связи снизилась как можно меньше, и алгоритмы кодирования/декодирования были оптимальными в смысле трудоемкости по времени и требуемой памяти.

Так же методы при выборе методов кодирования стоит учитывать особенности помех. Например в реальных каналах связи важно соотношение между временем передачи одного символа и средним временем воздействия случайной помехи. Если среднее время воздействия случайной помехи не больше времени передачи одного символа, то стоит выбирать кодирование позволяющее обнаружить и исправить ошибки отдельных символов, а если среднее время воздействия случайной помехи больше времени передачи 1 символа, то нужно такое кодирование, которое позволит корректировать целый набор символов (пакетов ошибок - последовательностей символов, внутри которых имеются ошибки). Так же следует учитывать как именно помехи влияют на передаваемое сообщение: например они могут изменять какой-то символ, удалять какой-то символ или добавлять новый. Так же методы кодирования разделяют на те в которых кодирование очередного символа или блока не зависит от того какой символ или блок кодировался до этого, и те в которых эта зависимость существует.

4. Блочный код

Рассмотрим класс блочных(блоковых) кодов.

Определение 4.1 (Источник: [1])

Пусть даны $k, n \in \mathbb{N}$, таковы, что $D^k \leq q^n$.

Блочным кодированием с входным алфавитом $B = \{b_1, \dots, b_D\}$, выходным алфавитом $X = \{x_1, \dots, x_q\}$, блоком информационных символов длины k и блоком кодовых символов длины n называется произвольное инъективное отображение $\varphi: B^k \rightarrow X^n$.

Множество $\mathcal{C} = \varphi(B^k) \subseteq X^n$ называется блочным кодом, а его элементы - кодовыми словами.

Т.е. при использовании блочного кодирования мы берем сообщение произвольной длины, делим его на не пересекающихся части по k символов. Потом каждую часть переводим в алфавит X (на практике часто $X = B$), и добавляем $n - k$ избыточных символов из алфавита X . Если длина исходного сообщения не кратна k , то к последнему блоку дописывается такое количество символов из B (каким способом дописывать оговорено заранее и известно как кодеру, так и декодеру) чтобы она стала кратной k .

Далее кодовые слова C последовательно, символ за символом, передаются по каналу связи, в котором могут подвергнуться влиянию помех. И в итоге на вход декодеру поочередно попадает последовательность кодовых слов $y_i^n = (y_{i_1}, \dots, y_{i_n}) \in Y^n$, которая в следствие влияния помех, может отличаться от последовательности $x_i^n = (x_{i_1}, \dots, x_{i_n})$ которая была отправлена. Далее задача декодера по последовательности y_i^n восстановить последовательность x_i^n , если произошли ошибки(искажения) исправить, если исправить не удастся, то делаем вывод о безуспешной передаче(в таких случаях обычно идет запрос на повторную передачу кодового слова).

(Чаще всего $B = X = Y = \{0, 1\}$)

Определение 4.2 (Источник: [1])

Расстояние Хемминга на множестве X^n называется отображение $\rho: X^n \times X^n \rightarrow \mathbb{N}_0$, задаваемое формулой

$$\rho(a^n, b^n) = \sum_{i=1}^n \chi(a_i \neq b_i) \quad (4)$$

где: $a^n \in X^n, b^n \in X^n, \chi(A)$ - индикатор события A

$$\chi(A) = \begin{cases} 0, & \text{если } A - \text{ложь} \\ 1, & \text{если } A - \text{истина} \end{cases}$$

Другими словами $\rho(a^n, b^n)$ это число координат, где $a_i \neq b_i$

Свойства 4.2.1 (Источник: [1])

1) (неотрицательность) $\rho(a^n, b^n) \geq 0$, причем $\rho(a^n, b^n) = 0 \Rightarrow a^n = b^n$

2) (симметричность) $\rho(a^n, b^n) = \rho(b^n, a^n)$

3) (неравенство треугольника) $\rho(a^n, b^n) + \rho(b^n, c^n) \geq \rho(a^n, c^n)$

Если в канал был передан вектор $x_i^n = (x_{i_1}, \dots, x_{i_n})$ и был получен вектор $y_i^n = (y_{i_1}, \dots, y_{i_n})$, то говорят, что произошло $\rho(x_i^n, y_i^n) = s$ ошибок.

Список использованных источников

- [1] Чечёта С.И. *Введение в дискретную теорию информации и кодирования*. МЦНМО, 2011, р. 224. ISBN: 978-5-94057-701-0.