

Лаба 5. Контроль и исправление ошибок.

Лабораторная работа состоит из трех частей:

1. Контроль ошибок (CRC - Cyclic Redundancy Checksum). Реализация принципа ARQ (Automatic Repeat reQuest)
2. Поиск и исправление ошибок (Error Correction Code, Hamming Code)
3. Маршрутизация

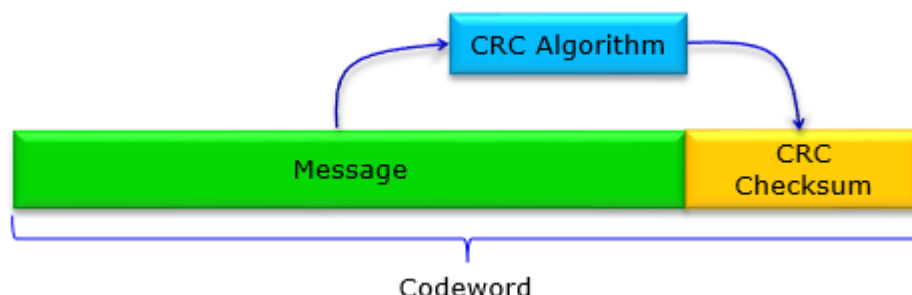
Часть 1. Контроль ошибок (CRC - Cyclic Redundancy Checksum)

CRC - технология проверки последовательности на ошибки. Вкратце, имеется исходное сообщение определенного размера, в которое необходимо добавить "избыточные" биты информации, которые помогут определить, была ли допущена ошибка в исходном сообщении при передаче данных.

Таким образом, имеется:

1. Сообщение
2. CRC алгоритм
3. "избыточные" биты внедренные в сообщение

Все вместе называется Codeword:

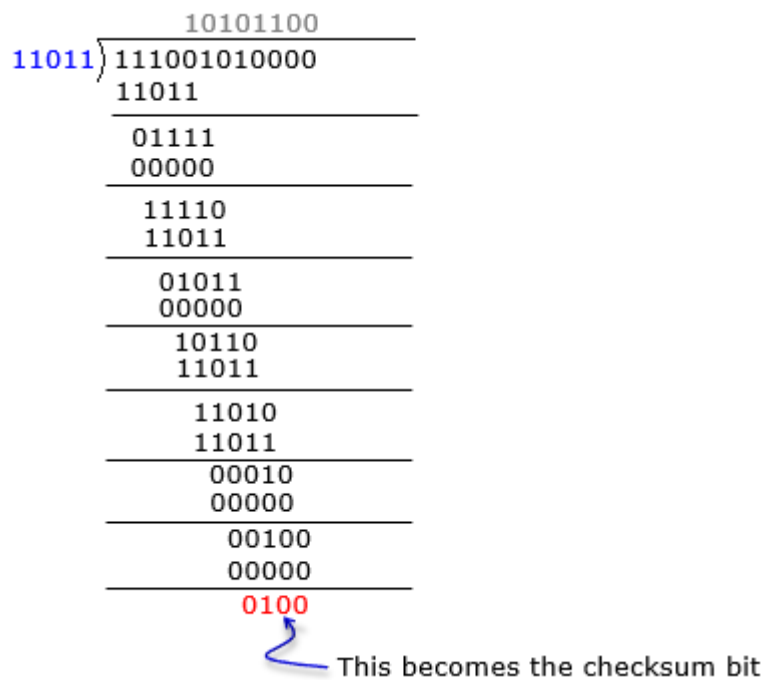


В нашем случае, будем использовать готовый генерирующий последовательность полином для подсчета контрольной суммы:

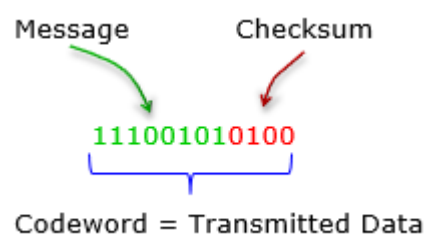
Generator Polynomial = $X^4 + X^3 + X + 1$

Message Data = 11100101

Алгоритм достаточно просто, необходимо делить генерирующий полином делить (применив операцию XOR) на исходной сообщении:



В итоге, получаем:



Задание Части 1

1. Взяв за основу лабораторную работу №4, проверить на наличие ошибок при передачи файла (изображение\текстовый файл\и т.д.).
2. Выбрать размер "пакета" (исходного сообщения) [от 8 до 32 бит].
3. Использовать полином из описания CRC принципа.
4. Добавить "избыточность".

5. Отправить от сервера к клиенту.
6. При получении проверить на ошибки.
7. Если присутствуют ошибки, реализовать ARQ процесс.
8. Построить график зависимости среднего количества переотправок пакета от значения SNR, где SNR = [start=-5; end=3; step=1]
9. Оценить % избыточности информации.

Часть 2. Поиск и исправление ошибок

В данном разделе рассмотрим один из простейших алгоритмов поиска и исправления ошибок, код Хэмминга.

Bit Position	1	2	3	4	5	6	7	8	9	10	11
Encoded data bits	p1	p2	d1	p4	d2	d3	d4	p8	d5	d6	d7
Parity bit coverage	p1	x		x		x		x		x	
	p2		x	x			x	x			x
	p4				x	x	x	x			
	p8								x	x	x

Задание Части 2

1. Поверх CRC добавить код Хэмминга.
2. Если присутствуют ошибки, реализовать ARQ процесс.
3. Построить график зависимости среднего количества переотправок пакета от значения SNR, где SNR = [start=-5; end=3; step=1]
4. Оценить % избыточности.

Часть 3. Маршрутизация

1. Необходимо добавить еще 2 клиентских приложения (+ 2 терминала, итого 3 клиентских, 1 серверный)
2. Разработать "протокол" передачи данных, где все данные будут проходить через серверное приложение. Протокол должен иметь возможность передавать каждый пакет конкретному пользователю.

3. Считать связь между всеми узлами - беспроводную.
4. CRC + Hamming присутствуют в каждом пакете.