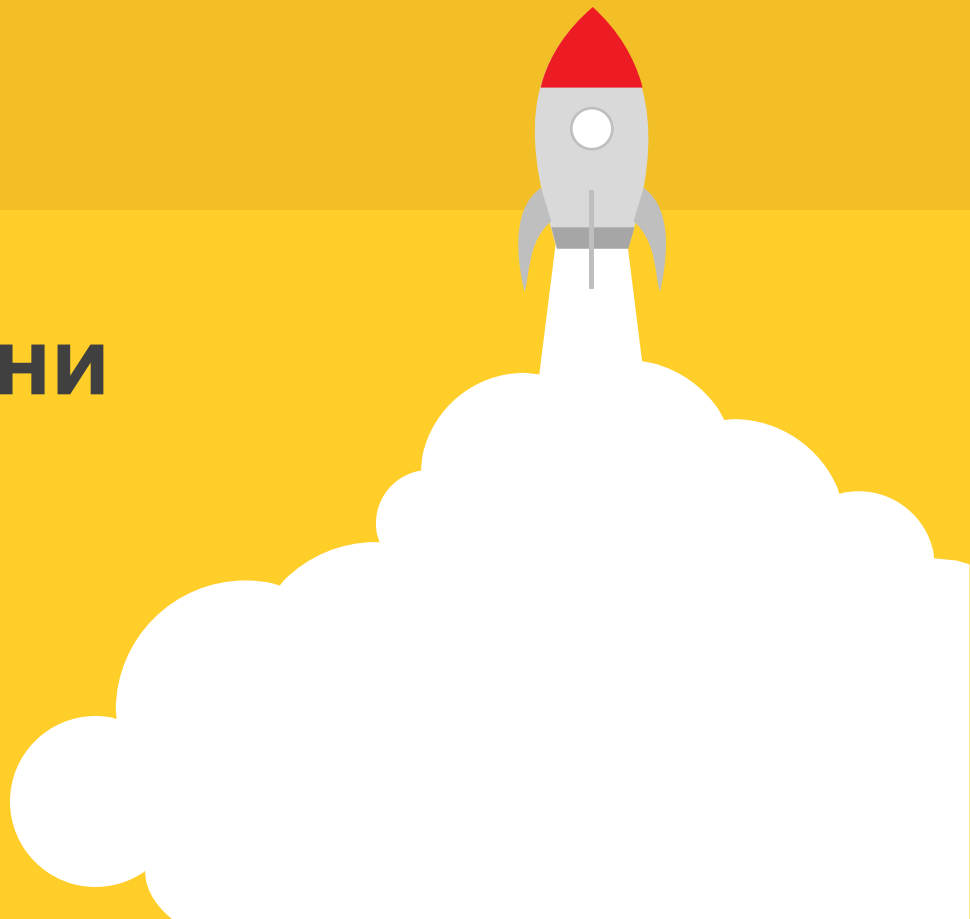
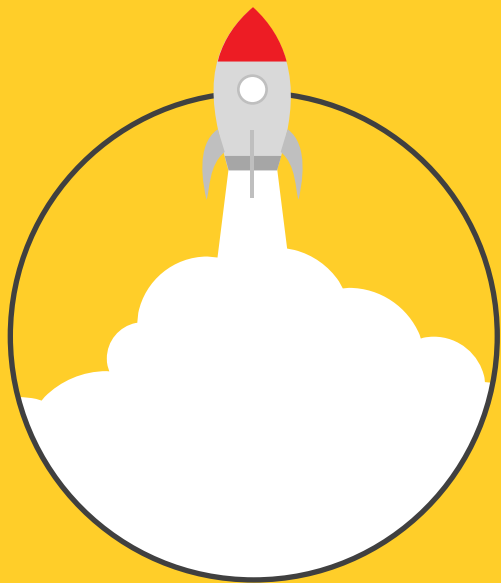


Предаване на данни и компютърни комуникации

Лекция 6





Грешки при предаване на цифрови данни

Лекция 6



Лекция 6

Грешки при предаване на цифрови данни

Видове грешки при последователно предаване.
Откриване на грешки чрез проверка за еднаквост.
Циклична проверка за информационен излишък.
Корекция на грешките.

Цел на обучението

След като преминете през Лекция 6, вие ще можете да:

01

Разбирате основния механизъм за използване на кодове за откриване на грешки.

02

Представяте в обобщение смисъла на контролната сума в IPv4 заглавната информация.

03

Разбирате работата на циклична проверка за информационен излишък. Определяте разстояние на Хеминг.

04

Особености на принципите за коригиране на грешки с помощта на блоков код.



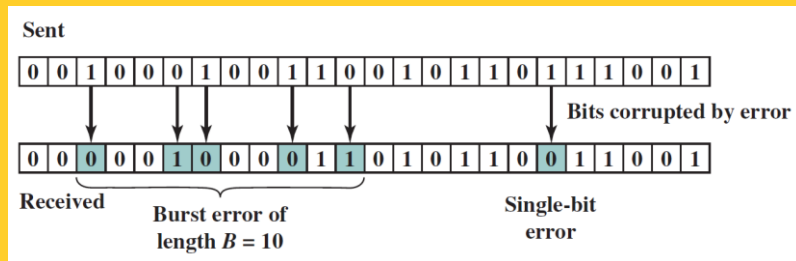
1. Грешки при предаване на цифрови сигнали

1.1. Видове грешки



Грешки при предаване на информация

- Сигналът представлява кодирани данни в двоичен вид – грешките променят смисъла на данните:
 - Грешка има когато сигналът, който е получен е различен от сигнала, който е изпратен (дори само с един бит)
 - Предава се двоична „0“, а се получава двоична „1“;
 - Предава се двоична „1“, а се получава двоична „0“.
 - Грешките биват два вида – **битови** и **динамични**.
- Причина за грешките е шум по време на предаване
- Използват се методи за **откриване на грешки** за да се провери дали приемникът е получил правилни данни или не.
- Използват се механизми за **корекция на грешки**, случай че бъде открита такава.



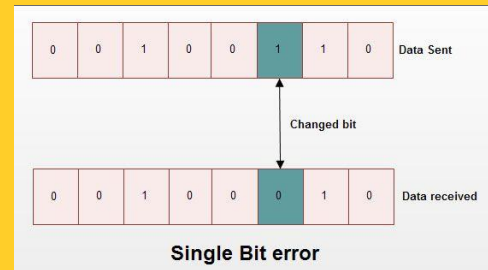
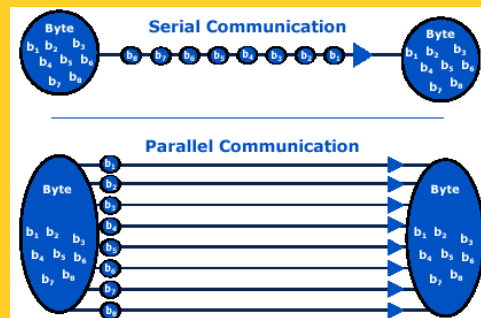
1. Грешки при предаване на цифрови сигнали

1.2. Битова грешка



Битова грешка

- **Битова грешка** – променя двоична „0“ в двоична „1“ или обратно – двоична „1“ в двоична „0“.
- Битова грешка може да се появи в паралелни връзки, където всеки бит се предава по отделни жици.
- По-малко вероятно е да се появи битова грешка в серийни връзки.
 - *Последователна (Серийна) комуникация* – битовете се предават последователно един по един в споделен канал.
 - *Паралелна комуникация* – няколко бита (обикновено 8) се предават едновременно по различни канали в един кабел. Предава се само синхронно.
- *Пример:* Импулсно смущение от 0,01 с при предаване със скорост 1200 bps променя 12 бита.



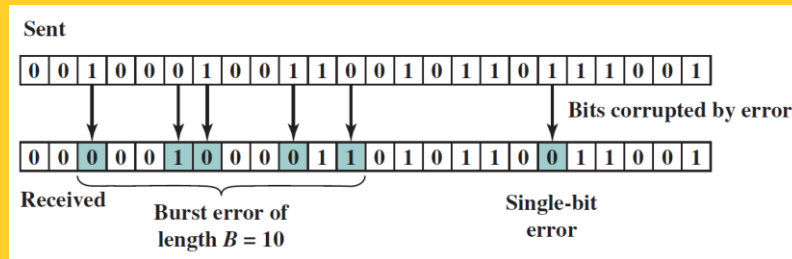
1. Грешки при предаване на цифрови сигнали

1.3. Динамична грешка



Динамична грешка (Burst Error)

- **Динамична грешка** – променя в два или повече бита в предаваните данни.
- **Дефиниция** според IEEE Std 100 и ITU-T Recommendation Q.9:
 - Група от битовете, в която два последователни грешни бита винаги са разделени с по-малко от x наброй правилни битовете. Последният сгрешен бит в поредицата на динамичната грешка и първия сгрешен след него са съответно разделени с x наброй или повече правилни битовете.
- **Причина за динамични грешки** могат да бъдат импулсен шум или затихване. Този тип грешки са по-силно изразено при по-високи скорости на предаване.
- **Пример:** Импулсен шум с дължина $1\ \mu\text{s}$ за поток от данни със скорост 10 Mbps причинява 10-битова динамична грешка. При 100 Mbps поток – 100-битова грешка.



2. Механизми за откриване на грешки при предаване

2.1 Основни понятия



Откриване на грешки

- Независимо от дизайна на преносната среда, грешки при предаването винаги настъпват. Не се търсят механизми за елиминиране на възникването на грешки, а по-скоро за тяхното откриване и коригиране.
- **Рамка** – поредица от една или повече последователности от битове, които се предават.
- Грешки при предаване – откриват се от приемника след декодиране на получените информационни данни.
- Методите за откриване на грешки се основават на т. нар. **информационен излишък**.
 - Концепция – в предавателя допълнителната информация от малка група битове в края на всеки блок;
 - Приемника – проверява цялата входяща информация;
 - Проверка за достоверност – при потвърждение излишната част от битове се премахва

2. Механизми за откриване на грешки при предаване

2.1 Основни понятия



Информационен излишък

- За предавателя – съществуват вероятности за грешки, променящи на един или повече битовете във всяка рамка:

P_b - вероятността получения бит да е сгрешен (Bit error rate BER)

P_1 - вероятността получената рамка да не съдържа грешки

P_2 - вероятността при използване на алгоритъм за откриване на грешка да пристигне рамка с една или повече неоткрити грешки

P_3 - вероятността при използване на алгоритъм за откриване на грешка да пристигне рамка с една или повече открити грешки и нула неоткрити.

- За всяка рамка с F брой битовете, при $P_3=0$ и P_b - const.:

$$P_1 = (1 - P_b)^F$$

$$P_2 = 1 - P_1$$

- При увеличаване на вероятността P_b вероятността P_1 намалява;
- При увеличаване на дължината на рамката (F), вероятността P_1 намалява.

2. Механизми за откриване на грешки при предаване

2.1 Основни понятия



Информационен излишък

- *Пример: Цифрова мрежа с интегрирани услуги ISDN*
 - Основна задача в 64-kbps канал: до 90% от едноминутни наблюдавани време интервали: **BER < 10^{-6}**
 - При 1000 бита рамка:
За 1 ден се предават до $5,529 \times 10^6$ бита – вероятността за открити BER - $P_2 = 1/(5,529 \times 10^6) = 0,18 \times 10^{-6}$
 - За **BER $P_b = 10^{-6}$**
Вероятността за приемане на рамка без битови грешки:

$$P_1 = (0,999999) \times 1000 = 0,999$$

Вероятността за открити BER:

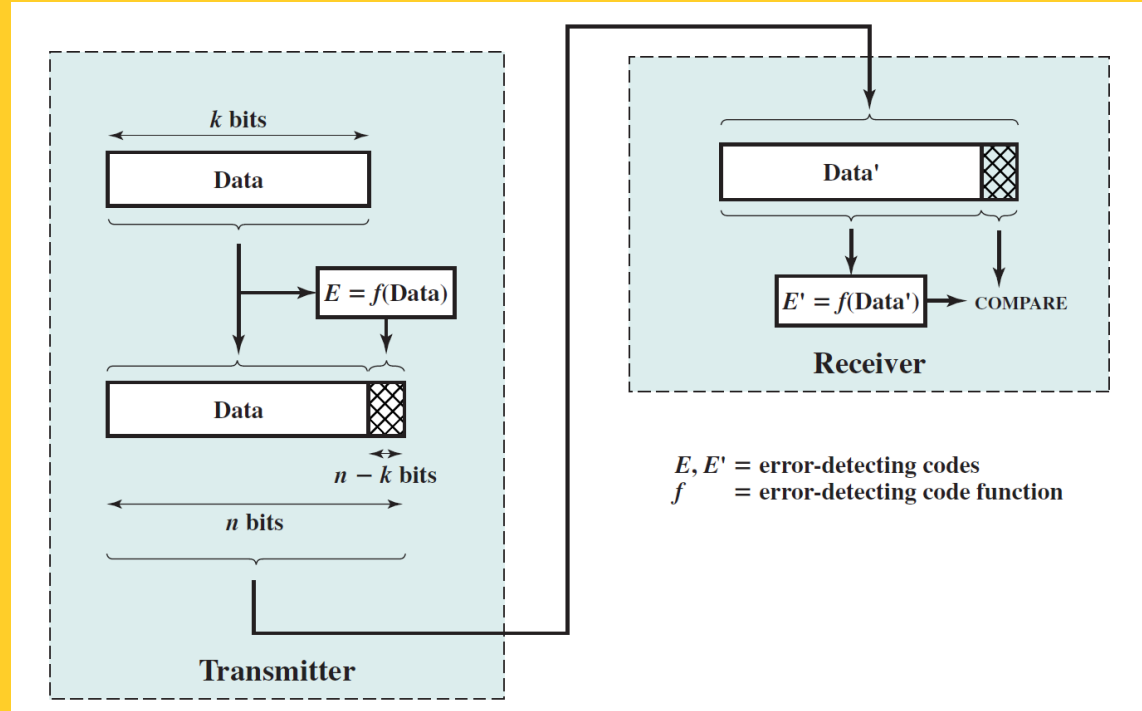
$$P_2 = 10^{-3}, \text{ т.е. три порядъка по-голяма.}$$

2. Механизми за откриване на грешки при предаване

2.1 Основни понятия



Процес на откриване на грешки



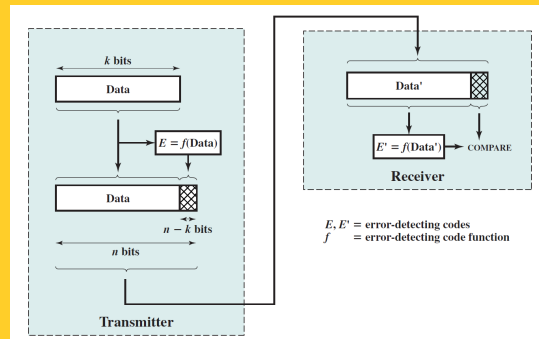
2. Механизми за откриване на грешки при предаване

2.1 Основни понятия



Процес на откриване на грешки

- Предавател:
 - Допълнителни $(n - k) < k$ бита – формират **кодът за откриване на грешки E** , като функция на предадените бита;
 - Изчисляват се **контролни бита**, които се прибавят към данните D .
- Приемник:
 - Разделя пристигащата рамка на k бита данни и $(n - k)$ бита код за откриване на грешки E ;
 - **Сравнява** приетите $(n - k)$ бита и получения контролен код $E = f(D')$
 - **Открива** грешки при несъответствие.



2. Механизми за откриване на грешки при предаване

2.2. Методи за откриване на грешки



Методи за проверка на информационен излишък

1. Проверка за еднаквост

- Вертикален контрол за излишък (VRC)
- Проверка по четност и нечетност (LRC)

2. Цикличен контрол

- Използва се контролна сума FCS (Frame Check Sequence)
- VRC, LRC и CRC – на канално ниво (OSI модел)
- FCS на по-горни OSI слоеве

2. Механизми за откриване на грешки при предаване

2.2. Методи за откриване на грешки



Проверка за еднаквост (Parity Check)

- **Проверка по четност (Parity bit)** - най-използвания метод за определяне на това дали двоичната информация е претърпяла промени по време на предаване.
- *Към всеки байт (8-бита - рамка), който се предава по мрежата се добавя един бит (parity bit), чиято стойност се определя от:*
 - Протокола – дали се прави проверка за четност на „0” или „1”
 - Бита за проверка по четност има стойност „0” или „1” в зависимост дали броя на „0” (или „1”) е четен или нечетен.



2. Механизми за откриване на грешки при предаване

2.2. Методи за откриване на грешки



Проверка за еднаквост (Parity Check)

- **Пример:** Предавателя изпраща съобщение, което съдържа думата *WORLD*
- **Предавателя** представя думата в ASCII код:

W	O	R	L	D
1010111	1001111	1010010	1001100	1000100
1 1010111	1 1001111	1 1010010	1 1001100	0 1000100

- Първите 4 символа (“W”, “O”, “R” и “L”) съдържат нечетен брой „1“, затова бита за проверка по четност има стойност **1**.
- Последния символ („D“) съдържа четен брой „1“, затова бита за проверка по четност има стойност **0**.
- Приемника – прави същата проверка. Ако бита за проверка по четност, който приемника получава е равен на бита, който е записан от предавателя се приема, че рамката е получена без грешки.

2. Механизми за откриване на грешки при предаване

2.2. Методи за откриване на грешки



Проверка за еднаквост (Parity Check)

- Недостатък... ?

2. Механизми за откриване на грешки при предаване

2.2. Методи за откриване на грешки



Проверка за еднаквост (Parity Check)

- Недостатък...
 - Проверката по четност с един бит не засича четен брой грешки!

Предавател	1	0	0	1	1	0	1	1
Приемник	1	1	1	1	1	0	1	1

2. Механизми за откриване на грешки при предаване

2.2. Методи за откриване на грешки



Проверка за еднаквост (Parity Check)

- Проверка по четност (Parity bit) – неефективен метод, заради основния недостатък!
- Проверка за нарушаване на четност/нечетност (**двумерна проверка за еднаквост – two-dimensional parity check**) – решава проблема на проверката по четност.
 - Битовите в рамката, която се проверява по четност се подреждат в двумерен масив. По този начин се открива точната позиция на сгрешения бит.
 - Един бит се проверява два пъти по четност.

			Row parity →
			↑
			r_1
			r_2
			r_i
Column parity ↓	$b_{1,1}$	• • •	$b_{1,j}$
	$b_{2,1}$	• • •	$b_{2,j}$
	$b_{i,1}$	• • •	$b_{i,j}$
	c_1	• • •	c_j
			p

(a) Parity calculation

2. Механизми за откриване на грешки при предаване

2.2. Методи за откриване на грешки



Проверка за еднаквост (Parity Check)

- **Пример:** Изпращат се 20 бита данни; подреждат се в масив от по 5 бита в 4 реда; за всеки ред и всяка колона се прави проверка по четност.
 - Ако четен брой грешки има в ред – те се засичат от проверката по колони;
 - Ако четен брой грешки има в колона – те се засичат от проверката по редове.

0	1	1	1	0	1
0	1	1	1	0	1
0	1	0	0	0	1
0	1	0	1	1	1
0	0	0	1	1	0

0	1	1	1	0	1
0	0	1	1	0	1
0	1	0	0	0	1
0	1	0	1	1	1
0	0	0	1	1	0

Row parity error

Column parity error

2. Механизми за откриване на грешки при предаване

2.2. Методи за откриване на грешки



Проверка за еднаквост (Parity Check)

- **Проблем:** четен брой грешки в два реда в една и съща позиция.
 - Тези грешки не могат да се засекат! Не могат да се поправят!

0	1	1	1	1	1	0	1
0	0	1	1	0	1	1	0
0	0	1	1	0	0	1	1
0	0	0	0	0	0	0	0
1	0	1	1	1	1	1	0
1	1	0	0	0	1	1	0

(d) Uncorrectable error pattern

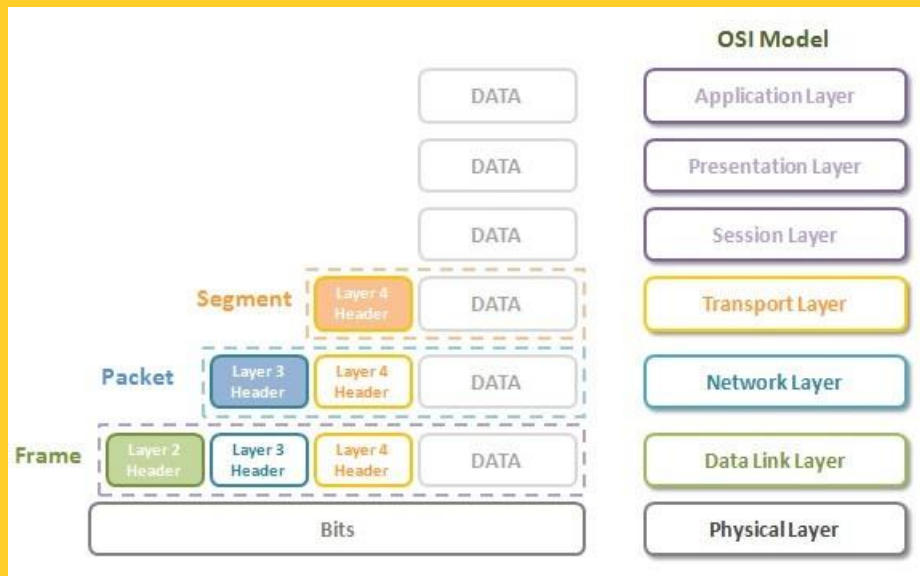
2. Механизми за откриване на грешки при предаване

2.2. Методи за откриване на грешки



Цикличен контрол на мрежово ниво (*Internet Checksum*)

- *OSI Модел и служебни данни (headers)*



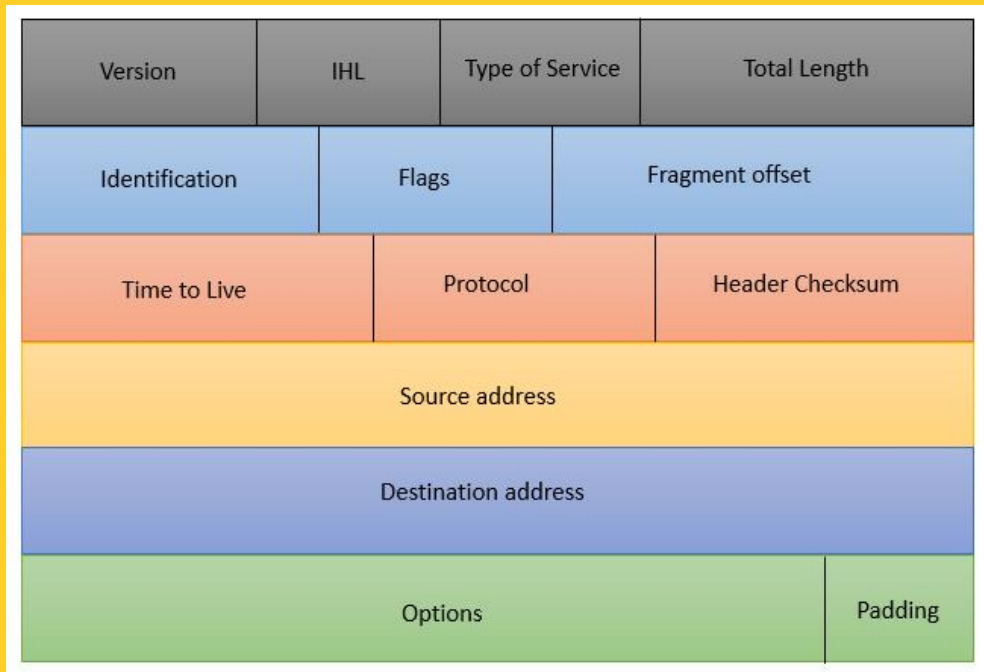
2. Механизми за откриване на грешки при предаване

2.2. Методи за откриване на грешки



Цикличен контрол на мрежово ниво (*Internet Checksum*)

- *IPv4 заглавна информация*



2. Механизми за откриване на грешки при предаване

2.2. Методи за откриване на грешки



Контрол за грешки на мрежово ниво (*Internet Checksum*)

- Проверка за грешки при предаване на мрежово ниво предоставя отличен контрол в сравнение с проверките по четност.
- По-малко ефективна е от CRC (Cyclic Redundancy Check), но е сравнително проста за реализиране – операции като събиране и сравняване.
- Предлага заключителен контрол – предполага се, че на ниско ниво се извършват по-сложни операции за откриване и поправяне на грешки, и затова цикличния контрол просто допълва проверката от край до край за грешки.
- Контрол за грешки на мрежово ниво се основава на правилото за обръщане на единици в нули и нули в единици. Нарича се още обърнат код (ones' complement). Операцията се изпълнява като следва:
 - Двете числа се третират като двоични цели числа без знак и се събират;
 - Ако има пренасяне се добавя 1 към сумата.

0011	1101
<u>+1100</u>	<u>+1011</u>
1111	11000
	<u>+ 1</u>
	1001

2. Механизми за откриване на грешки при предаване

2.2. Методи за откриване на грешки



Контрол за грешки на мрежово ниво (*Internet Checksum*)

- Пример:** Нека заглавната информация съдържа 10 октета, като последните два октета представят сумата за проверка на грешки:

00 01 F2 03 F4 F5 F6 F7 00 00

Partial sum	$\begin{array}{r} 0001 \\ F203 \\ \hline F204 \end{array}$
Partial sum	$\begin{array}{r} F204 \\ F4F5 \\ \hline 1E6F9 \end{array}$
Carry	$\begin{array}{r} E6F9 \\ 1 \\ \hline E6FA \end{array}$
Partial sum	$\begin{array}{r} E6FA \\ F6F7 \\ \hline 1DDF1 \end{array}$
Carry	$\begin{array}{r} DDF1 \\ 1 \\ \hline DDF2 \end{array}$
Ones complement of the result	220D

(a) Checksum calculation by sender

Partial sum	$\begin{array}{r} 0001 \\ F203 \\ \hline F204 \end{array}$
Partial sum	$\begin{array}{r} F204 \\ F4F5 \\ \hline 1E6F9 \end{array}$
Carry	$\begin{array}{r} E6F9 \\ 1 \\ \hline E6FA \end{array}$
Partial sum	$\begin{array}{r} E6FA \\ F6F7 \\ \hline 1DDF1 \end{array}$
Carry	$\begin{array}{r} DDF1 \\ 1 \\ \hline DDF2 \end{array}$
Partial sum	$\begin{array}{r} DDF2 \\ 220D \\ \hline FFFF \end{array}$

(b) Checksum verification by receiver

2. Механизми за откриване на грешки при предаване

2.2. Методи за откриване на грешки



Контрол за грешки на мрежово ниво (*Internet Checksum*)

- Пример:** Нека заглавната информация съдържа 10 октета, като последните два октета представят сумата за проверка на грешки:

00 01 F2 03 F4 F5 F6 F7 **22 0D**

Partial sum	$\begin{array}{r} 0001 \\ F203 \\ \hline F204 \end{array}$
Partial sum	$\begin{array}{r} F204 \\ F4F5 \\ \hline 1E6F9 \end{array}$
Carry	$\begin{array}{r} E6F9 \\ 1 \\ \hline E6FA \end{array}$
Partial sum	$\begin{array}{r} E6FA \\ F6F7 \\ \hline 1DDF1 \end{array}$
Carry	$\begin{array}{r} DDF1 \\ 1 \\ \hline DDF2 \end{array}$
Ones complement of the result	220D

(a) Checksum calculation by sender

Partial sum	$\begin{array}{r} 0001 \\ F203 \\ \hline F204 \end{array}$
Partial sum	$\begin{array}{r} F204 \\ F4F5 \\ \hline 1E6F9 \end{array}$
Carry	$\begin{array}{r} E6F9 \\ 1 \\ \hline E6FA \end{array}$
Partial sum	$\begin{array}{r} E6FA \\ F6F7 \\ \hline 1DDF1 \end{array}$
Carry	$\begin{array}{r} DDF1 \\ 1 \\ \hline DDF2 \end{array}$
Partial sum	$\begin{array}{r} DDF2 \\ 220D \\ \hline FFFF \end{array}$

(b) Checksum verification by receiver

2. Механизми за откриване на грешки при предаване

2.2. Методи за откриване на грешки



Циклична проверка за информационен излишък (CRC)

- Един от най-често използваните механизми за откриване на грешки; най-ефективен и мощен – ***Cyclic Redundancy Control***.
- Основана се на проверка по четност чрез двоично деление вместо събиране на битове:
 - k – битов блок от данни;
 - предавателя – генерира контролна сума FCS с дължина $n - k$ бита
 - приемника – дели пристигащата рамка на FCS, при резултат без остатък **няма грешки при предаване**
- Необходими свойства на CRC проверката:
 - рамката да се дели точно на даден делител;
 - остатъкът да им един бит по-малко от делителя
- Методи за генериране на циклична проверка:
 - Сума по модул 2
 - Полиноми
 - Цифрова логика

2. Механизми за откриване на грешки при предаване

2.2. Методи за откриване на грешки



Циклична проверка за информационен излишък (CRC)

Сума по модул 2

- Използва двоично събиране без пренос, което е същото като изключващо „ИЛИ“ (XOR):

X1	X2	Y
0	0	0
0	1	1
1	0	1
1	1	0

- Примери:

$$\begin{array}{r} 1111 \\ +1010 \\ \hline 0101 \end{array}$$

$$\begin{array}{r} 1111 \\ -0101 \\ \hline 1010 \end{array}$$

$$\begin{array}{r} 11001 \\ \times 11 \\ \hline 11001 \\ 11001 \\ \hline 101011 \end{array}$$

2. Механизми за откриване на грешки при предаване

2.2. Методи за откриване на грешки



Циклична проверка за информационен излишък (CRC)

- Дефинирани променливи:

$T = n$ – битова рамка за предаване

$D = k$ – битов блок от данни, или съобщение, първите k бита от T

$F = (n - k)$ – битова контролна сума (FCS), последните $(n - k)$ бита от T

P – делител по шаблон $n - k + 1$ бита

- Основна задача при сумиране по модул 2 е да има делене **T/P без остатък**:

$$T = 2^{n-k}D + F$$

- В *предавателя* ефектът от делението съответства на изместване на ляво с $(n - k)$ бита и допълване на резултата с нули. Това води до частно Q и остатък R :

$$\frac{2^{n-k}D}{P} = Q + \frac{R}{P}$$

- При деленето по модул 2 винаги остатъкът R е поне с един бит по-малко от делителя P , като това се използва за контролната сума FCS:

$$T = 2^{n-k}D + R$$

2. Механизми за откриване на грешки при предаване

2.2. Методи за откриване на грешки



Циклична проверка за информационен излишък (CRC)

- В *приемника* деленето се извършва с контролната сума:

$$\frac{T}{P} = \frac{2^{n-k}D + R}{P} = \frac{2^{n-k}D}{P} + \frac{R}{P}$$

- Заместваме израза в предавателя и получаваме:

$$\frac{T}{P} = Q + \frac{R}{P} + \frac{R}{P}$$

- Всяко двоично число събрано със себе си по модул 2 е нула, а ***T*** е точно делимо на ***P***:

$$\frac{T}{P} = Q + \frac{R + R}{P} = Q$$

- Определяне на грешките: в n – битова рамка се представят като поле от **1** за всеки грешен бит – приета рамка ***T_r***, предадена рамка ***T***; шаблон за грешките ***E***; \oplus - изключващо ИЛИ:

$$T_r = T \oplus E$$

2. Механизми за откриване на грешки при предаване

2.2. Методи за откриване на грешки



Циклична проверка за информационен излишък (CRC)

Пример:

- **Задание:**

Съобщение ***D*** ($k = 10$ бита): 1010001101

Шаблон ***P*** (6 бита) 110101

Дължина на рамката за предаване ***T*** (n бита) = 15 бита

Контролна сума ***R*** ($n - k$) = 5 бита; **R=?**

- Намира се произведението $2^5 \times D = 101000110100000$
- Произведението $2^5 \times D$ се дели на шаблона ***P*** – получават се частното ***Q*** и остатъка ***R***.

2.2. Методи за откриване на грешки



Пример:

- Произведението $2^5 \times D$ се дели на шаблона P – получават се частното Q и остатъка R .

$P \rightarrow$

1	1	0	1	0	1		1	0	1	0	0	1	0	1	1	0	$\leftarrow Q$
							1	0	1	0	0	0	1	0	0	0	$\leftarrow 2^{n-k}D$
							<u>1</u>	<u>1</u>	<u>0</u>	<u>1</u>	<u>0</u>	<u>1</u>					
							1	1	1	0	1	1					
							<u>1</u>	<u>1</u>	<u>0</u>	<u>1</u>	<u>0</u>	<u>1</u>					
							1	1	1	0	1	0					
							<u>1</u>	<u>1</u>	<u>0</u>	<u>1</u>	<u>0</u>	<u>1</u>					
							1	1	1	1	1	0					
							<u>1</u>	<u>1</u>	<u>0</u>	<u>1</u>	<u>0</u>	<u>1</u>					
							1	0	1	1	0	0					
							<u>1</u>	<u>1</u>	<u>0</u>	<u>1</u>	<u>0</u>	<u>1</u>					
							1	1	0	0	0	1	0				
							1	1	0	1	0	1					
							0	1	1	1	0		$\leftarrow R$				

2. Механизми за откриване на грешки при предаване

2.2. Методи за откриване на грешки



Циклична проверка за информационен излишък (CRC)

Полиноми

- Цикличната проверка може да се представи и като полиноми на променлива X , с двоични коефициенти. Коефициентите съответстват на битовете в двоичното число.
 - Например, ако $D = 110011$, а $P = 11001$, тогава имаме следните полиноми:
$$D(X) = X^5 + X^4 + X + 1$$
$$P(X) = X^4 + X^3 + 1$$
- Аритметичната операция отново е сума по модул 2. Тогава цикличната проверка за информационен излишък се описва с изрази:

$$\frac{X^{n-k}D(X)}{P(X)} = Q(X) + \frac{R(X)}{P(X)}$$

$$T(x) = X^{n-k}D(X) + R(X)$$

2. Механизми за откриване на грешки при предаване

2.2. Методи за откриване на грешки



Циклична проверка за информационен излишък (CRC)

Пример (същия):

- Задание:

Съобщение D ($k = 10$ бита): 1010001101

$$D(X) = X^9 + X^7 + X^3 + X^2 + 1$$

Шаблон P (6 бита): 110101

$$P(X) = X^5 + X^4 + X^2 + 1$$

Дължина на рамката за предаване T (n бита) = 15 бита

Контролна сума R ($n - k$) = 5 бита; $R=?$

- Контролната сума R се получава 01110, което съответства на:

$$R(X) = X^3 + X^2 + X$$

2. Механизми за откриване на грешки при предаване

2.2. Методи за откриване на грешки



Циклична проверка за информационен излишък (CRC)

Полиноми

- Грешката $E(X)$ ще бъде неоткриваема, ако се дели на $P(X)$
- Всички от следните грешки не се делят на $P(X)$ (подходящо избран делител) и следователно могат да се открият:
 - Всички еднобитови грешки, ако $P(X)$ има повече от един ненулев член;
 - Всички двубитови грешки, когато $P(X)$ има следната форма:
$$P(X) = q(X)$$
$$P(X) = (X + 1)q(X)$$
$$q(X) - \text{примитивен полином}$$
- Всички грешки с нечетен брой, ако $P(X)$ съдържа фактор $X+1$
- Всяка пакетна грешка, при която дължината на пакета е по-малка или равна на $n-k$, т.е. по-малко или равно на дължината на FCS.
- Част от поредица грешки с дължина $n-k+1$, при част = $1 - 2^{-(n-k-1)}$
- Част от поредица грешки с дължина по-голяма от $n-k+1$, при част = $1 - 2^{-(n-k)}$

2. Механизми за откриване на грешки при предаване

2.2. Методи за откриване на грешки



Циклична проверка за информационен излишък (CRC)

Полиноми

- Използват се стандартни полиноми за шаблони $P(X)$:
CRC-12 $= X^{12} + X^{11} + X^3 + X^2 + X + 1 = (X + 1)(X^{11} + X^2 + 1)$
CRC-ANSI $= X^{16} + X^{15} + X^2 + 1 = (X + 1)(X^{15} + X + 1)$
CRC-CCITT $= X^{16} + X^{12} + X^5 + 1 = (X + 1)(X^{15} + X^{14} + X^{13} + X^{12} + X^4 + X^3 + X^2 + X + 1)$
IEEE-802 $= X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$

2. Механизми за откриване на грешки при предаване

2.2. Методи за откриване на грешки



Циклична проверка за информационен излишък (CRC)

Цифрова логика

- CRC генераторът се представя чрез схема съдържаща изключващо ИЛИ и преместващ регистър
 - *Преместващ регистър* – последователност от 1 bit запомнящи клетки, които се преместват последователно чрез тактуване;
 - Регистърът съдържа $(n-k)$ бита равни на дължината на контролната сума FCS;
 - повече от $(n-k)$ елементи за изключващо ИЛИ;
 - Наличието/липсата на суматор съответства на наличието/липсата на полином в шаблона $P(X)$ (изключение полиномните членове 1 и X^{n-k})

2. Механизми за откриване на грешки при предаване

2.2. Методи за откриване на грешки



Циклична проверка за информационен излишък (CRC)

Цифрова логика (Пример)

- Преместващ регистър за делене с полинома $P(x) = X^5 + X^4 + X^2 + 1$
- Данни **D** = 1010001101

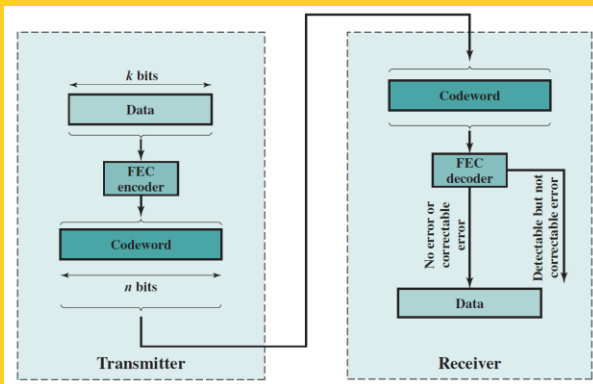
стъпка	C_4	C_3	C_2	C_1	C_0	$C_4 \oplus C_3 \oplus I$	$C_4 \oplus C_1 \oplus I$	$C_4 \oplus I$	изход
иниц.	0	0	0	0	0	1	1	1	1
1	1	0	1	0	1	1	1	1	0
2	1	1	1	1	1	1	1	0	1
3	1	1	1	1	0	0	0	1	0
4	0	1	0	0	1	1	0	0	0
5	1	0	0	1	0	1	0	1	0
6	1	0	0	0	1	0	0	0	1
7	0	0	0	1	0	1	0	1	1
8	1	0	0	0	1	1	1	1	0
9	1	0	1	1	1	0	1	0	1
10	0	1	1	1	0				

3. Корекция на грешките



Корекция на грешки

- Някои протоколи на канално ниво (HDLC), мрежово (IP, ICMP) или транспортно (TCP) ниво поправят грешките при предаване, чрез **повторно изпращане** на сгрешените кадри.
- За безжичните технологии повторното предаване на сгрешени рамки е неефективен подход поради следните причини:
 - BER в безжичните връзки е относително висока, което би довело до множество повторни предавания;
 - В някои случаи, например сателитни комуникации, закъснението при разпространението на сигнала е много по-голямо от предаването на единична рамка.



3. Корекция на грешките



Корекция на грешки

- Четири възможни случая при FEC декодирането:
 1. **Няма битова грешка** – на изхода на декодерът се получава оригиналния блок от данни;
 2. **Откриване и коригиране на образци от грешки** – декодерът има възможност да възстанови данните до оригиналния им вид, когато грешките имат шаблонен характер;
 3. **Откриване на грешки с невъзможност за поправка** – декодерът засича грешки, но не може да ги отстрани. В най-общия случай декодерът докладва за грешки.
 4. **Грешки, които не могат да бъдат открити** – в редки случаи декодерът не намира различия при съпоставянето на оригиналния и приетия блок от данни.

3. Корекция на грешките



Корекция на грешки

- Как е възможно поправянето на грешки?
 - Чрез добавяне на достатъчен излишък към съобщението за предаване.
 - Излишъкът дава възможност на получателя да определи какво е било оригиналното съобщение, дори при определено ниво на грешка.
- Процес на коригиране на грешките:
 - Използва се FEC (Forwarding Error Correction) кодиращо и декодиращо устройство, което:
 - Извършва сравнение при приемане на k -битов блок от данни с n -битова кодова дума ($n > k$)

3. Корекция на грешките



Корекция на грешки

- Принцип на блоково кодиране за корекция на грешки:
 - Използва се **разстояние на Хеминг** $d(v_1 v_2)$ между две n -битови двоични последователности v_1 и v_2 - брой на различаващите се компоненти.
 - *Пример:* $v_1 = 011011$, $v_2 = 110001$

$$d(v_1 v_2) = 3$$

- Блоково кодиране за корекция на грешки:
 - Нека предаваните блокове от данни са с дължина k -бита.
 - Вместо да се предава всеки k -битов блок по отделно, той се отбелязва като n -битова кодова дума.
 - *Пример:* $k = 2$ бита, $n = 5$ бита

Data Block	Codeword
00	00000
01	00111
10	11001
11	11110

3. Корекция на грешките



Корекция на грешки

- Блоково кодиране за корекция на грешки:
 - Нека предаваните блокове от данни са с дължина k -бита.
 - Вместо да се предава всеки k -битов блок по отделно, той се отбелязва като n -битова кодова дума.
 - *Пример:* $k = 2$ бита, $n = 5$ бита
 - Приемника получава кодова дума 00100, която е невалидна, защото не съответства на никоя от дефинираните.
 - Може ли да се направи корекция?
 - 00100 може да бъде 00000
 - 00100 може да бъде 00111
 - 00100 може да бъде 11001
 - 00100 може да бъде 11110

Data Block	Codeword
00	00000
01	00111
10	11001
11	11110

3. Корекция на грешките



Корекция на грешки

- Блоково кодиране за корекция на грешки:
 - Нека предаваните блокове от данни са с дължина k -бита.
 - Вместо да се предава всеки k -битов блок по отделно, той се отбелязва като n -битова кодова дума.
 - *Пример:* $k = 2$ бита, $n = 5$ бита
 - Приемника получава кодова дума 00100, която е невалидна, защото не съответства на никоя от дефинираните.
 - Може ли да се направи корекция?

00100 може да бъде 00000

00100 може да бъде 00111

00100 може да бъде 11001

00100 може да бъде 11110

Data Block	Codeword
00	00000
01	00111
10	11001
11	11110

3. Корекция на грешките



Корекция на грешки

- Блоково кодиране за корекция на грешки:
 - Нека предаваните блокове от данни са с дължина k -бита.
 - Вместо да се предава всеки k -битов блок по отделно, той се отбелязва като n -битова кодова дума.
 - Пример:* $k = 2$ бита, $n = 5$ бита
 - Приемника получава кодова дума 00100, която е невалидна, защото не съответства на никоя от дефинираните.
 - Може ли да се направи корекция?

00100 може да бъде 00000

00100 може да бъде 00111

00100 може да бъде 11001

00100 може да бъде 11110

$d(00000, 00100) = 1$

$d(00111, 00100) = 2$

$d(11001, 00100) = 4$

$d(11110, 00100) = 3$

Data Block	Codeword
00	00000
01	00111
10	11001
11	11110

3. Корекция на грешките



Корекция на грешки

- Блоково кодиране за корекция на грешки:
 - Нека предаваните блокове от данни са с дължина k -бита.
 - Вместо да се предава всеки k -битов блок по отделно, той се отбелязва като n -битова кодова дума.
 - Пример:* $k = 2$ бита, $n = 5$ бита
 - Приемника получава кодова дума 00100, която е невалидна, защото не съответства на никоя от дефинираните.
 - Може ли да се направи корекция?

00100 може да бъде 00000

$d(00000, 00100) = 1$

00100 може да бъде 00111

$d(00111, 00100) = 2$

00100 може да бъде 11001

$d(11001, 00100) = 4$

00100 може да бъде 11110

$d(11110, 00100) = 3$

- Ако бъде получена невалидна кодова дума, кода за корекция ще избере една от валидните, която е най-близка до нея според разстоянието на Хеминг.***

Data Block	Codeword
00	00000
01	00111
10	11001
11	11110

3. Корекция на грешките



Корекция на грешки

- Проблем при блоково кодиране за корекция на грешки?
 - Когато невалидната кодова дума е на еднакво Хемингово разстояние с повече от една валидна кодова дума.
 - Код за поправка на грешките не може да определи коя е правилната кодова дума.
 - Грешката е засечена, но не коригирана.
 - При еднобитова грешка, винаги Хеминговото разстояние е 1 от точно една валидна кодова дума, следователно този тип корекция е приложима винаги за еднобитови грешки.
- За примера: $2^5 = 32$ възможни кодови думи, 4 валидни, 28 невалидни

3. Корекция на грешките



Корекция на грешки

- За примера: $2^5 = 32$ възможни кодови думи, 4 валидни, 28 невалидни

Invalid Codeword	Minimum Distance	Valid Codeword	Invalid Codeword	Minimum Distance	Valid Codeword
00001	1	00000	10000	1	00000
00010	1	00000	10001	1	11001
00011	1	00111	10010	2	00000 or 11110
00100	1	00000	10011	2	00111 or 11001
00101	1	00111	10100	2	00000 or 11110
00110	1	00111	10101	2	00111 or 11001
01000	1	00000	10110	1	11110
01001	1	11001	10111	1	00111
01010	2	00000 or 11110	11000	1	11001
01011	2	00111 or 11001	11010	1	11110
01100	2	00000 or 11110	11011	1	11001
01101	2	00111 or 11001	11100	1	11110
01110	1	11110	11101	1	11001
01111	1	00111	11111	1	11110

3. Корекция на грешките



Корекция на грешки

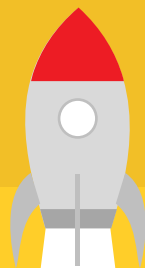
- Примера илюстрира основния принцип на блоковото кодиране за корекция на грешки:
 - Обобщено: (n, k) блоков код трансформира k – битови блокове в n – битови кодови думи. Всяка валидна кодова дума представя k – битов блок от данни с добавени $(n - k)$ *битове за проверка*.
 - При (n, k) блоков код съществуват 2^k валидни кодови думи, от които 2^n са приложими за декодиране на съобщението.
 - Отношението между добавените битове и броя на битовете в блока $(n - k)/k$ се нарича **излишък (redundancy)** на кода;
 - Отношението брой битове в блока към брой битове в кода k/n се нарича **степен на кодиране (code rate)**.

3. Корекция на грешките



Корекция на грешки

- Проектирането на блокови кодове включва следните правила:
 1. За зададени стойност n и k се търси най-голяма стойност на d_{min}
 2. Кодът трябва относително лесно да се кодира и декодира, изисква минимална памет и време за обработка;
 3. Броят на допълнителните битове $(n - k)$ трябва да е малък, за да се намали изискваната честотна лента за предаване;
 4. Броят на допълнителните битове $(n - k)$ трябва да е голям за да се намали честотата на грешките.



**Благодаря за
вниманието!**