计算机网络 第4课　差错控制编码 作业

**班级：** 软工23级普1班 **学号：** 37220232203786 **姓名：** 潘腾凯

# 一、选择题

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 题号 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 选项 | B | B | C | C | D |  |  |  |  |  |
| 题号 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 选项 |  |  |  |  |  |  |  |  |  |  |

# 二、简答题

## 第1题

传输差错来源分类：

干扰（ Interference）：元器件电子辐射；宇宙背景辐射。

失真（ Distortion）：长距离传输会受到干扰。

衰减（ Attenuation）：通过介质的信号会变弱。

差错类型：单个比特、突发差错、模糊差错

## 第2题

首先，查表将字符串 "Hello Network!" 转换为 ASCII 码：

H: 72 e: 101 l: 108 l: 108 o: 111 ‘ ’: 32 N: 78 e: 101 t: 116 w: 119 o: 111 r: 114 k: 107 !: 33

a. 按 8 位分组，计算奇、偶校验码：

将每个字符的 ASCII 码按 8 位分组，然后计算奇、偶校验码。

偶校验编码：原数据+校验位‘1’的个数为偶数。即当原数据 '1' 的个数为偶数，偶校验位为 0，反之为1。

奇校验编码：原数据+校验位‘1’的个数为奇数。即当原数据 '1' 的个数为奇数，偶校验位为 0，反之为1

用末尾再加一位作为校验位，计算过程举例：

如H的ASCII码为01001000，则偶校验码为010010000，奇校验码为010010001

按如上算法可得该数据的奇校验码为010010001 011001011 011011000 011011000 011011110 001000000 010011101 011001011 011101000 011101110 011011110 011100101 011011110 001000011

偶校验码为01001000 01100101 01101100 01101100 01101111 00100000 01001110 01100101 01110100 01110111 01101111 01110010 01101111 00100001

这样的编码能检测 1 位错误，因为如果有 1 位错误，奇偶性就会改变。但不能自动纠正错误，因为无法确定是哪一位发生了错误。

b.计算 Internet 校验和。

数据对应的二进制表示为（8位一组）：

01001000 01100101 01101100 01101100 01101111 00100000 01001110 01100101 01110100 01110111 01101111 01110010 01101111 00100001

先将数据按字节大端序分成 16位一组。如果最后一组不足16位，则用 0 填充。

分组如下：

01001000 01100101 → 0x4865

01101100 01101100 → 0x6C6C

01101111 00100000 → 0x6F20

01001110 01100101 → 0x4E65

01110100 01110111 → 0x7477

01101111 01110010 → 0x6F72

01101111 00100001 → 0x6F21

计算步骤

1. 按 16 位分组相加

将每组 16 位数据加到 sum 中。如果 sum 超过 16 位，则保留低 16 位，并将高 16 位加回到低 16 位中（即“折叠”）。

分组相加：

0x4865 + 0x6C6C = 0xB4D1

0xB4D1 + 0x6F20 = 0x123F1 → 折叠为 0x23F1 (保留低 16 位)

0x23F1 + 0x4E65 = 0x7256

0x7256 + 0x7477 = 0xE6CB

0xE6CB + 0x6F72 = 0x1563D → 折叠为 0x563D

0x563D + 0x6F21 = 0xC55E

最终 sum = 0xC55E。

2.将 sum 按位取反，得到校验和。

0xC55E 的二进制表示为：1100 0101 0101 1110

取反后为：0011 1010 1010 0001

即最终结果Internet Checksum 为 ​0x3AA1。

## 第3题

已知发送字节 “.”，对应二进制数据是 00101110，采用 CRC 的产生多项式是 P(x)=x^3+x+1，其对应的二进制序列为 1011。

a. 用算数除法演算添加在数据后的余数：

将原始数据 00101110 后面添加 3 个 0（因为产生多项式的最高次幂为 3），得到 00101110000。然后用 1011 去除 00101110000：

00010011

-------------

1011 | 00101110000

- 00000000

-------------

001011

- 000000

-------------

001011

- 001011

-------------

0000

所以余数为 000。

b. 最终发送的数据

将余数 000 添加到原始数据 00101110 后面，最终发送的数据是 00101110000。

c. 整个过程传输中，最后一位发生错误，接收端能否发现

假设最后一位由 0 变为 1，接收数据变为 00101110001。用 1011 去除 00101110001：

00010011

-------------

1011 | 00101110001

- 00000000

-------------

001011

-000000

-------------

001011

-001011

-------------

0001

余数不为 0，接收端可以发现错误。

d. 举例说明发生何种错误，接收端无法发现？错误的位应尽量少。

如果错误发生在能被生成多项式整除的位置上，接收端就无法发现错误。

对于生成多项式 P(x)=x^3+x+1，如果错误发生在使得错误数据能被 1011 整除的位置上，接收端就无法发现。

例如，假设错误发生在第 4 位和第 7 位（从左到右，从 0 开始计数），原始数据 00101110 变为 00111100，添加余数 000 后发送数据为 00111100000。

用 1011 去除 00111100000：

00011001

-------------

1011 | 00111100000

- 00000000

-------------

001111

- 001011

-------------

00100

- 00000

-------------

001000

- 001011

-------------

1100

- 1011

-------------

0110

- 0000

-------------

01100

- 01011

-------------

0010

- 0000

-------------

00100

- 001011

-------------

0000

余数为 0，接收端无法发现错误。

# 三、编程题

代码上传于：。