Reliability of Data Transmission

The Method of Hamming Code to Detect and Correct Error

Wang Zhongsheng
Xi'an Technological University

201 8 .9

1. Introduction

In the process of data transmission, how to ensure the consistent and correct is very important. This lesson introduces a method to detect and correct an error code in the data transmission process.

Hamming Code is an Error correction Code that can Check multiple bits, and can correct one bit Error Code, so it is only used in a good channel characteristics, such as Ethernet (Internet and Intranet), if the channel environment are bad, the Error is usually not one bits. This method is invalid.

2．Method

The method is divided effective information into several groups according to certain rule, each group to arrange a Check bit and generates a number of testing information, then tested (Odd or even test), and find the exact location of the error bit, finally invert the wrong code (also convert 1 to 0 or 0 to 1) to correct it.

3. Sequence

There are following steps to use Hamming Code, (1) Calculate the number of the check bits (how many?), (2) Determine the position of the check bits(where?), (3) Determine the value of check code(what?), (4) checking and correct error.

The first step is to calculate the number of the check bits, and is also called Hamming distance or the Check code length.

3.1 Method

Rule is as following: suppose N is used to represent the binary numbers of the whole information after adding the check code bit, used K stand for the information code number, r stand for the added check code number, the relationship between them should meet:

$$N=K+r\leq 2^r-1$$

or $K+1\leq 2^r-r$ **.**

For example, If K=5, it is required that 2^r-r is greater than or equal to 5+1=6. According to the calculation, it can be known that the minimum number of r is 4, that is, to verify 5 bits information code, the verification code should be 4bit.

If the information code is 8 bits (K=8), it is required that 2^r-r is greater than or equal to 8+1=9. According to the calculation, the minimum value of r is also 4 .

According to the experience, the relationship between the information code number and the check code number is shown in table 1.

Inf. Code Number	1	2~4	5~11	12~26	27~57	58~120	121~247
Check Code Number	2	3	4	5	6	7	8

3.2 Determine the position of the check code

Although we have determined the number of check code, we must know the position to insert into the code line, as they are not directly attached to the front, back or middle, but are inserted separately into different positions.

The position of the check code must be at position 2^n (Two to the n) , that is in No. 1, 2, 4, 8, 16,... (corresponding to 2^0, 2^1, 2^2, 2^3, 2^4..., starting from the leftmost), then the position of the information code can be determined, which is the not 2^n (Two to the n) position, such as No.3, 5, 6, 7, 9, 10, 11, 12, 13... (from the leftmost digit).

For example, suppose there is an 8 bits information code, namely b1, b2, b3, b4, b5, b6, b7 and b8. It is known from table 1 that it needs to insert 4 bits check code, namely p1, p2, p3, p4, that is, the whole coded is total to 12 bits. According to the above mentioned location distribution rules of the check code, the data after recoding is p1, p2, b1, p3, b2, b3, b4, p4, b5, b6, b7 , b8.

Suppose the original 8 bits code is 10011101, because the value of each check code has not been calculated yet. The final code sequence is: ??1?001?1101.

3.3 Determine the Value of the check code

After the two steps, we have determined the numbers and the position of the check code, but we also do not know the value. The values of these check codes are not random, and the values of each check code represent the parity of some data bits in the code (ultimately determined to use odd or even check).

The general rule is:

the No. i check code starts from the current position, each time continuous check i (here is the number i, not the bit i, the same below) bits, then skip bit i, continuous check i bits, then skip bit i, and so on.

Finally, the value of check i can be obtained according to the odd check or even check.

1） Calculate the value

To calculate the value of the check code is as follows：

The check rule of p1 (the first check code, also the first bit of the whole code) is:

from the current number, check 1 bit, then skip 1, then go on check 1 bit, then skip 1. Thus, it can be concluded that the code bits that can be verified by the p1 check code include: 1 (that is, p1 itself, the interval is 1), 3, 5, 7, 9, 11, 13, 15...., Then, the value of the check code can be determined according to the odd or even checking.

The check rule of p2 (the second check code, also the second bit of the whole code word, the interval is 2) is: from the current number, check 2 bit, then skip 2, then go on check 2 bit, then skip 2. Thus, it can be concluded that the code bits that can be verified by the p2 check code include: 2 (that is, p2 itself), 3, 6, 7, 10, 11, 14, 15...., Then, the value of the check code can be determined according to the odd or even checking.

As the same above, p3 (the third check code, also the forth bit of the whole code word, the interval is 4) check code include: 4, (that is, p4 itself), 5 ,6, 7 , 12 ,13, 14 ,15, 20, 21 ,22, 23 ,.....

As the same above, p4 (the forth check code, also the 8th bit of the whole code word, the interval is 8) check code include: 8, (that is, p4 itself), 9 ,10, 11 , 12 ,13, 14 ,15, 24, 25 ,26, 27,28 ,29, 30 ,31, ...

We divide the bits verified by the above check codes into corresponding groups, and on the receiving end their detect results are expressed as G1, G2, G3, G4...,

The check codes perform an exclusive OR with the information code, under normal circumstance; the correct answer would be zeroes.

EOR: $0 \oplus 0=0$, $1 \oplus 0=1$, $0 \oplus 1=1$, $1 \oplus 1=0$.

2）Go on the above example

Calculate the example, the code is: ？？1？001？1101.

Let's find the first "? ".Since the length of the whole code number is 12 (including 8 information code and 4 check code), it can be concluded the p1 in this example is 1, 3, 5, 7, 9, 11. The values of the other five's are known except that the first (i.e. the p1). They are: 1, 0, 1, 1 and 0. Now, we adapt even check (that is, the number of "1" in the whole tested bit is required to be even). From the known 5 code value, it can be known that there are 3 "1", so the value of the p1 must be "1", and it is concluded that p1=1.

And then the second "?"(that is, p2). According to the above rules, it can be quickly concluded that the number of p2 check code is 2, 3, 6, 7, 10, 11, and a total of 6. The values of the other 5 are known except that the second (i.e. the p2). They are: 1, 0, 1, 1 and 0. It is assumed the even check. It can be seen the 5 code value there are also 3 "1". Therefore, the value of p2 must be "1", and p2=1 is obtained.

And in the same way, we figured out that P3= 0, and p4=1.

At last, it can be concluded the whole code word is: 11100011101 (the four bits with black are check code)

3.4. Detect and Correct error

Hamming code is a multiple check code, that is, the information code is checked by multiple check codes at the same time, the check codes perform an exclusive OR with the information code, and finally found out the wrong bit.

1）Detect error

When the total code word is 18 bits, which can be obtained quickly according to table 1, among which 5 bits are the check code. According to the check code rule introduced earlier in this section, the code position can be obtained quickly, as shown in table 2.

Whole code position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
check code/Inf. code	P1	P2	b1	P3	b2	b3	b4	P4	b5	b6	b7	b8	b9	b10	b11	P5	b12	b13
G1 Associated P1	√		√		√		√		√		√		√		√		√	
G2 Associated P2		√	√			√	√			√	√			√	√			√
G3 Associated P3				√	√	√	√					√	√	√	√			
G4 Associated P4								√	√	√	√	√	√	√	√			
G5 Associated P5																√	√	√

It can be found from table 2 that all information code are checked by at least two check codes, that is, at least twice. Check which two groups of verification results are inconsistent, and then, according to table 2, you can quickly determine which bit of information code is wrong during transmission.

The check formula as follows：

G1=p1⊕b1⊕b2⊕b4⊕b5⊕b7⊕……

G2=p2⊕b1⊕b3⊕b4⊕b6⊕b7⊕b10⊕b11⊕……

G3= p3⊕b2⊕b3⊕b4⊕b8⊕b9⊕b10⊕b11⊕……

G4= p4⊕b5⊕b6⊕b7⊕b8⊕b9⊕b10⊕b11⊕……

Under normal circumstances (that is, the whole code word does not have error), when using even check, the check result after each check group should be 0, that is, the G1, G2, G3, G4..., All of them are zero, because at this point, one is an even number, and when you do the XOR, you get zero; In the case of odd check, the result of each group should be 1.

Now let's go on the example, the whole code is:

11100011101.

We can know that there are four test set: G1, G2 and G3, G4, arrived at the receiving end, however, found that only after checking G4 = 1 (that is, only this set of checking result is not zero), according the above rules can be found quickly is G4 group p4 in No.8 position) in which is the code word wrong, because only a set of calibration results appear error, corresponding check digit is sure just goes wrong), also is the final code word becomes: 11100001101.

We also assume that neither G3 nor G4 is 0, which is equal to 1. Compare the two check groups of G3 and G4 in table 2 (note that the length of code word in this example is only 12 bits, only the first 12 bits need to be compared), but the code bit of the common check is found to be b8, that is, the 12th bit has an error, that is, the final code word becomes 1110000111100.

Prof. Wang Zhongsheng

wzhsh1681@63.com

Contact :

School of Computer Science and Engineering,
Xi'an Technological University,
Xi'an, 710032, China
VK, id506332858
Wechat: diamond-wzs
QQ-mail: 59483672@qq.com