P1 CSC0056 Data Communication Homework 2 2020 version

Answer

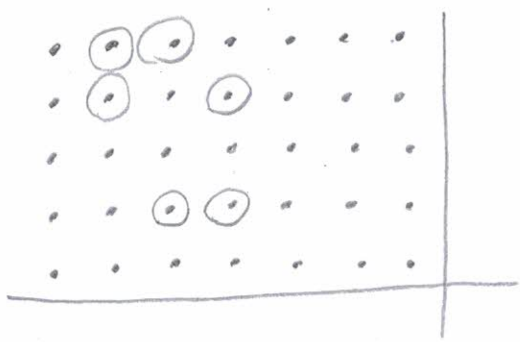
P1

(1) You may give your answers based on the content in Section 1.3 in the textbook. Basically, we'd like to see your reasoning here (e.g., not just say "it offers better understandability; explain why it offers better understandability.)

(2a) Refer to Section 2.3.1. Basically, using the single parity check, the resulting "data bits + parity check bit" must have even number of 1's. Now, in presence of odd number of errors (be it from 0 to 1 or from 1 to 0) we will have odd number of 1's in "data bits + parity check bit."

(2b) Following the answer of 2a, a random bit string has 50% of chance to have even number of 1's in total. Alternatively, you may use the metric in Section 2.3.3 and argue that $L=1$ in the single parity check and thus $2^{-1} = 50\%$.

(2c)



(2d) Refer to Section 2.3.4. If $g(D)$ is of degree 4, then the remainder polynomial is of degree 3 at most, which means we will use 4 bits to hold the coefficients of the remainder polynomial. Thus the number of parity checks in this case is 4.

Student ID: ____________ Name: ____________

P2

2e

```
      000
101 ) 00000
      000
       000
       000
        000
        000
         00
```

⇒ code word 00000

```
      001
101 ) 00100
      000
       010
       000
        100
        101
         01
```

⇒ code word 00101

```
      010
101 ) 01000
      000
       100
       101
        010
        000
         10
```

⇒ code word 01010

```
      011
101 ) 01100
      000
       110
       101
        110
        101
         11
```

⇒ code word 01111

```
      101
101 ) 10000
      101
       010
       000
        100
        101
         01
```

⇒ code word 10001

```
      100
101 ) 10100
      101
       000
       000
        000
        000
          0
```

⇒ code word 10100

```
      111
101 ) 11000
      101
       110
       101
        110
        101
         11
```

⇒ code word 11011

```
      110
101 ) 11100
      101
       100
       101
        010
        000
         10
```

⇒ code word 11110

The minimum distance of

{ 00000
00101
01010
01111
10001
10100
11011
11110

is 2

#

Name: ______ Student ID: ______

(2f)

```
        11011
101 ) 1110100
      101
      100
      101
       011
       000
        110
        101
         110
         101
          11
```

$\Rightarrow$ the code word is <u>1110111</u> \#

(3a) Because otherwise the Hamming weight of a linear code (which we stated as Theorem 1 in clas would be zero, which in turn would make Theorem 3 (in the useless. On the other hand, Theorem 3 still applies to a linear code within which a code word may be all-zero, even if we do not consider that code word when determinin the Hamming weight of the linear code: the definition of the Hamming weight implicitly considers the Hamming distance between ANY non-zero code word and the all-zero code word, since $d(u,v) = wt(u-v)$.

(3b1)

| data bits | | code word |
|---|---|---|
| 000 | $\rightarrow$ | 000000 |
| 001 | $\rightarrow$ | 001101 |
| 010 | $\rightarrow$ | 010001 |
| 011 | $\rightarrow$ | 011100 |
| 100 | $\rightarrow$ | 100110 |
| 101 | $\rightarrow$ | 101011 |
| 110 | $\rightarrow$ | 110111 |
| 111 | $\rightarrow$ | 111010 |

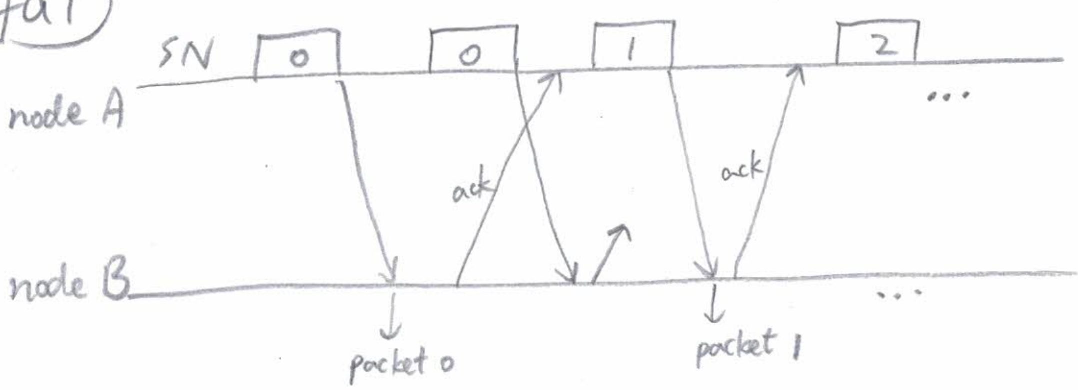
$\Rightarrow$ the Hamming weight of this linear code is 2 \#

(3b2) Using the nearest-neighbor rule, we will say that the original data is $(100110)_2$, which differs by one bit wit respect to $(101110)_2$.

Interestingly, suppose the received bit vector is $(000001)_2$ instead. The nearest-neighbor rule would give two plausible answer : $(000000)_2$ and $(010001)_2$. Because the Hamming weight of this code is 2, our Theorem does not give us any guarantee (since $t=0$ in this case).

(3b3) Suppose the original data is $(000)_2$ and the code word is thus $(000000)_2$.
If the two-bit error leds to $(010001)_2$ received by a receiver, then we cannot correct the error using the nearest-neighbor rule because $(010001)_2$ itself is a code word.

(4a1)




Name: Student ID:

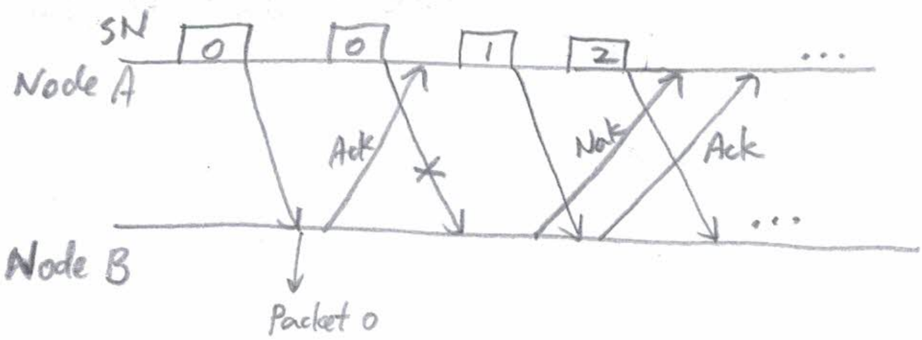
4a2 It cannot. Although the one who has a global view (such as we) can tell for sure, from the viewpoint of Node A it is indistinguishable that which of the following case happened:

case ①: Figure 2.19

case ②: the Nak of packet 1 was lost, and packet 2 was corrupted, and then the Nak for packet 2 was delivered successfully.

Although the Figure illustrated case ①, Node A cannot make sure that was really the case.

4a3 ① Yes, and this is possible if Node A does not need to wait for an Ack before it sends a packet with a new SN or that packets may arrive out-of-order:



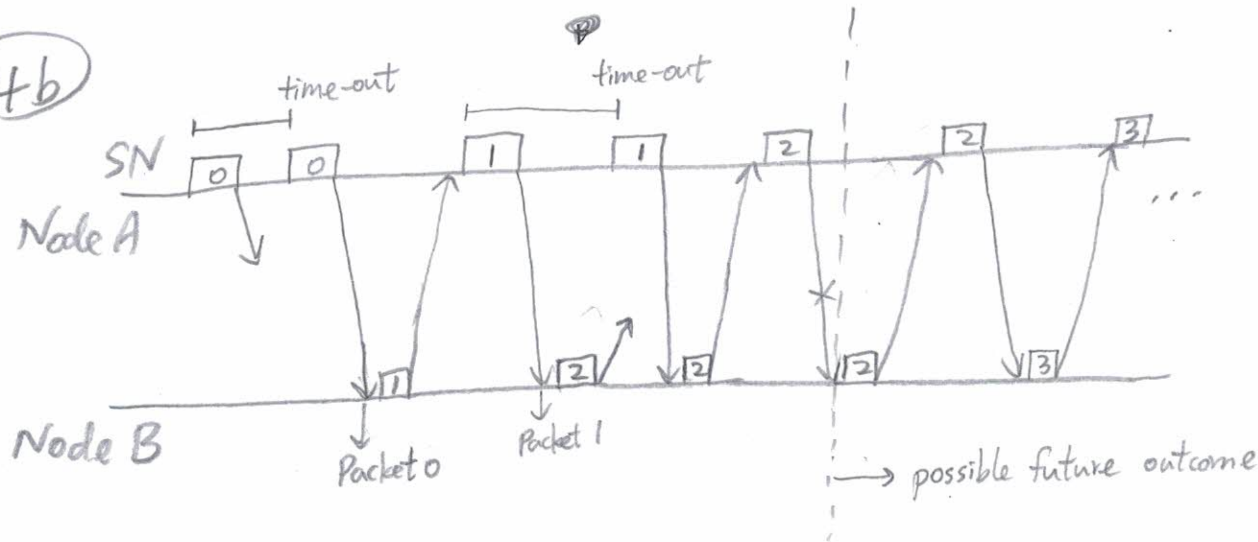

Give either ① or ② is sufficient to receive full score

or

② No, if we suppose Node A must receive an Ack before it can send a packet with a new SN, and that packets arrive in order.

P6

4b




4c

$$T_2 = T_1 + T_{to}(q^{-1}-1) \le 1.1\,T_1$$

$$\Rightarrow T_1 + T_o\left(\frac{1}{0.8}-1\right) \le 1.1\,T_1$$

$$\Rightarrow T_o \le \frac{0.1}{0.25}\,T_1 = \underline{0.4\,T_1}$$

#

Student ID: Name: