Solutions to exercise 2 in EITF25 Internet - Techniques and Applications - 2013

November 25, 2014

1

If a transmission is subject to period of severe interference, the bits during that time might have been corrupted. For example: a 1 Mbps transmission is subject to a period of 1 milliseconds of interference. A 1 Mbps transmission produces bits at a rate of 10^6 bit per second or $\frac{10^6}{10^3}$ bits per millisecond. As such, during a period of 1 milliseconds 1000 bits might have been corrupted.

a

Following the reasoning in the example above: $10 \cdot \frac{10^3}{10^3} = 10$ bits per millisecond, during 2 milliseconds $10 \cdot 2 = 20$ bits.

Answer: 20 bits

b

Following the reasoning in the example above: $100 \cdot \frac{10^3}{10^3} \cdot 2 = 200$ bits.

Answer: 200 bits

c

Following the reasoning in the example above: $\frac{10^6}{10^3} \cdot 2 = 2000$ bits.

Answer: 2000 bits

2

Adding a parity bit to a sequence is the simplest for of error detection. You compute the even parity bit by doing a bit-wise sum of the sequence. Which will give you a one if there is an odd number of ones in a sequence or zero if there is a even number of ones in the sequence.

a

Following the reasoning in the explanation above.

Answer: 1

b

Following the reasoning in the explanation above.

Answer: 1

c

Following the reasoning in the explanation above.

Answer: 0

d

Following the reasoning in the explanation above.

Answer: 0

3

In this set of problems we are asked to compute the CRC bit sequence $R(x)$ for the specified outbound bit sequences $M(x)$ to produce the $F(x)$ sent sequence, on the sender side. See Figure 1.

$$F(x) = \underbrace{M(x) \cdot x^k}_{B(x)} + R(x)$$

Figure 1: CRC

Moreover, k is the degree of the CRC generator polynomial. You may have noticed that the bit sequences are represented as a function of x. You can express a bit sequence $M(x)$ as a polynomial by, from right to left, multiplying the n^{th} bit with the n^{th} polynomial term. See Figure 2 for example.

$$011001 \rightarrow 0 \cdot x^5 + 1 \cdot x^4 + 1 \cdot x^3 + 0 \cdot x^2 + 0 \cdot x^1 + 1 \cdot x^0 \rightarrow x^4 + x^3 + 1$$

Figure 2: Polynomial representation of a bit sequence.

Multiplying the resulting polynomial $M(x)$ with x^k yields the term $B(x)$. This done to shift the sequence enough to give space to the k CRC bits. Dividing the polynomials $\frac{B(x)}{C(x)}$ in modulo-2 yields a reminder that when appended to the target bit sequence polynomial $B(x)$ yields the polynomial $P(x)$ that is evenly divisible with the generator polynomial $C(x)$. The final polynomial $P(x)$ is then expressed as a sequence of bits and transmitted.

a

$C(x) = x^3 + x^2 + 1 \rightarrow k = 3$
$M(x) = x^5 + x^4 + x^3 + x$
$B(x) = x^8 + x^7 + x^6 + x^4$
$R(x) = x$

Answer: 010

b

$C(x) = x^3 + x^2 + 1 \rightarrow k = 3$
$M(x) = x^9 + x^7 + x^4 + x^3 + x^2 + x$
$B(x) = x^{12} + x^{10} + x^7 + x^6 + x^5 + x^4$
$R(x) = x^2 + 1$

Answer: 101

c

$C(x) = x^3 + x^2 + 1 \rightarrow k = 3$
$M(x) = x^8 + x^7 + x^6 + x^2 + x + 1$
$B(x) = x^{11} + x^{10} + x^9 + x^5 + x^4 + x^3$
$R(x) = x^2 + x + 1$

Answer: 111

d

$C(x) = x^3 + x^2 + 1 \rightarrow k = 3$
$M(x) = x^9 + x^8 + x^5 + x^4 + x + 1$
$B(x) = x^{12} + x^{11} + x^8 + x^7 + x^4 + x^3$
$R(x) = x^2 + 1$

Answer: 101

4

On the receiver side, if no errors have been injected into the inbound bit sequence $P(x)$, it should be evenly (modulo-2) divisible with the CRC generator polynomial $C(x)$, i.e. no reminder, $E(x) = 0$. If no remainder is produced, removing the appended CRC bit sequence of length k from inbound sequence yields the original sequence.

a

$C(x) = x^4 + x^3 + 1 \rightarrow k = 4$
$P(x) = x^7 + x^6 + x^4 + x^2 + x + 1$
$E(x) = x^2 + x$

Answer: NOK

b

$C(x) = x^4 + x^3 + 1 \rightarrow k = 4$
$P(x) = x^{10} + x^8 + x^6 + x^5 + x^3 + x^2 + 1$
$E(x) = x^3 + x^2 + 1$

Answer: NOK

c

$C(x) = x^4 + x^3 + 1 \rightarrow k = 4$
$P(x) = x^{10} + x^6 + x^5 + x^4 + x^2 + x + 1$
$E(x) = 0$

Answer: OK

5

The checksum in produced by performing a bitwise addition of the two halves of the sequence from right to left, and where the remainder is carried to the next digit. The inverse of the resulting bit sequence is then appended to the original sequence and transmitted. In other words, an n-bit checksum is the inverse of the bit-wise sum of size n data bit-sequences.

a

Following the reasoning in the explanation above.

Answer: 11011000

b

Following the reasoning in the explanation above.

Answer: 10010011

c

Following the reasoning in the explanation above.

Answer: 00101011

6

On the receiver side, the inbound sequence is bit-wise summed as sections of size n. The inverted result is compared to the checksum sequence, if they match the sequence is accepted, if not errors have most likely been introduced during transmission.

a

Sum: 1101000∅

Answer: NOK

b

Sum: 00010101

Answer: OK

c

Sum: 01010111

Answer: OK

7

The avoid wrongful detection of the HDLC (High-Level Data Link Control) frame delimiter, represented by the bit-sequence: 01111110. Therefore, on the transmitter side, zeros have been added after each consecutive five ones. As such, the bit-stuffed bits carry no information that is relevant to the sender/receiver applications and are thus removed on the receiver end.

$$000111110111110111001 0110 \rightarrow 00011111\emptyset11111\emptyset1110010110$$

HDLC deploys NRZ, and uses the frame delimiter to synchronize the signal.

a

$$\ldots 0101011111\not{0}10111011111\not{0}0 \ldots$$

When bit stuffed zeros are removed:

Answer: ...0101011111101110111110 ...

b

$$\ldots 0101\underline{01111110}10111011111\not{0} \ldots$$

Note that this sequence contains an actual frame delimiter, underlined. When bit stuffed zeros are removed:

Answer: ...0101011111101010111011111 ...

8

The frame delimiter in HDLC (High-Level Data Link Control) is 01111110. To avoid any wrongful detection on the receiver side, any occurrence of a sequence of ones of length six or greater is stuffed with a zero after each consecutive five ones. For example, at the transmitter side:

$$0001111111001111110010110 \rightarrow 000111110110011111011100 10110$$

As seen in the previous question 7, the bit-stuffed zeros are removed on the receiver side.

a

$$000111111011111001111 1001 \rightarrow 00011111010111110001111 10001$$

Which yields:

Answer: 0001111101011111000111110001

b

$$000111111111111111111111111111110011111001 \rightarrow$$
$$000111110111110111110111110111110111100111110001$$

Which yields:

> **Answer**: 000111110111110111110111110111110111100111110001

9

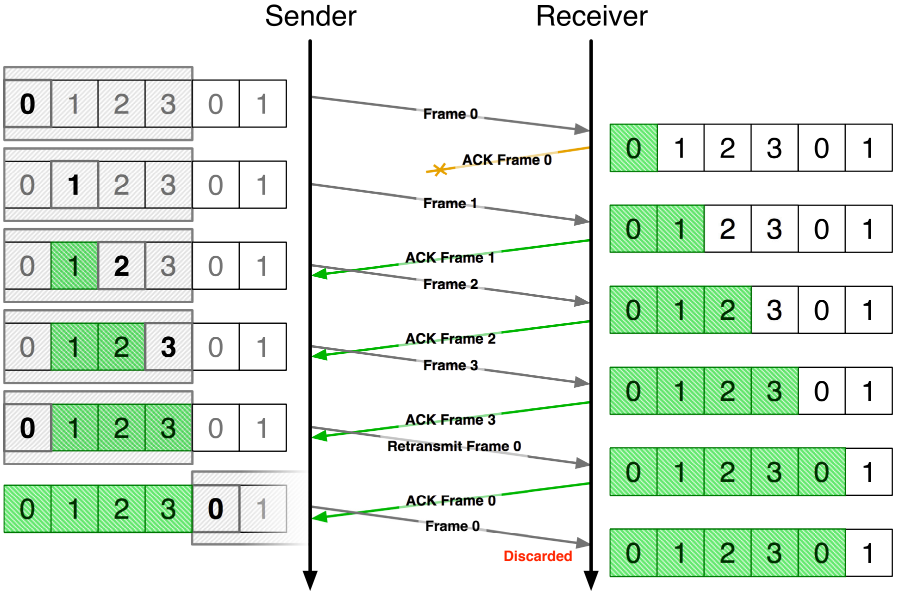
In Go-Back-N-ARQ a windows size of 15 needs to be represented with at least $\lceil \log_2(15) \rceil = 4$ bits.

> **Answer**: 4 bits

10

In a communication system, the sequence number is represented by 2 (N=2) bits and this thus able to hold $2^2 = 4$ frames. If the sliding-window size also was four, and the sender has transmitted all four frames in the sequence but an acknowledgement for frame 0 has not been received. Nevertheless, all frames were received on the reviver side. If the sender was now to retransmit the first frame, the 0^{th} frame, then the receiver would treat that as the first frame in the subsequent sequence. The sender would then treat the resulting acknowledgment as as if though it was intended for the first frame from the first sequence. The sender would proceed with sending the first frame from the second sequence, which will be discarded by the receiver as there it thinks it has already received that frame. On the other hand, if the sliding windows was instead of size 3 (N-1) then the receiver would be bounded to the first sequence.

Answer:



11

In Selective repeat-ARQ the windows size can at the most be half of the largest possible sequence number. It is stated that the sequence number is represented by 7 bits, thus a total of $2^7 = \frac{128}{2} = 64$ frames can be monitored.

Answer: 64 frames

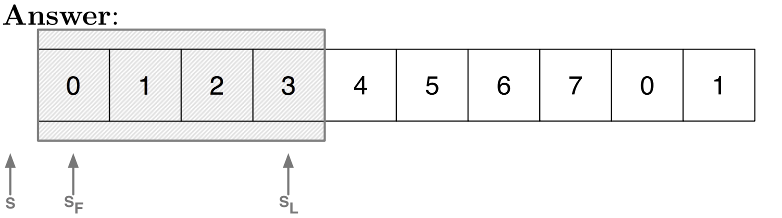
12

Since the sequence number is represented with 3 bits, it can represent 8 frames. The S marker marks the last transmitted frame, S_F, where the window starts i.e. one frame after the last consecutively acknowledge frame. Moreover, thus S_L marks the end of the window.

a

Before anything has been sent, the windows will envelope the first 4 frames.

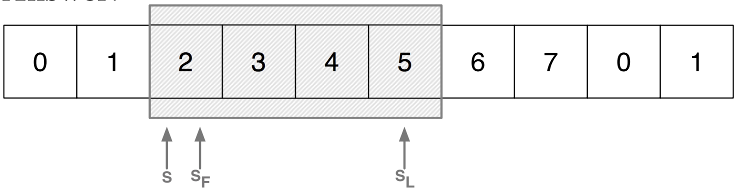
Answer:



b

ACKs have been received for frames 0 and 1, hence the S_F marker will be at frame 2, the earliest unacknowledged frame.

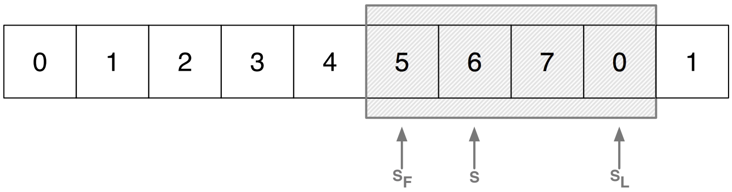
Answer:



c

The P marker till point to packet 6, as it is the highest sent frame. Nevertheless, only frame 4 has been acknowledged, as such S_F will not de mark the beginning of the open window at frame 5.

Answer:



13

a

The speed of light in vacuum is $c = 299792458$ m/s. Sending a packet in one direction thus takes $\frac{4000000\text{m}}{c} = 0.01334$ seconds $= 13.34$ milliseconds. Assuming that the process time is 0 seconds, the total time will henceforth be packet round trip of $2 \cdot 13.34 = 26.68$ milliseconds.

Answer: 26.68 milliseconds

b

As one byte contains 8 bits, 1000 bytes equates to 8000 bits. Transmission time is thus: $\frac{8 \cdot 10^3 \text{bits}}{100 \cdot 10^6 \text{bps}} = 0.08$ milliseconds $= 80$ microseconds.

Note that bytes are mainly used when referring to data storage as one byte is historically the size of an ASCII symbol, and thus the smallest addressable space in a computers memory.

Answer: 0.08 milliseconds

c

Out of the total transmission (0.08 ms) and propagation (26.68 ms) time the sender is only occupied while transmitting. Hence, for the event the that sender sends multiple consecutive frames it is vacant $\frac{26.69}{26.68+0.08} = 99,7\%$ of the time.

Answer: 99.7 %