Computer Networks
Problem Set 3

Instructor: Shashi Prabh

3.1 The following character encoding is used in a data link protocol:
A: 01000111 B: 11100011 FLAG: 01111110 ESC: 11100000
Show the bit sequence transmitted (in binary) for the four-character frame A B ESC FLAG when each of the following framing methods is used:

a. Byte count.

b. Flag bytes with byte stuffing.

c. Starting and ending flag bytes with bit stuffing.

3.2 The following data fragment occurs in the middle of a data stream for which the byte-stuffing algorithm described in the text is used: A B ESC C ESC FLAG FLAG D.
What is the output after stuffing?

3.3 What is the maximum overhead in byte-stuffing algorithm?

3.4 You receive the following data fragment: A ESC FLAG A B A FLAG FLAG C B ESC FLAG ESC ESC ESC FLAG FLAG. You know that the protocol uses byte stuffing. Show the contents of each frame after destuffing.

3.5 You receive the following data fragment: 0110 0111 1100 1110 0111 1101. You know that the protocol uses bit stuffing. Show the data after destuffing.

3.6 One of your classmates, Scrooge, has pointed out that it is wasteful to end each frame with a flag byte and then begin the next one with a second flag byte. One flag byte could do the job as well, and a byte saved is a byte earned. Do you agree?

3.7 A bit string, 0111101111101111110, needs to be transmitted at the data link layer. What is the string actually transmitted after bit stuffing?

3.8 An upper-layer packet is split into 10 frames, each of which has an 80% chance of arriving undamaged. If no error control is done by the data link protocol, how many times must the message be sent on average to get the entire thing through?

3.9 Can you think of any circumstances under which an open-loop protocol (e.g., a Hamming code) might be preferable to the feedback-type protocols discussed throughout this chapter?

3.10 To provide more reliability than a single parity bit can give, an error-detecting coding scheme uses one parity bit for checking all the odd-numbered bits and a second parity bit for all the even-numbered bits. What is the Hamming distance of this code?

3.11 Sixteen-bit messages are transmitted using a Hamming code. How many check bits are needed to ensure that the receiver can detect and correct single-bit errors? Show the bit pattern

transmitted for the message 1101001100110101. Assume that even parity is used in the Hamming code.

3.12

a. An 8-bit byte with binary value 10101111 is to be encoded using an even-parity Hamming code. What is the binary value after encoding?

b. Assume a 12-bit Hamming codeword consisting of 8-bit data 110x0101 and 4 check bits Y010. What are X and Y if data is encoded using even parity?

3.13 A 12-bit Hamming code whose hexadecimal value is 0xE4F arrives at a receiver. What was the original value in hexadecimal? Assume that not more than 1 bit is in error.

3.14 One way of detecting errors is to transmit data as a block of n rows of k bits per row and add parity bits to each row and each column. The bit in the lower-right corner is a parity bit that checks its row and its column. Will this scheme detect all single errors? Double errors? Triple errors? Show that this scheme cannot detect some four-bit errors.

3.15 A block of bits with n rows and k columns uses horizontal and vertical parity bits for error detection. Suppose that exactly 4 bits are inverted due to transmission errors. Derive an expression for the probability that the error will be undetected.

3.16 Using the convolutional coder given below, what is the output sequence when the input sequence is 10101010 (left to right) and the internal state is initially all zero?

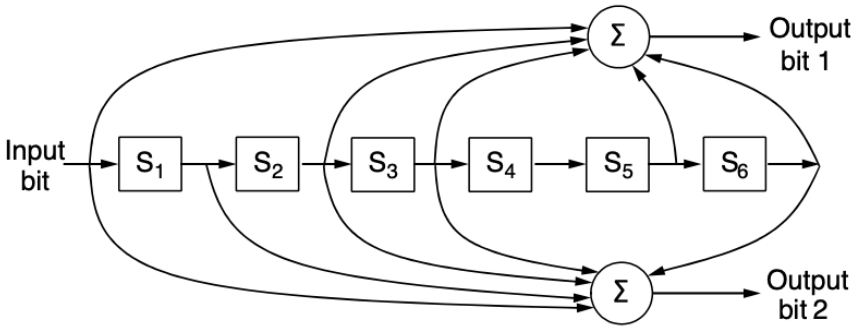

Figure 3-7. The NASA binary convolutional code used in 802.11.

Figure 1: Exercise 3.16

3.17 A bit stream 10011101 is transmitted using the standard CRC method described in the text. The generator polynomial is x^3+1. Show the actual bit string transmitted. Suppose that the third bit from the left is inverted during transmission. Show that this error is detected at the receiver's end. Give an example of bit errors in the bit string transmitted that will not be detected by the receiver.

3.18 A 1024-bit message is sent that contains 992 data bits and 32 CRC bits. CRC is computed using the IEEE 802 standardized, 32-degree CRC polynomial. For each of the following, explain whether the errors during message transmission will be detected by the receiver:

a. There was a single-bit error.

b. There were two isolated bit errors

c. There were 18 isolated bit errors

d. There were 47 isolated bit errors

e. There was a 24-bit long burst error

f. There was a 35-bit long burst error

3.19 In the discussion of ARQ protocol in Section 3.3.3, a scenario was outlined that resulted in the receiver accepting two copies of the same frame due to a loss of acknowledgement frame. Is it possible that a receiver may accept multiple copies of the same frame when none of the frames (message or acknowledgement) are lost?

3.20

a. A channel has a bit rate of 4 kbps and a propagation delay of 20 msec. For what range of frame sizes does stop-and-wait give an efficiency of at least 50%?

b. What is the throughput of a system that uses stop-and-wait ARQ for transmitting 1000 Byte frames on a 40 Kbps link? The receiver sends 100 Byte ACKs on an 8 Kbps link. The one-way propagation delay is 50 ms.

3.21 A 3000-km-long T1 trunk is used to transmit 64-byte frames using protocol 5. If the propagation speed is 6 μsec/km, how many bits should the sequence numbers be?

3.22 Imagine a sliding window protocol using so many bits for sequence numbers that wraparound never occurs. What relations must hold among the four window edges and the window size, which is constant and the same for both the sender and the receiver?

3.23 The distance from earth to a distant planet is approximately 9×10^{10} m. What is the channel utilization if a stop-and-wait protocol is used for frame transmission on a 64 Mbps point-to-point link? Assume that the frame size is 32 KB and the speed of light is 3×10^8 m/s.

3.24 In the previous problem, suppose a sliding window protocol is used instead. For what send window size will the link utilization be 100%? You may ignore the protocol processing times at the sender and the receiver.

3.25 In protocol 6, $MAX\_SEQ = 2^n - 1$. While this condition is obviously desirable to make efficient use of header bits, we have not demonstrated that it is essential. Does the protocol work correctly for $MAX\_SEQ = 4$, for example?

3.26 Consider a protocol that uses piggybacking, a sending window size of 4, and 400-bit frames. This protocol is used to transfer data over a 200 kbps channel with a 4 msec one-way propagation delay. Unfortunately, the receiver has no data to send back. It needs to send its acknowledgements in separate frames. What is the maximum amount of time the receiver can wait before sending, such that the bandwidth efficiency does not drop below 50%?

3.27 Compute the fraction of the bandwidth that is wasted on overhead (headers and retransmissions) for protocol 6 on a heavily loaded 50-kbps satellite channel with data frames consisting of 40 header and 3960 data bits. Assume that the signal propagation time from the earth to the satellite is 270 msec. ACK frames never occur. NAK frames are 40 bits. The error rate for data frames is 1%, and the error rate for NAK frames is negligible. The sequence numbers are 8 bits.

3.28 A 100-km-long cable runs at the T1 data rate. The propagation speed in the cable is 2/3 the speed of light in vacuum. How many bits fit in the cable?

3.29 Give at least one reason why PPP uses byte stuffing instead of bit stuffing to prevent accidental flag bytes within the payload from causing confusion.

3.30 A 100-byte IP packet is transmitted over a local loop using ADSL protocol stack. How many ATM cells will be transmitted? Briefly describe their contents.

3.31 The goal of this lab exercise is to implement an error-detection mechanism using the standard CRC algorithm described in the text. Write two programs, `generator` and `verifier`. The `generator` program reads from standard input a line of ASCII text containing an n-bit message consisting of a string of 0s and 1s. The second line is the k-bit polynomial, also in ASCII. It outputs to standard output a line of ASCII text with $n + k$ 0s and 1s representing the message to be transmitted. Then it outputs the polynomial, just as it read it in. The `verifier` program reads in the output of the generator program and outputs a message indicating whether it is correct or not. Finally, write a program, `alter`, that inverts 1 bit on the first line depending on its argument (the bit number counting the leftmost bit as 1) but copies the rest of the two lines correctly. By typing:

```
generator <file | verifier
```

you should see that the message is correct, but by typing:

```
generator <file | alter arg | verifier
```

you should get the error message.