BACHELOR OF COMPUTER APPLICATIONS

SEMESTER 3

DCA2104

BASICS OF DATA COMMUNICATION

Unit 13

Data Link Layer – Error Detection and Correction

Table of Contents

1. INTRODUCTION

This unit introduces the concept of data link layer- error detection and correction. Data link control functions include framing, flow and error control. Data link layer encapsulate each network layer packet with in a link layer frame before transmission over the link. Various error control measures are required to ensure a reliable transmission of data.

In this unit, we will discuss the different error detection and correction techniques namely one and two dimensional parity checks, hamming code and cyclic redundancy check. In the subsequent section we will discuss framing and different framing methods such as character stuffing, bit stuffing etc. In the last section we will discuss flow control and error control.

1.1 Objectives

After studying this unit, you should be able to:

- *describe error detection methods*
- *explain error correction methods*
- *describe framing*
- *explain flow control*
- *describe error control*

2. ERROR DETECTION AND CORRECTION

Error detection and correction are the two major issues to be dealt with when transmitting data over a network. There are two basic strategies for dealing with errors of data transmission. Both the strategies add redundant information to the data that is sent. One strategy is to include enough redundant information to enable the receiver to derive what the transmitted data must have been. This strategy uses *error-correcting codes*. Another strategy is to include only enough redundancy to allow the receiver to find out that an error has occurred and have it request a retransmission. This strategy *uses error-detecting* codes. The use of error-correcting codes is often referred to as *FEC (Forward Error Correction)*. We will discuss both error-correcting codes and error-detecting codes in the following sections.

Error- correcting codes are used on wireless links, which are very noisy and error prone when compared to optical fibers. However, over fiber or high quality copper, error rate is much lower, so error detection and retransmission is usually more efficient. *Parity* and *Cyclic Redundancy Checks (CRCs)* are error detecting codes.

2.1 One And Two Dimensional Parity Checks

Consider the first error-detecting code, in which a single parity bit is appended to the data. The parity bit is chosen so that the number of 1 bits in the codeword is even or odd. For example, when 1011010 is sent in even parity, a bit is added to the end to make it 10110100. With odd parity 1011010 becomes 10110101.

One difficulty with this scheme is that a single parity bit can only reliably detect a single-bit error in the block. If the block is badly garbled by a long burst error, the probability that the error will be detected is only 0.5, which is hardly acceptable. The odds can be improved considerably if each block to be sent is regarded as a rectangular matrix n bits wide and k bits high. Now, if we compute and send one parity bit for each row, up to k bit errors will be reliably detected as long as there is at most one error per row.

Another method which provides better protection against burst errors is known as interleaving. In this method, we can compute the parity bits over the data in a different order than the order in which the data bits are transmitted. In this case, we will compute a parity

bit for each of the n columns and send all the data bits as *k* rows, sending the rows from top to bottom and the bits in each row from left to right in the usual manner. At the last row, we send the n parity bits. This transmission order is shown in figure

12.1 for n=7 and k=7.

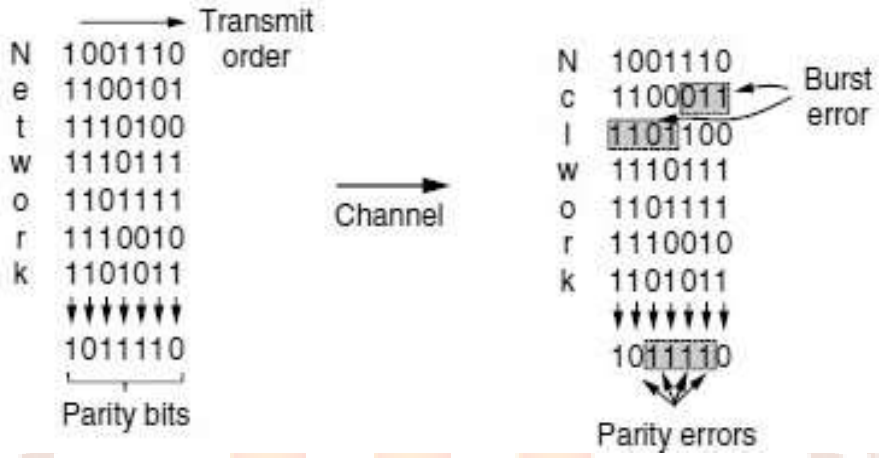

Figure 12.1: Interleaving of parity bits to detect a burst error.

Interleaving is a general technique to convert a code that detects isolated errors into a code that detects burst errors.

2.2 Cyclic Redundancy Check (CRC)

Another method of error detecting code is CRC which is also known as a polynomial code. This is a stronger kind of error-detecting code which is in widespread use at the link layer. Polynomial codes are based upon treating bit strings as representations of polynomials with coefficients of 0 and 1 only. A k-bit frame is regarded as the coefficient list for a polynomial with k terms, ranging from x^{k-1} to x^0. Such a polynomial is said to be of degree k – 1. The high-order (leftmost) bit is the coefficient of x^{k-1}, the next bit is the coefficient of x^{k-2}, and so on.

For example, 110001 has 6 bits and thus represents a six-term polynomial with coefficients 1, 1, 0, 0, 0, and 1: $1x^5 + 1x^4 + 0x^3 + 0x^2 + 0x^1 + 1x^0$. When the polynomial code method is employed, the sender and receiver must agree upon a *generator polynomial*, G(x), in advance. Both the high- and low order bits of the generator must be 1. To compute the CRC for some frame with m bits corresponding to the polynomial M(x), the frame must be longer than the

generator polynomial. The idea is to append a CRC to the end of the frame in such a way that the polynomial represented by the checksummed frame is divisible by G(x). When the receiver gets the checksummed frame, it tries dividing it by G(x). If there is a remainder, there has been a transmission error.

2.3 Hamming Code

A frame consists of 'm' data bits and 'r' redundant bits. In a *systematic code*, the *m* data bits are sent directly, along with the check bits, rather than being encoded themselves before they are sent. In a linear code, the r check bits are computed as a *linear* function of the m data bits. Let total length of a block be n *(n=m + r).* This is known as *(n, m)* code. An n-bit unit containing data and check bits is referred to as an *n* bit *codeword*. The *code rate*, or simply rate, is the fraction of the codeword that carries information that is not redundant, or *m/n.*

Before discussing error correction, it is necessary to understand what error is. Consider two codewords that may be transmitted or received- Say, 10001001 and 10110001. From these codewords, it is possible to determine how many corresponding bits differ. In this case, 3 bits differ. To determine how many bits differ, XOR the two codewords and count the number of 1 bits in the result. That is,

10001001
<u>10110001</u>
00111000

The number of bit positions in which two codewords differ is called the ***Hamming distance.*** Its significance is that if two codewords are a Hamming distance *d* apart, it will require *d* single-bit errors to convert one into the other.

In case of a 4 digit data, number of redundant/parity bits required is 3 (This can be calculated using the equation $2^p \geq m+p+1$, where m is number of data bits and p is the number of parity bits). Consider an example of a 4 bit data, the code is composed of four information bits and three parity bits. The left-most bit is designated as bit 1, next is bit 2 and so on…

bit 1, bit 2, bit 3, bit 4, bit 5, bit 6, bit 7

In this example, the parity bits are located in the positions that are numbered corresponding to ascending powers of 2 (1, 2, 4, 8…) as shown in Table 12.1.

Table 12.1: Assignment of Parity Bit values

Bit Designation	P_1,	P_2,	M_1,	P_3,	M_2,	M_3,	M_4
Bit Position	1	2	3	4	5	6	7
Binary position number	001	010	011	100	101	110	111

Here, P_n designates particular parity bit and M_n shows information bit. Note that, the binary position number of parity bit P_1 has a 1 for its right most digit. So this parity bit checks for all bit positions, including itself that have 1's in the same location in the binary position numbers. Therefore, parity bit P_1 checks bit positions 1, 3, 5 and 7. Similarly, binary position number of parity bit P_2 has a 1 for its middle bit, it checks all bit positions including itself, that have 1s in this same position. Therefore, P_2 checks bits 2, 3, 6 and 7. Similarly P_3 is in binary position 4 and has 1 for the leftmost bit, so it checks bit position 4, 5, 6 and 7.

Let's discuss one example, consider the data 1001 (information bits) being transmitted using even parity. Number of parity bits required is 3 (Let p=3, then $2^p \geq m+p+1$, i.e. $2^3=8$ and m+p+1=4+3+1=8. So three parity bits are sufficient) and total code bits = 4+3=7. Next step is to determine the parity bits. Bits P1 checks bit positions 1, 3, 5 and 7 and must be a 0 for there to be an even number of 1s in this group. Bit P_2 checks bit positions 2, 3, 6 and 7 and must be a 0 for there to be an even number of 1s in this group. Bit P_3 checks bit positions 4, 5, 6 and 7 and must be a 1 for there to be an even number of 1s in this group. These bits are entered in a table 12.2 and resulting combined code is 0011001.

Table 12.2: Representation of bits in case of information bits 1001

Bit Designation	P_1	P_2	M_1	P_3	M_2	M_3	M_4
Bit Position	1	2	3	4	5	6	7
Binary Position No.	001	010	011	100	101	110	111
Information Bits			1		0	0	1
Parity bits	0	0		1			

Here, codeword is 0011001. Suppose, there is an error occurred during transmission and the codeword received is 0010001. The receiver doesn't know what was transmitted and must look for proper parities to determine if the code is correct. Designate any error that has occurred in transmission if even parity is used. Table 12.3 shows the bit position table of received code.

Table 12.3: Bit Position table of Received code

Bit Designation	**P_1**	**P_2**	**M_1**	**P_3**	**M_2**	**M_3**	**M_4**
Bit Position	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Binary Position No.	**001**	**010**	**011**	**100**	**101**	**110**	**111**
Received code	0	0	1	0	0	0	1

Bit P_1 checks positions 1, 3, 5 and 7. There are two 1s in this group, so parity check is good. Bit P_2 checks positions 2, 3, 6 and 7. There are two 1s in this group. Parity check is good. Bit P_3 checks positions 4, 5, 6 and 7. There is one 1 in this group. Parity check is bad. So the error position code is 100 (binary four). This says that bit in position 4 is in error. It is a 0 and should be a 1. The corrected code is 0011001, which agrees with the transmitted code.

Self-Assessment Questions - 1

1. The use of error-correcting codes is often referred to as_____ .
2. Which layer is dealing with error detection and correction?
   a) Physical layer
   b) Datalink layer
   c) Network layer
   d) Transport layer
3. _____and_____ are error detecting codes.
4. A method which provides better protection against burst errors is known as _____.
5. Error detecting code is CRC which is also known as _____.

3. FRAMING

Data link layer break up the data packets received from the network layer to discrete frames, compute a token called checksum for each frame and include the checksum in the frame when transmitted. At the receiver side, checksum is recomputed. If there is a mismatch in the newly computed checksum and the one in the frame, the data link layer knows that an error has occurred and takes steps to deal it. There are four different methods for framing. They are:

- byte count
- flag bytes with byte stuffing
- flag bits with bit stuffing
- Physical layer coding violations.

The first framing method uses a field in the header to specify the number of bytes in the frame. At the receiver side, byte count is analysed to find out the number of bytes in the frame and hence the end of the frame is found. Figure 12.2 (a) depicts the byte count method. In this figure, there are four example frames of sizes 5, 5, 8, and 8 bytes, respectively.

One of the major problem with this algorithm is that the count can be misrepresented by a transmission error. For example, if the byte count of 5 in the second frame of Fig. 12.2 (b) becomes a 7 due to a single bit flip, the destination will get out of synchronization. It will then be unable to locate the correct start of the next frame. Since checksum is incorrect, destination can identify that an error occurred during transmission, but it has no way to detect the starting of the next frame. Sending back acknowledgement for retransmission is not effective because receiver cannot find out how many bytes to skip over to get to the start of the retransmission. For this reason, the byte count method is rarely used by itself.

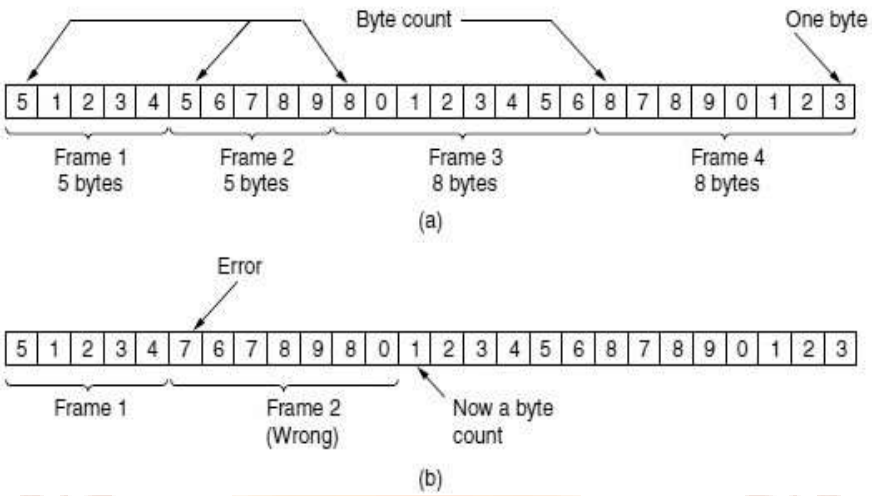

Figure 12.2: A byte stream (a) without errors (b) with one error

The second method is the use of a flag byte, which is special byte used as both the starting and ending delimiter. Two consecutive flag bytes indicate the end of one frame and the start of the next. Thus, if the receiver ever loses synchronization it can just search for two flag bytes to find the end of the current frame and the start of the next frame. The trouble in this case is that the flag byte can occur in data as well. One way to solve this problem is to have the sender's data link layer insert a special escape byte (ESC) just before each accidental flag byte in the data. Thus, a framing flag byte can be distinguished from one in the data by the absence or presence of an escape byte before it. The data link layer on the receiving end removes the escape bytes before giving the data to the network layer. This technique is called **byte stuffing**. If an escape byte occurs at the middle of data, which too stuffed with an escape byte. At the receiver, the first escape byte is removed, leaving the data byte that follows it. The process of byte stuffing is shown in figure 12.3.

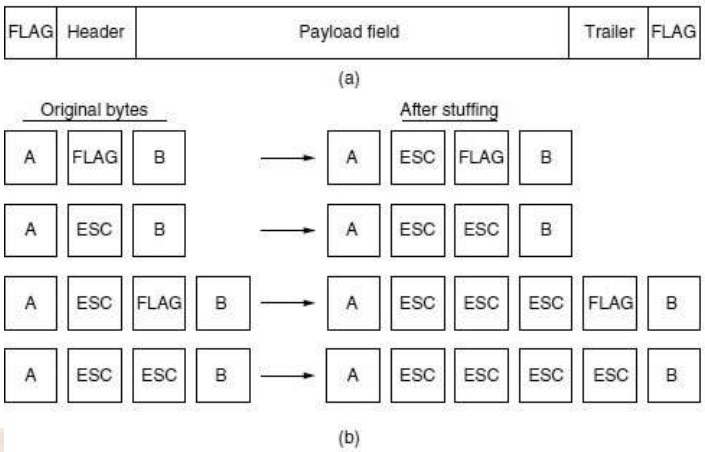

Figure 12.3: (a) A frame delimited by flag bytes. (b) Four examples of byte sequences before and after byte stuffing.

The third method is bit stuffing. In this scheme, framing has been done at the bit level, so frames can contain an arbitrary number of bits made up of units of any size. As we discussed in the previous chapter, bit stuffing was developed for the once popular HDLC (High level Data Link Control) protocol. In this method, each frame begins and ends with a special bit pattern, 01111110. This pattern is a flag byte. Whenever the sender's data link layer encounters five consecutive 1s in the data, it automatically stuffs a 0 bit into the outgoing bit stream. When the receiver sees five consecutive incoming 1 bits, followed by a 0 bit, it automatically deletes the 0 bit. Figure

12.4 gives an example of bit stuffing.

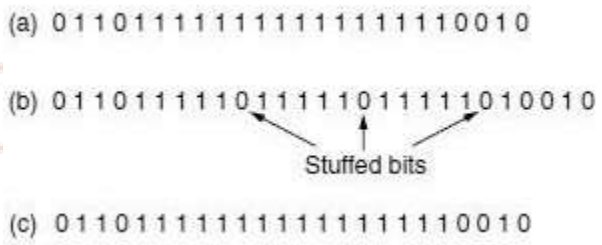

Figure 12.4: bit stuffing (a) The original data (b) the data as they appear on the line. (c) The data as they stored in receiver's memory after destuffing.

The last method of framing is physical layer coding violations, which is to use a shortcut from the physical layer. We can use reserved signals to indicate the start and end of frames. In effect, we are using coding violations to delimit frames. The beauty of this scheme is that, because they are reserved signals, it is easy to find the start and end of frames and there is no need to stuff the data. Many data link protocols use a combination of these methods for safety.

Self-Assessment Questions - 2

6. The data link layer on the receiving end removes the escape bytes before giving the data to the network layer, this technique is known as _____ .
7. In_____ framing has been done at the bit level.
8. _____ framing method use a shortcut from the physical layer.

4. FLOW AND ERROR CONTROL

In this section, we will discuss flow control and error control.

Flow Control

The measures of data flow control in a network should be managed by data link layer when there is a fast sender and slow receiver and if sender wants to send frames faster than the receiver which can accept them. This can occur when the sender is running on a fast, powerful computer and the receiver is running on a slow, low-end machine. There are two different flow control approaches, they are: feedback-based flow control and rate-based flow control. In feedback-based flow control, the receiver sends back information to the sender giving it permission to send more data. In rate- based flow control, the protocol has a built-in mechanism that limits the rate at which senders may send data, without using feedback from the receiver.

Error Control

At the destination, when data link layer receives bit stream from physical layer is not guaranteed to be error free. Some bits may have different values and the number of bits received may be less than, equal to, or more than the number of bits transmitted. It is the responsibility of data link layer to detect and correct errors.

The usual way to ensure reliable delivery is to provide the sender with feedback about what is happening at the other end of the line. The protocol calls for the receiver to send back special control frames bearing positive or negative acknowledgements about the incoming frames. If the sender receives a positive acknowledgement about a frame, it knows the frame has arrived safely. On the other hand, a negative acknowledgement means that something has gone wrong and the frame must be transmitted again. Another trouble is that the frame may vanish completely due to hardware troubles. In this case receiver will not react and will not give any feedback. Similarly, if the acknowledgement frame is lost, the sender will not know how to proceed. This problem is handled by introducing timers into the data link layer. When the sender transmits a frame, it generally starts a timer. The timer will wait for a period which is equal to the time required for the frame to reach the destination, be

processed there and have the acknowledgement propagate back to the sender. In normal case, if the frame is correctly received, the acknowledgement will get back before the timer runs out, in which case the timer will be cancelled. But, if either the frame or the acknowledgement is lost, the timer will go off after a specific period known as round trip time (which is equivalent to the time required for the frame to reach destination plus the time required for the acknowledgement to reach the sender), so that sender will be alert about the problem and it will send the frame again.

However, when frames may be transmitted multiple times there is a danger that the receiver will accept the same frame two or more times and pass it to the network layer more than once. To prevent this from happening, it is necessary to assign sequence numbers to outgoing frames, so that the receiver can distinguish retransmissions from originals. Hence, this is one of the major duty of data link layer to manage the timers and sequence numbers to ensure that each frame is ultimately passed to the network layer at the destination exactly once.

Self-Assessment Questions - 3

9. The measures of data flow control in a network should be managed by _____layer.
10. Two different flow control approaches are_____ and _____.

5. SUMMARY

Let us recapitulate the important concepts discussed in this unit:

- Data link control functions include framing, flow and error control.
- One strategy is to include enough redundant information to enable the receiver to derive what the transmitted data must have been. This strategy uses error-correcting codes.
- Another strategy is to include only enough redundancy to allow the receiver to find out that an error has occurred and have it request a retransmission. This strategy uses error-detecting codes.
- Another method of error detecting code is CRC which is also known as a polynomial code.
- The number of bit positions in which two codewords differ is called the Hamming distance. Its significance is that if two codewords are a Hamming distance d apart, it will require d single-bit errors to convert one into the other.
- Data link layer break up the data packets received from the network layer to discrete frames, compute a token called checksum for each frame and include the checksum in the frame when transmitted.
- There are two different flow control approaches, they are feedback- based flow control and rate-based flow control.
- In feedback-based flow control, the receiver sends back information to the sender giving it permission to send more data.
- In rate-based flow control, the protocol has a built-in mechanism that limits the rate at which senders may send data, without using feedback from the receiver.

6. TERMINAL QUESTIONS

1. Explain different error detecting codes.
2. Describe hamming code.
3. Explain framing and four different framing methods.
4. Write short note on error control.
5. Explain flow control.

7. ANSWERS

Self-Assessment Questions

1. Forward error correction
2. (b) datalink layer
3. Parity, cyclic redundancy check (CRC)
4. Interleaving
5. Polynomial code
6. Byte stuffing
7. Bit stuffing
8. Physical layer coding violations
9. Datalink layer
10. Feedback based, rate-based

Terminal Questions

1. *Parity* and *Cyclic Redundancy Checks (CRCs)* are error detecting codes. Consider the first error-detecting code, in which a single parity bit is appended to the data. The parity bit is chosen so that the number of 1 bits in the codeword is even or odd. (Refer section 2.1 and 2.2 for detail).
2. Hamming code is an error correcting code. Before discussing error correction, it is necessary to understand what error is. Consider two codewords that may be transmitted or received- Say, 10001001 and 10110001. (Refer section 2.3 for detail).

3. Data link layer break up the data packets received from the network layer to discrete frames, compute a token called checksum for each frame and include the checksum in the frame when transmitted. (Refer section 3 for detail).
4. At the destination, when data link layer receives bit stream from physical layer is not guaranteed to be error free. Some bits may have different values and the number of bits received may be less than, equal to, or more than the number of bits transmitted. It is the responsibility of data link layer to detect and correct errors. (Refer section 4 for detail).
5. The measures of data flow control in a network should be managed by data link layer when there is a fast sender and slow receiver and if sender wants to send frames faster than the receiver which can accept them. (Refer section 4 for detail)

References:

1. Behrouz A. Forouzan, Sophia Chung Fegan, "Data Communications and Networking", Fourth edition.
2. William Stallings, "Data and Computer Communications", Sixth edition, Pearson Education, Delhi, 2002.
3. Taub and Schilling, "Principles of Communication Systems", Tata Mc Graw Hill, Delhi, 2002.
4. S. Tanenbaum, "Computer Networks", Pearson Education, Fourth Edition.
5. N. Olifer, V. Olifer, "Computer Networks: Principles, technologies and Protocols for Network design", Wiley India Edition, First edition.
6. Simon Poulton (2003), packet switching and X.25 Networking, Pitman publishing.
7. Walrand, P. Varaiya, "high performance communication networks", Morgan kaufmann.