COMPUTER NETWORKS

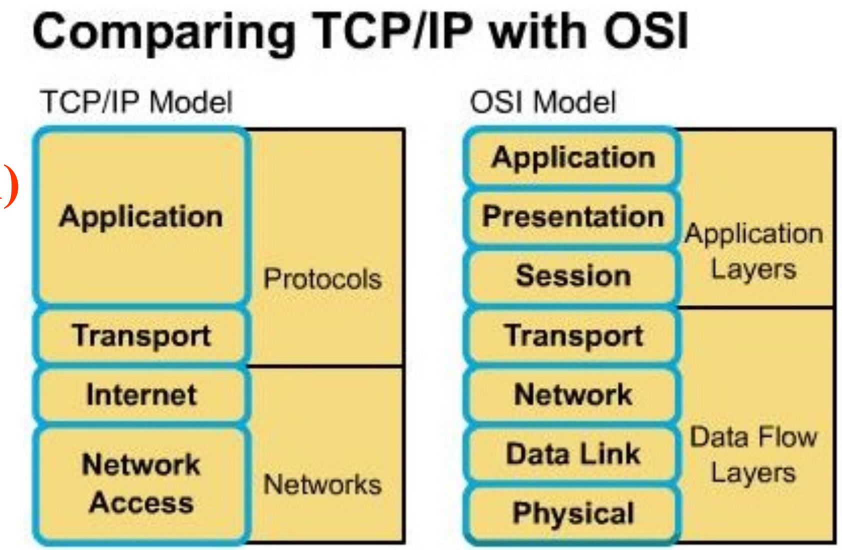
- Chapter 3. Data Link Layer 1

王昊翔
WANG Haoxiang
hxwang@scut.edu.cn

School of Computer Science & Engineering
国家双语教学试点项目 广东省精品课

Contents presented in Chap1&2

- Chapter 1
  - Services vs. Protocols
  - Reference Models(Encapsulation)
  - Example Networks
  - Networks Standardization
- Chapter 2
  - Theoretical Basis
  - Three transmission media
    - Guided transmission
    - Wireless transmission
    - Communication satellites
  - Three communication system
    - Public Switched Telephone Network



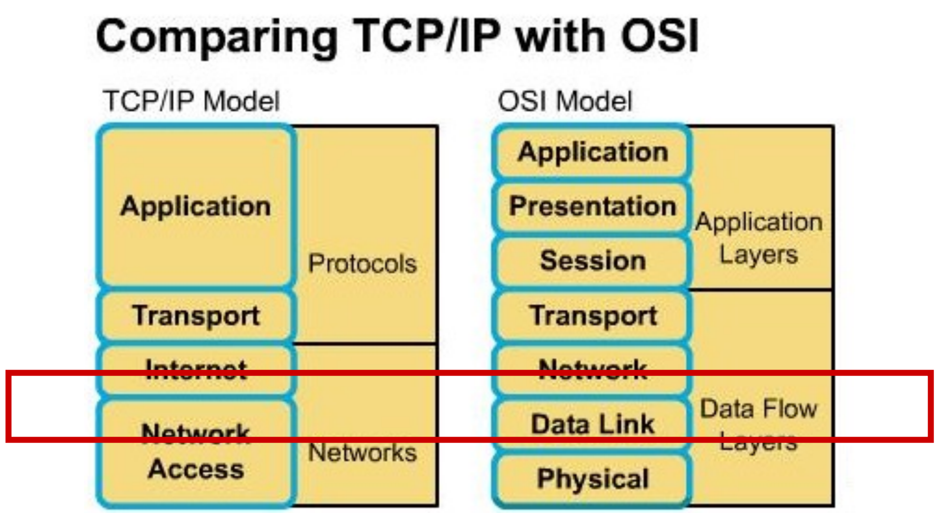
5	Application layer
4	Transport layer
3	Network layer
2	Data link layer
1	Physical layer

Main object of the chapter3

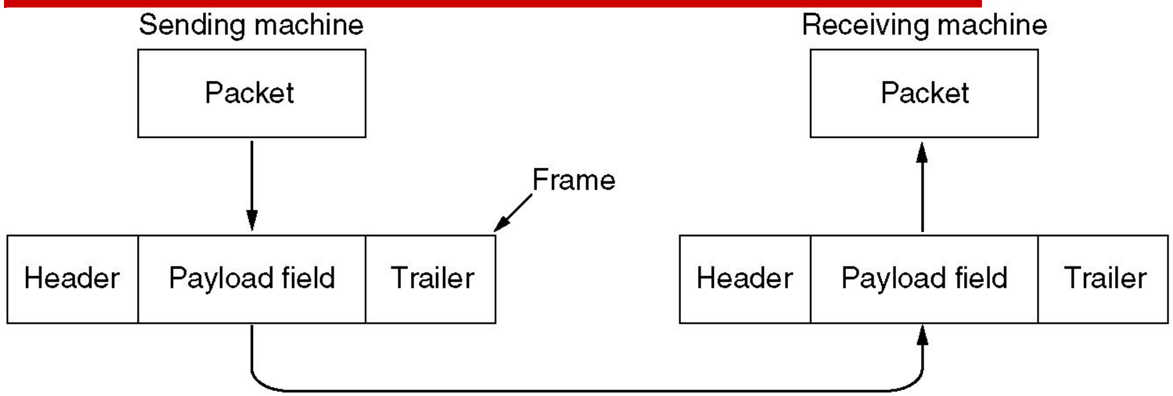
- The DLL is responsible for taking the packets of information that it receives from the Network Layer and putting them into frames for transmission.
- Each frame holds the payload plus a header and a trailer (overhead).
- It is the frames that are transmitted over the physical layer.
- Achieving reliable, efficient communication between two adjacent machines.

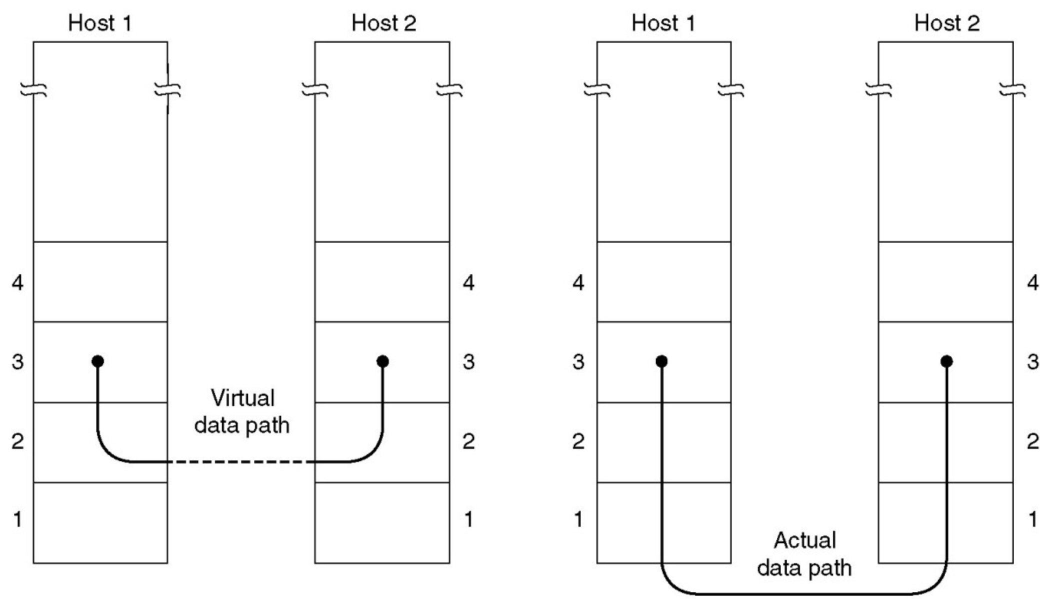
Main Functions of DLL

- Provide a well-defined service interface to the network layer.
- Deal with transmission errors.
- Regulate the flow of data , so that slow receivers are not swamped.



Relationship between packets and frames





(a) (b)

Outline

- Data Link Layer Design Issues
- Error Detection and Correction
- Elementary Data Link Protocols
- Sliding Window Protocols
- Example Data Link Protocols

Contents of this lecture

- Overview of data link layer
- Learn Framing methods
- Learn error-detection and error-control
  - Hamming code（海明码）
  - Cyclic Redundancy Check（循环冗余码CRC）

Service provided by DLL

- The data link layer can offer many kinds of service.
- The actual offered services can vary from system to system.
- Three common services:
  - Unacknowledged connectionless service.
    - 无确认的无连接服务
  - Acknowledged connectionless service.
    - 有确认的无连接服务
  - Acknowledged connection-oriented service.
    - 有确认的面向连接服务

Unacknowledged connectionless service

- The source machine send independent frames to the destination machine, and there is no acknowledgement from the destination machine.
- No logical connection is established beforehand or released afterward.
- The DLL will make no attempt to detect the loss of or recover a lost frame.
- This service is useful for low error rate networks and for real-time traffic where late data is worse than no data.

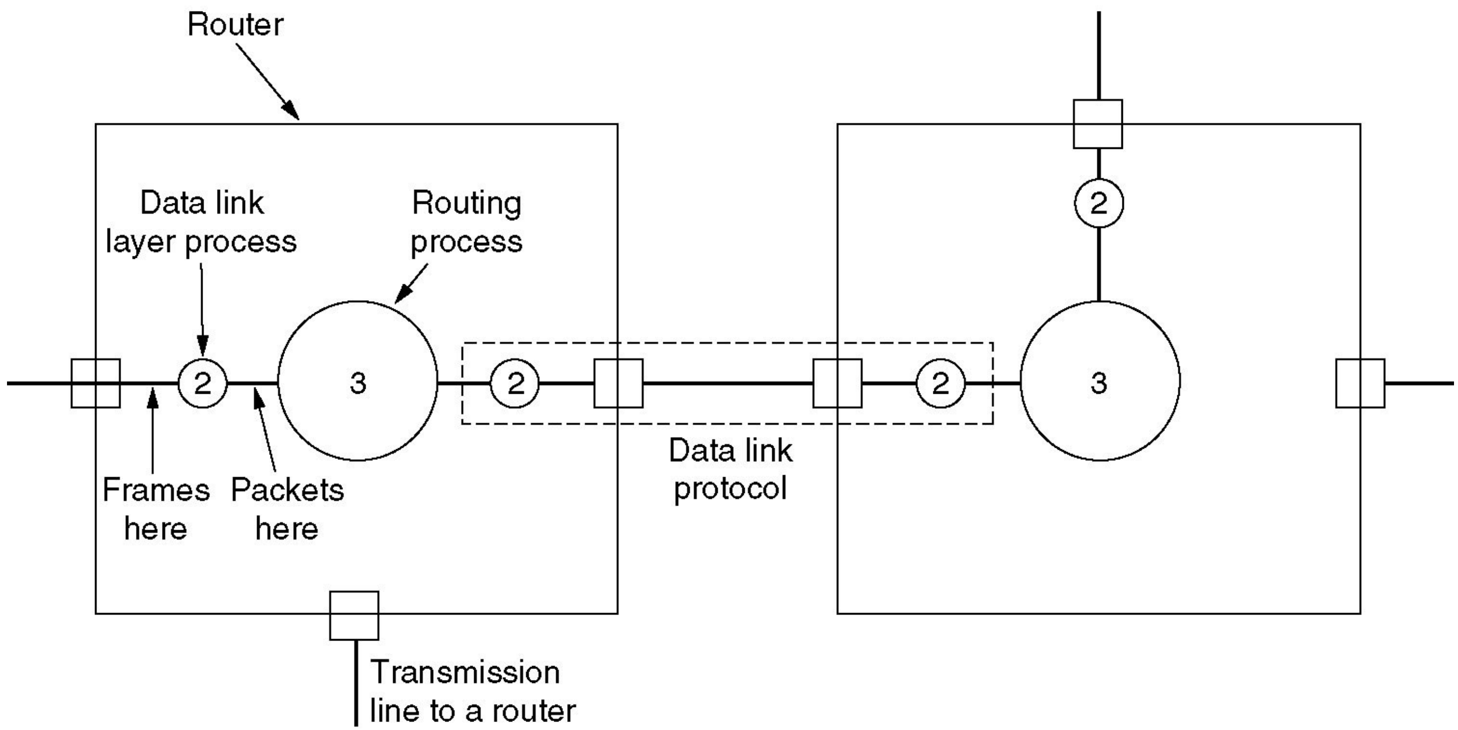
Acknowledged Connectionless Service

- The receiver acknowledges the arrival of each frame.
  - If it hasn’t arrived correctly (or within a specified time interval), it can be resent.
- This is a useful service when the connection is unreliable (such as wireless systems)
- There is no requirement for such an acknowledgement service to be implemented by the data link layer.

Acknowledged Connection-Oriented Service

- A connection is established between the two machines, and the frames are then transmitted.
- Each frame sent over the connection is numbered and each frame is acknowledged.
- The frames are guaranteed to arrive only once and in order.
- The connection is released once the communication is complete.
- This is the same as a “reliable” bit stream.

Data Flow Over Two Routers



Framing

- The data link layer must use the service provided by the physical layer in order to provide service to the network layer.
- The Physical Layer is only able to put a raw bit stream on the transmission media.
- Bit stream is not guaranteed to be error free.
- It is up to the data link layer to detect and, if necessary, correct errors.

Framing(cont'd)

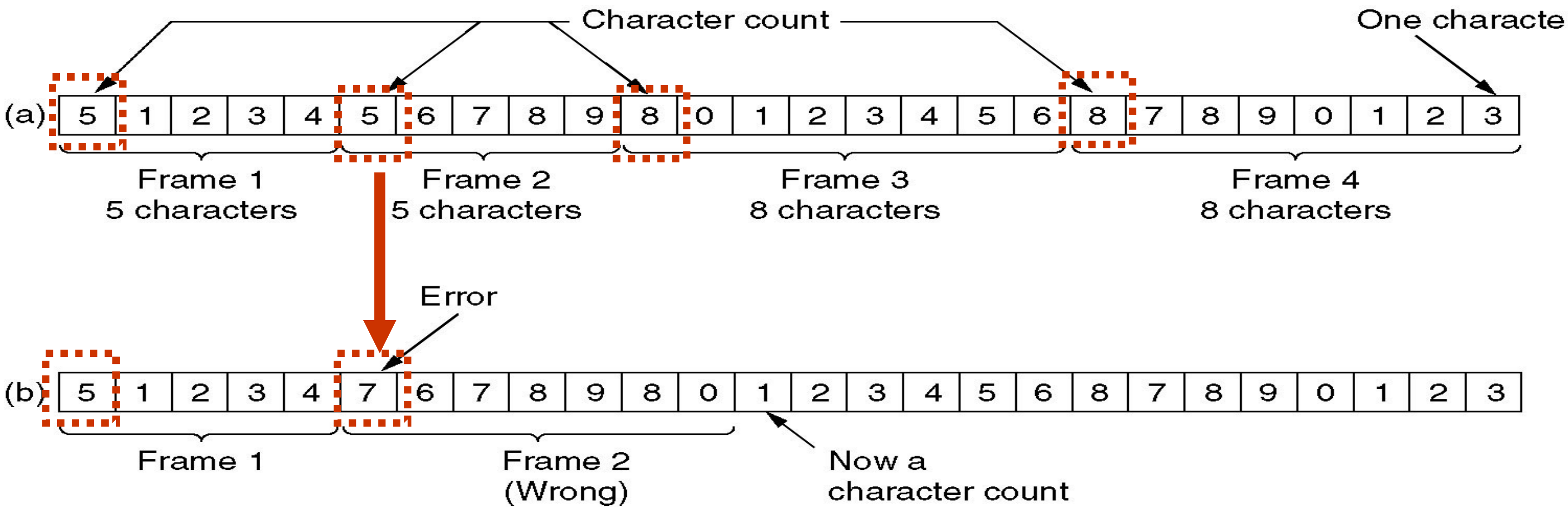
- The DDL can be able to break up the bit stream into discrete frames.
- Compute the checksum for each frame and the checksum is recomputed when a frame arrives at the destination.
- Breaking the bit stream up into frames is somewhat difficult.
  - Time gaps
- We need to look at other methods of denoting the start and finish of a frame.

Four framing method

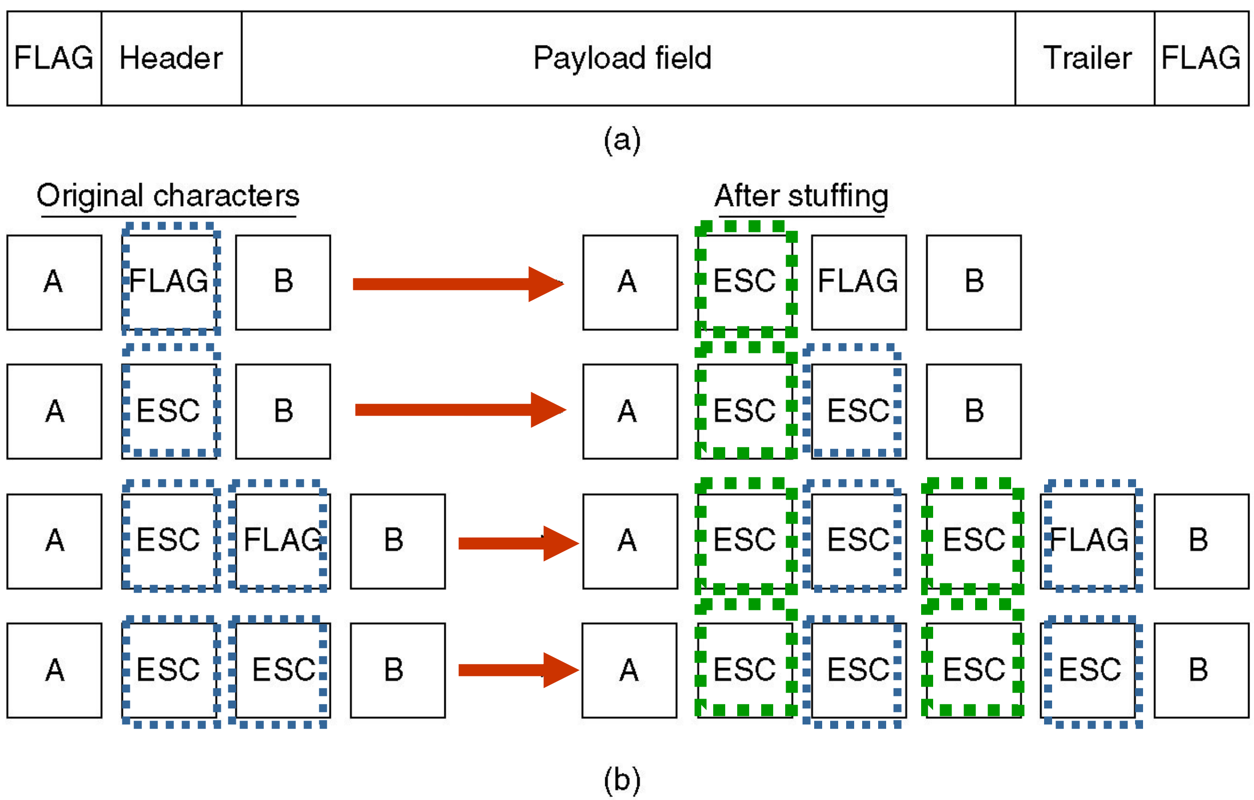
- Character count（字符计数法）
- Flag bytes with byte stuffing（带字节/字符填充的分界符法）
- Starting and ending flags, with bit stuffing（带位填充的分界标志法）
- Physical layer coding violations（物理层编码违例法）

Character count

- Insert time gaps between frames,impossible
- Uses a field in the header to specify the number of characters in the frame
- Problem



Flag Bytes with byte stuffing



Flag Bytes with bit stuffing

- Allows data frames to contain an arbitrary number of bits and allows character codes with an arbitrary number of bits per character.
- Each frame begins and ends with a special bit pattern, 01111110 (in fact, 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.
- With bit stuffing, the boundary between two frames can be unambiguously recognized by the flag pattern.

Example

(a) 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0

(b) 0 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 0 0 1 0

Stuffed bits

(c) 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0

Physical layer coding violations

- 物理层编码违例法
- Only applicable to networks in which the encoding on the physical medium contains some redundancy.
- For example, some LANs encode 1 bit of data by using 2 physical bits. Normally, a 1 bit is a high-low pair and a 0 bit is a low-high pair.
- Most DLL protocols use a combination of character count with another method for extra safety. This increases the chances of catching an error.

Error Control

- We use byte stuffing, bit stuffing and checksum as a method for detecting and determining errors in the data that we send.
- We also have to deal with making sure that the frames make it to their destination.
- The receiver sends back a control frame acknowledging the received frame and the condition of the frame.
- A timeout can occur if the acknowledgement doesn't arrive, resulting in the frame being resent.
  - Timeout interval

Error Control (cont'd)

- Resending the frame can also cause problems – what happens when the same frame is received twice or more times?
- We can also sequentially number the frames to prevent this problem.
- There are many different ways to do this type of error control (and it can be done at different levels as well).
- Managing the timers and sequence numbers are important parts of the data link layer's duties.

Flow Control

- We must deal with the issue where the sender is sending data at a higher rate than the receiver can receive the data.
- There are two approaches to this problem:
  - feedback-based flow control
    - feedback is used to tell the sender how the receiver is doing or to send another frame
  - rate-based flow control
    - the transfer rate is fixed by the sender
    - this is never used in the DLL

Why do we need Error Detection And Correction?

- The local loops are still analog twisted copper pairs and errors are still common.
- Wireless communication is becoming more common, and the error rates are orders of magnitude worse than on the interoffice fiber trunks.
- Transmission errors are going to be with us for many years to come.

Types of Error

- Errors come in bursts
- Independent single-bit errors
- Example
  - block size is 1000 bits
  - error rate is 0.001 per bit
- Burst errors are much harder to correct than isolated errors

Error Processing

- **Error-correcting codes**
  - **Include enough redundant information along with each block of data sent.**
  - **The receiver can deduce the transmitted data**
  - **The use of error-correcting codes is often referred to as forward error correction(前向纠错).**
- **Error-detecting codes**
  - **Include only enough redundancy**
  - **Allow the receiver to deduce that an error occurred, but not correct error, and just have it request a retransmission.**

Error Processing (cont'd)

- Each of the two techniques is applicable to different circumstances.
- Error-correcting code
  - The overhead is high but it reduces the need to resend frames.
  - best suited for networks with high error (wireless).
- Error-detecting code
  - suite for highly reliable channel, such as fiber
  - just retransmit when errors are found

Presentation

- Error correction
- Hamming encoding

- detect d bits errors
- correct d bits errors

Do exercise（1/2）

- **Original data:** 10101111,even**-parity Hamming code, if hope to correct one single error, What is the Hamming code for it?**
- Solution： m=8， According $(m+r+1)\le 2^r$:

r=4， ??1?010?1111

P1=B1⊕B3⊕B5⊕B7⊕B9⊕B11 =∑(0,1,0,0,1,1)=1

P2=B2⊕B3⊕B6⊕B7⊕B10⊕B11=∑(0,1,1,0,1,1)=0

P3=B4⊕B5⊕B6⊕B7 ⊕B12 =∑(0,0,1,0,1)=0

P4=B8⊕B9⊕B10⊕B11⊕B12 =∑(0,1,1,1,1)=0

So, Hamming code is: 101001001111

Do exercise（2/2）

- **All conditions is as above, if receiver has received a codeword like: 1 0 0 1 1 0 0 0 1 1 0 0 (m=8,r=4）, Question: Is the codeword is correct or not? What is the corresponding correct one if wrong?**

- **Solution:**

$$P1=B1\oplus B3\oplus B5\oplus B7\oplus B9\oplus B11 =\sum(1,0,1,0,1,0)=1$$

$$P2=B2\oplus B3\oplus B6\oplus B7\oplus B10\oplus B11=\sum(0,0,0,0,1,0)=1$$

$$P3=B4\oplus B5\oplus B6\oplus B7 \oplus B12 =\sum(1,1,0,0,0)=0$$

$$P4=B8\oplus B9\oplus B10\oplus B11\oplus B12 =\sum(0,1,1,0,0)=0$$

So, Counter=1+2=3， the 3rd bit is wrong, correct one is:

1 0 1 1 1 0 0 0 1 1 0 0

Error Detection

- Error-detecting codes **only include enough data to let the receiver determine whether the data is faulty.**
- **If the error rate of physical link is much lower, error detection and retransmission is usually more efficient.**
  - **copper wire or fiber**

How efficient, an example

- For comparison, consider a channel with error rate of 10^{-6} per bit. Let block size be 1000 bits.
  - To correct a single error (by Hamming code), 10 check bits per block are needed. To transmit 1000 blocks, 10,000 check bits (overhead) are required.
  - To detect a single error, a single parity bit per block will suffice. To transmit 1000 blocks, only one extra block (due to the error rate of 10^{-6} per bit) will have to be retransmitted, giving the overhead（开销） of only 2001 (= 1000*1 + 1001) bits.

Polynomial Code(多项式编码)

- Also known as a CRC (Cyclic Redundancy Check，循环冗余校验码).
- Based upon treating bit strings as representations of polynomials with coefficients of 0 and 1
  - Example: 110001
  - five-degree six-term polynomial（6项5阶多项式）
  - $1*x^5 + 1*x^4+0*x^3+0*x^2+0*x^1+1*x^0=x^5+x^4+1$
- Polynomial arithmetic is done modulo 2. Both addition and subtraction are identical to EXCLUSIVE OR（等同于异或）:
  - 10011011 01010101
  - + 11001010 - 10101111
  - --------------- ---------------
  - 01010001 11111010

What is modulo 2?

- **Modulo 2 addition & substraction: XOR logic**

  $$0\oplus 0=0;\ 0\oplus 1=1;$$
  $$1\oplus 0=1;\ 1\oplus 1=0.$$

– **Modulo 2 mltiplication:**

```
    1 0 1 0
×     1 0 1
    -------
    1 0 1 0
  0 0 0 0
1 0 1 0
-----------
1 0 0 0 1 0
```

-- **Modulo 2 division:**

```
          1 0 1
       __________
1 0 1/ 1 0 0 0 0
       1 0 1
       -----
         0 1 0
         0 0 0
         -----
           1 0 0
           1 0 1
           -----
             0 1
```

Polynomial Code (cont'd)

- The basic idea of the CRC method:
  - The sender and receiver agree upon a **generator polynomial（生成多项式）**, G(x), in advance.
  - The sender appends a **checksum** 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 frame, it tries dividing it by the same G(x). If there is a remainder, there must have been an error and a retransmission will be requested.

Algorithm For Computing Checksum

1. Let r be the degree of G(x). Append r zero bits to the low-order end of the frame so it now contains m + r bits and corresponds to the polynomial $x^r M(x)$.
2. Divide the bit string corresponding to G(x) into the bit string corresponding to $x^r M(x)$, using modulo 2 division.
3. Subtract the remainder (which is always r or fewer bits) from the bit string corresponding to $x^r M(x)$ using modulo 2 subtraction. The result is the checksummed frame to be transmitted. Call its polynomial T(x).

An example of CRC

- **Frame: 1101011011（m=10）**

  $M(x)= x^9+x^8+x^6+x^4+x^3+x+1$
- $G(x)=x^4+x+1$ （r=4阶）
- $x^4M(x)$ （相当于在原码字后加r个0）

  $=x^4(x^9+x^8+x^6+x^4+x^3+x+1)$

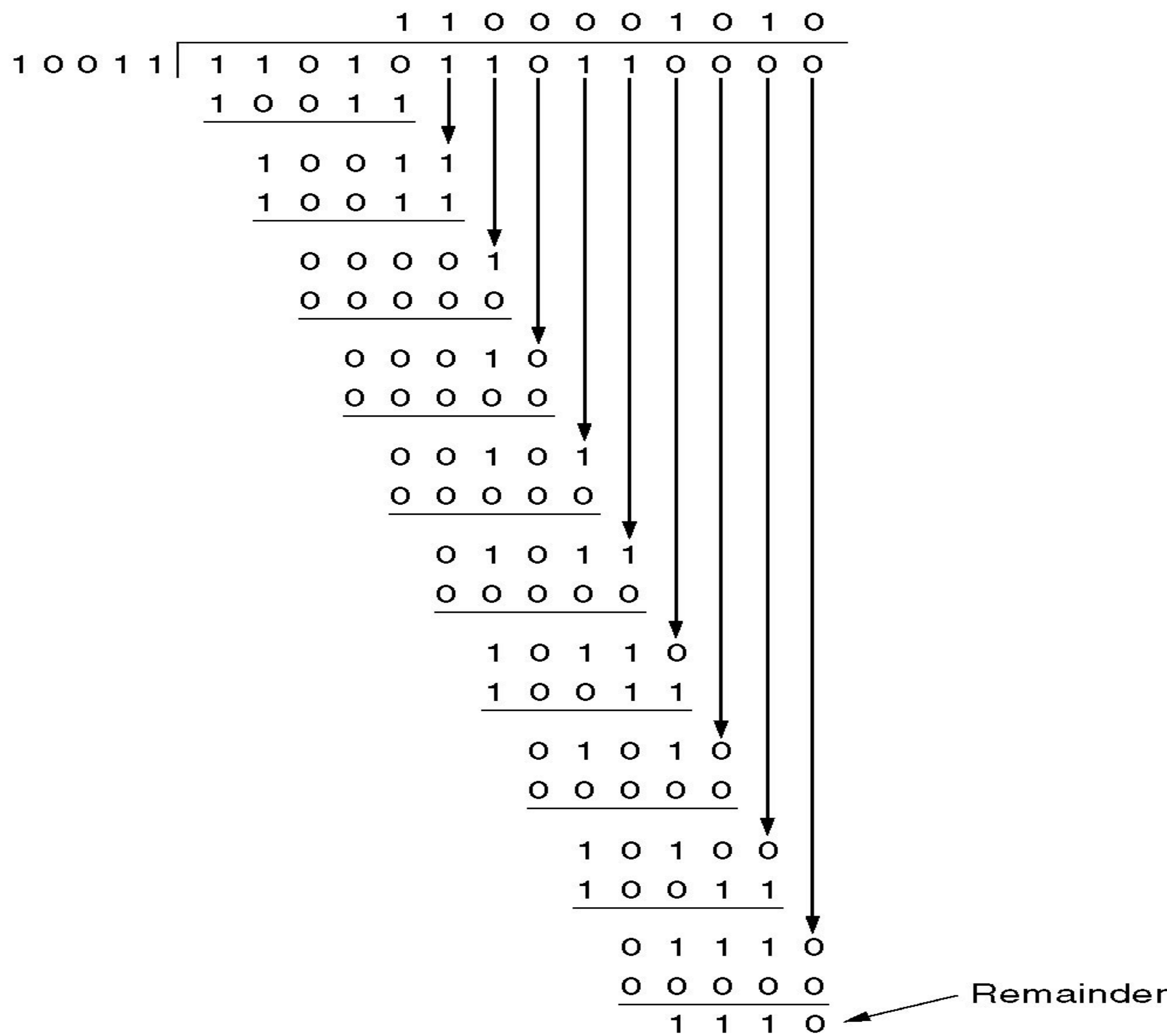
  $=x^{13}+x^{12}+x^{10}+x^8+x^7+x^5+x^4$
- $x^4M(x)$- reminder$(x^4M(x)/ G(x))$=?

An

Frame : 1 1 0 1 0 1 1 0 1 1

Generator: 1 0 0 1 1

Message after 4 zero bits are appended: 1 1 0 1 0 1 1 0 1 1 0 0 0 0



Transmitted frame: 1 1 0 1 0 1 1 0 1 1 1 1 1 0

Summary of this lecture

- Learn functions of DLL
- Learn and Master framing method
- Master error detection and correction methods
  - Hamming code海明码
  - Polynomial code(CRC)