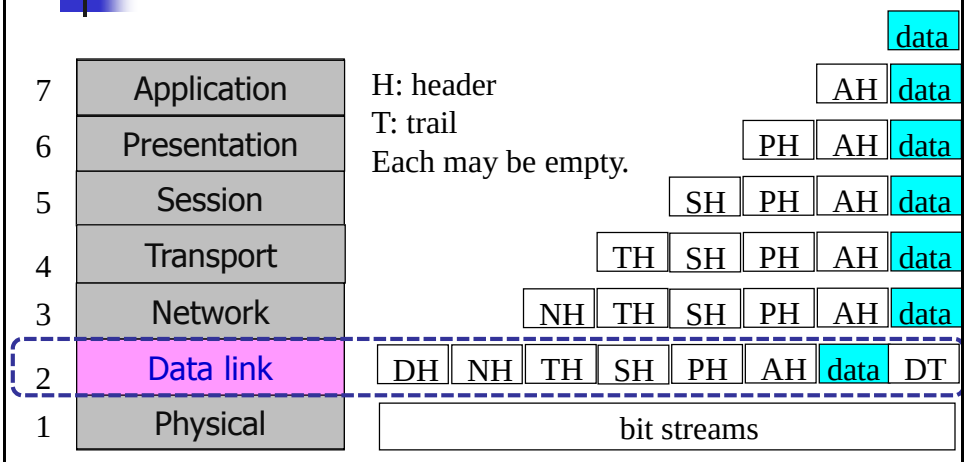
Chapter 3
The Data Link Layer

陳瑞奇(Rikki)
亞洲大學資訊工程學系

Adapted from Computer Networks,
Andrew S. Tanenbaum, Vrije University, Netherlands
& Computer Networking: A Top Down Approach,
Jim Kurose, Keith Ross

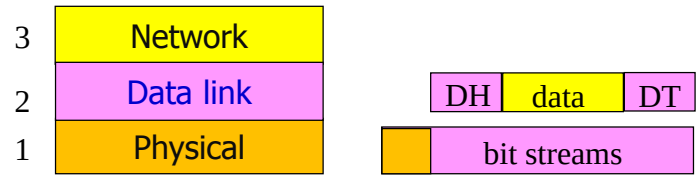
The Data Link Layer



OSI Reference Model

Data Link Layer Design Issues

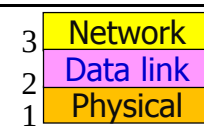
- Network layer services



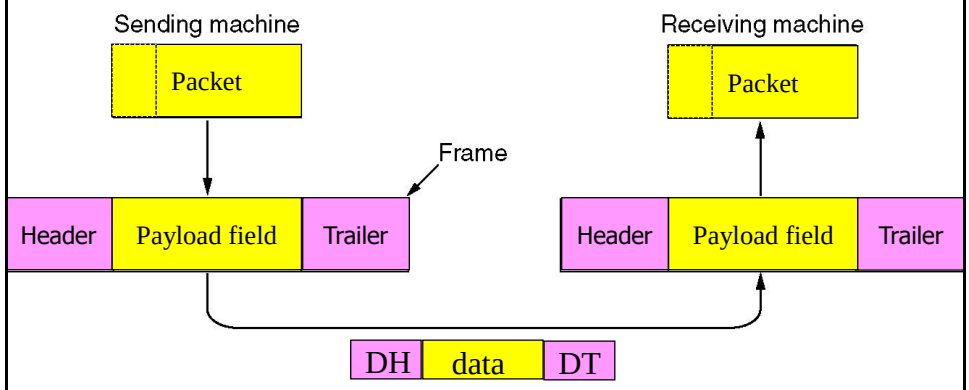
- Framing
- Error Control
- Flow Control

3

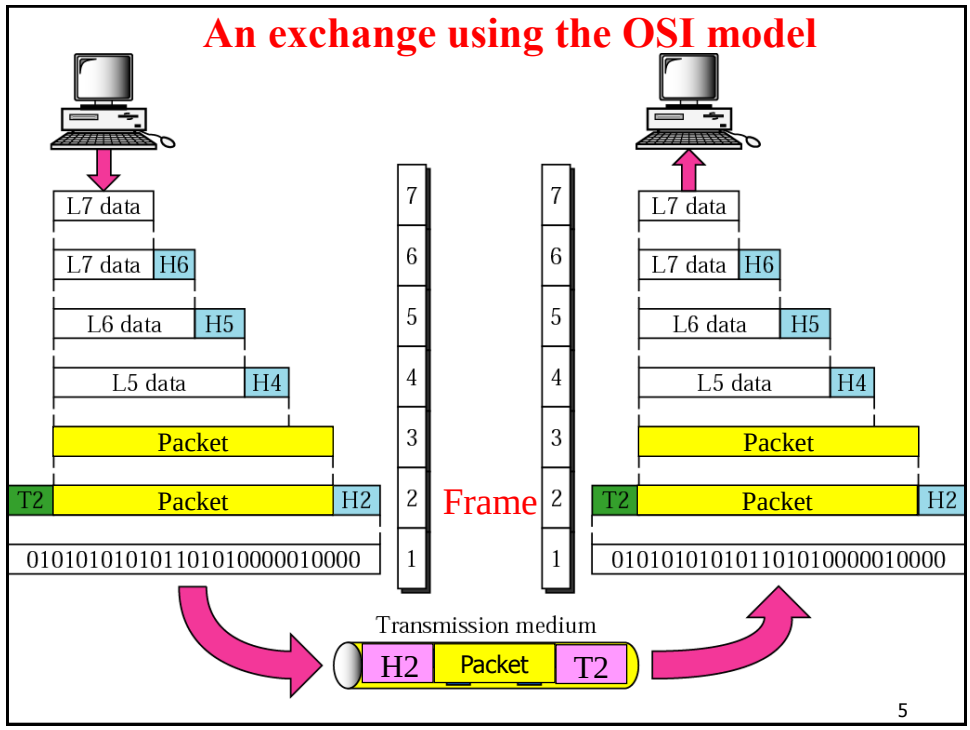
Packets and Frames

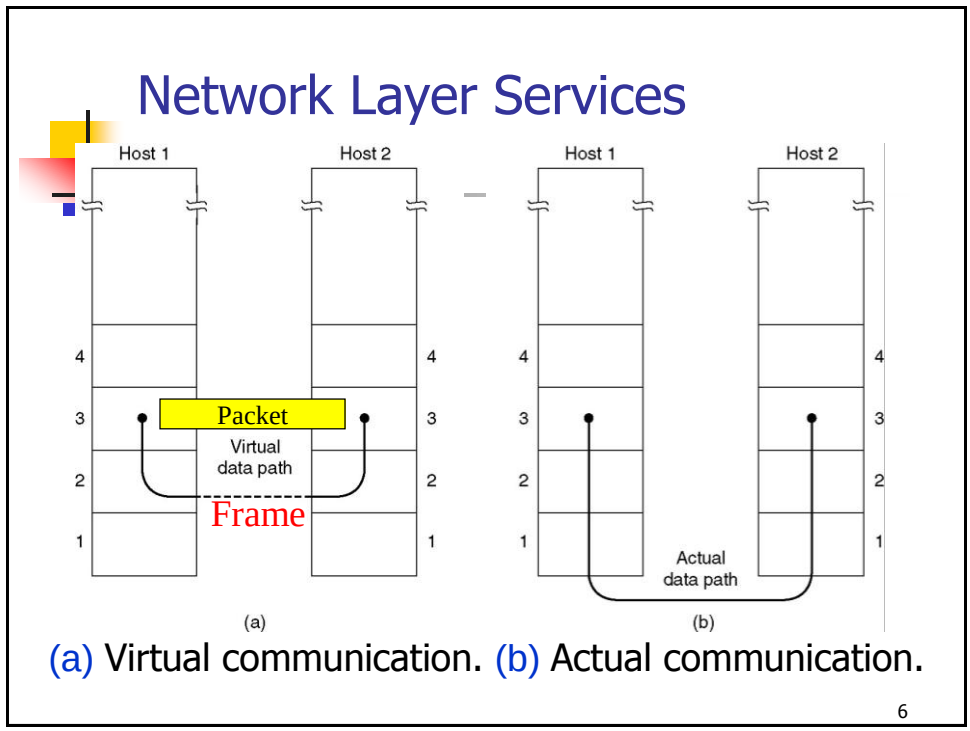


Relationship between packets and frames.

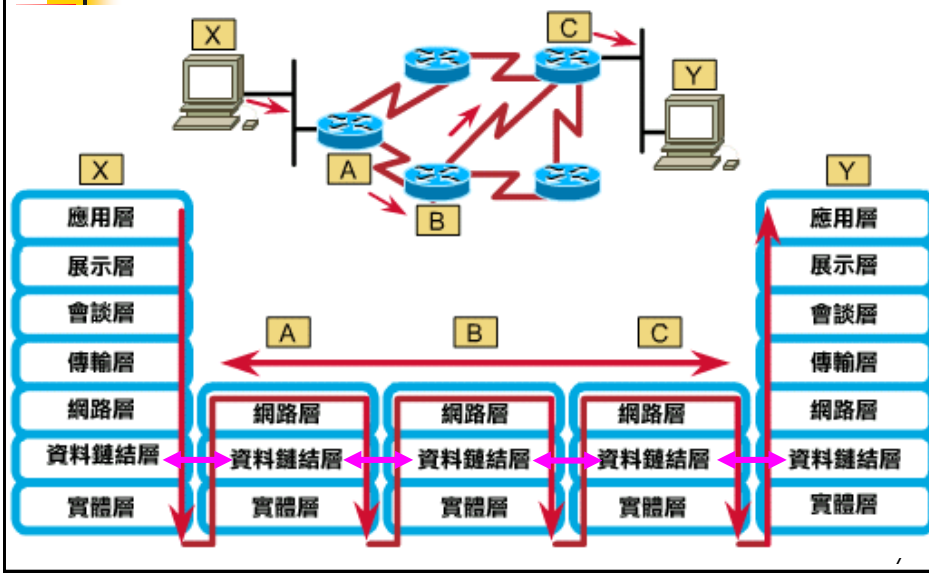


4


An exchange using the OSI model
L7 data
L7 data H6
L6 data H5
L5 data H4
Packet
T2 Packet H2
0101010101011010100000010000
7
6
5
4
3
2
1
Frame
Transmission medium
H2 Packet T2
5


Network Layer Services
Host 1
Host 2
Packet
Virtual data path
Frame
Actual data path
(a)
(b)
(a) Virtual communication. (b) Actual communication.
6

Services Provided to Network Layer



Possible Services Offered

(like Packet Switching)

1. Unacknowledged connectionless service: Ethernet

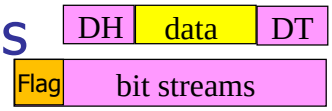
2. Acknowledged connectionless service: WLAN, WiFi

3. Acknowledged connection-oriented service: ATM, Frame Relay

(like Circuit Switching)

8

Framing methods

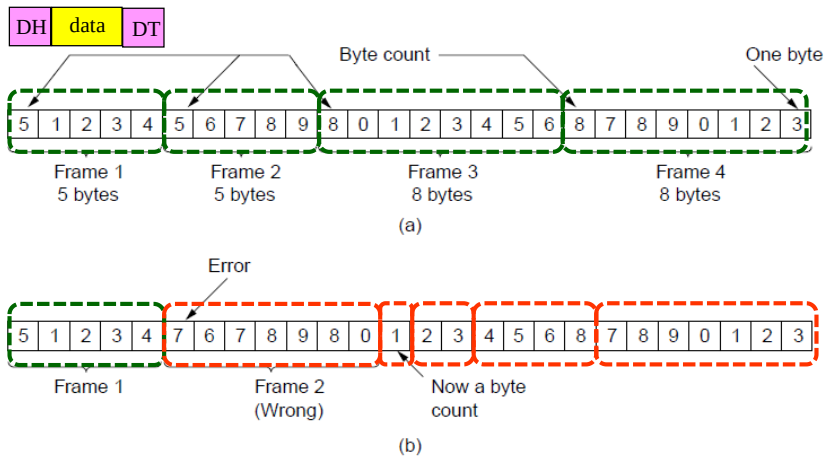


- Too risky to count on timing to mark the start and end of each frame
- Other methods:
  1. Byte Count (rarely used anymore)
  2. Flag bytes with byte stuffing (PPP)
  3. Flag bits with bit stuffing
  4. Physical layer coding violations

9

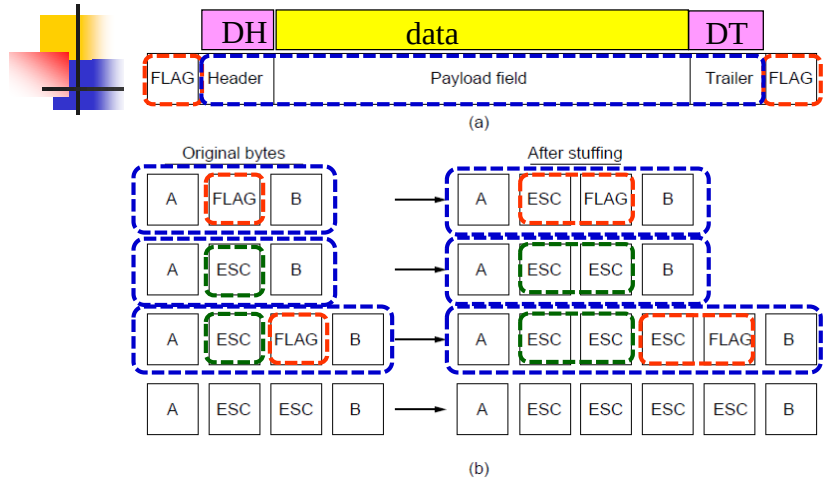
Framing (1): Byte Count

A byte stream. (a) Without errors. (b) With one error.



10

Framing (2):Flag bytes with byte stuffing

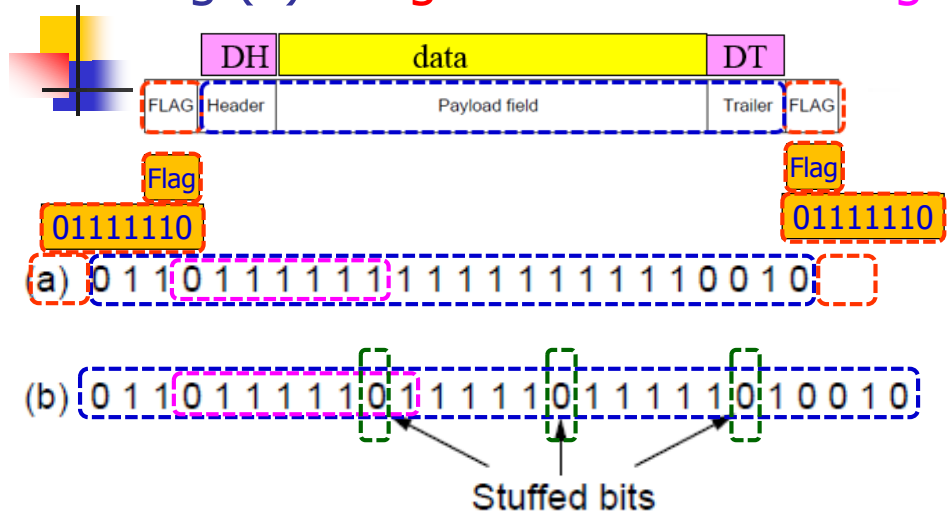


a) A frame delimited by flag bytes.

b) Four examples of byte sequences before and after byte stuffing.

11

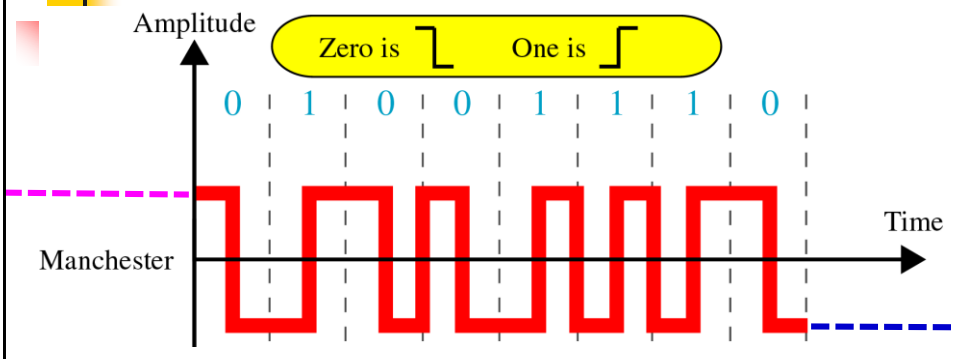
Framing (3): Flag bits with bit stuffing



Bit stuffing. (a) The original data. (b) The data as they appear on the line.

12

Framing (4): Physical layer coding violations



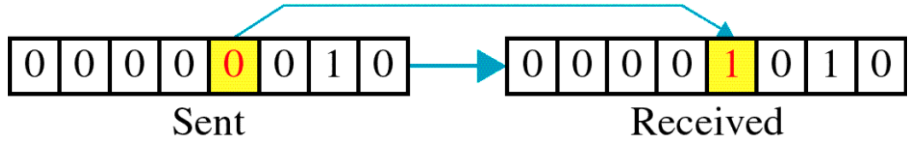
The combinations of low-low and high-high which are not used for data may be used for marking frame boundaries.

13

Error Control Process (1)

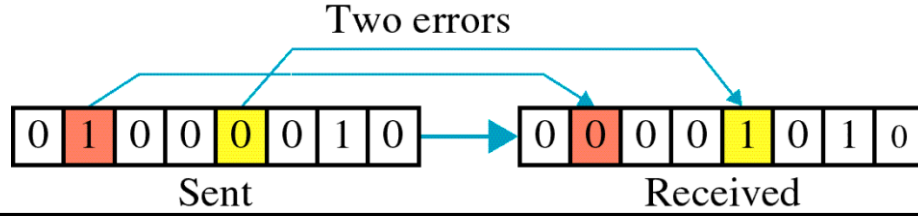
Single-bit error

0 changed to 1



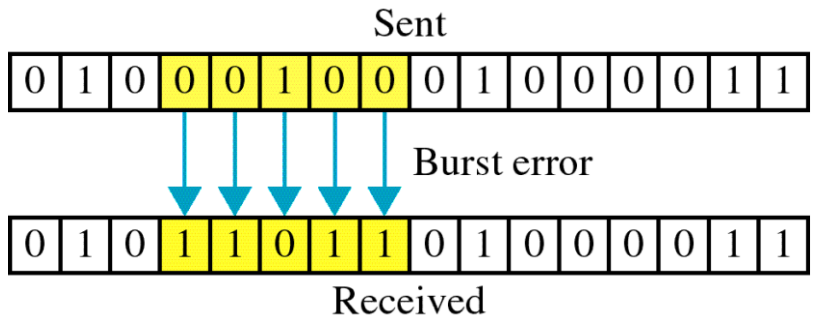
Multiple-bit error

Two errors



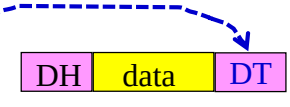
Error Control Process (2)

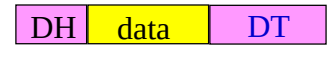
Burst error



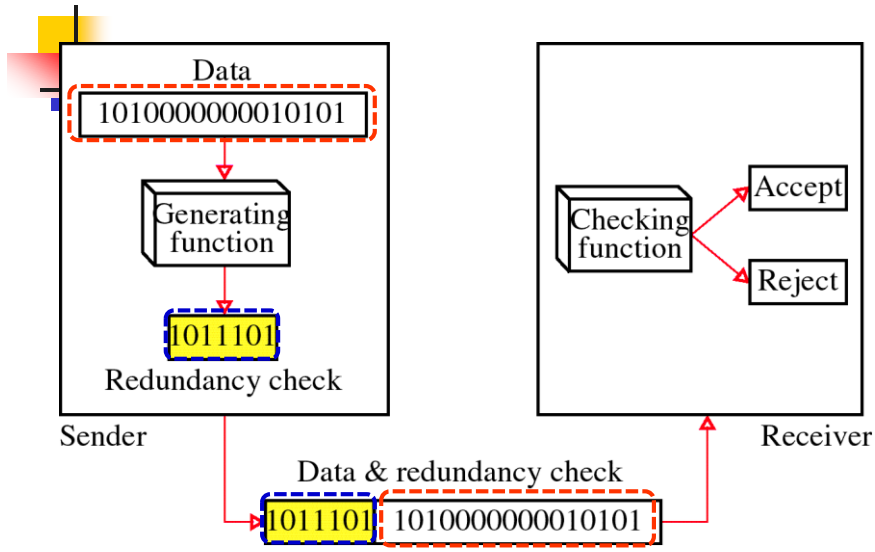
Error Control Process (3)

- All transmission media have potential for introduction of errors
- Error control process has two components
- Error detection
  - redundancy in data so that error can be detected
- Error correction
  - once error detected, frames corrected or retransmitted





Redundancy



17

Error Detection Codes

Linear, systematic block codes

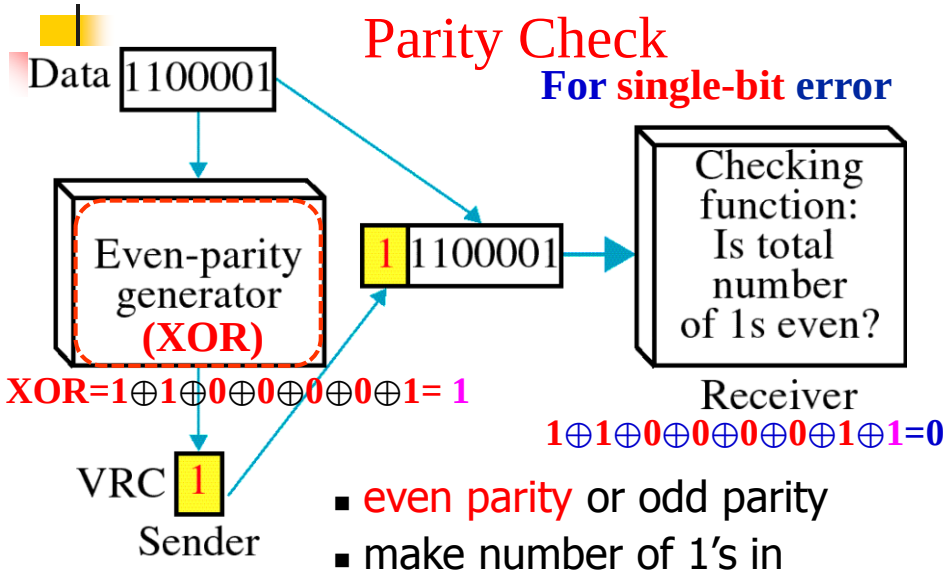
1. Parity.
2. Checksums.
3. Cyclic Redundancy Checks (CRCs).
4. Hamming codes.
5. Binary convolutional codes.
6. Reed-Solomon codes.
7. Low-Density Parity Check codes.

18

Error Detection Codes (1)

Parity Check

For single-bit error



- even parity or odd parity
- make number of 1's in character even or odd

WCB/McGraw-Hill

21

XOR GATE

(Exclusive-OR gate)

BOOLEAN EXPRESSION

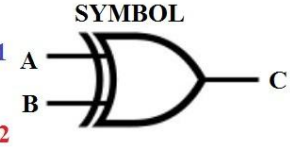
$$A \cdot \overline{B} + \overline{A} \cdot B$$

$$(A + B) \cdot (\overline{A} + \overline{B})$$

C = A⊕B

Input1 Input2 Output

SYMBOL



INPUT		OUTPUT
A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

Sender:

Even parity bit of 1100001 = 1

(C= 1⊕1⊕0⊕0⊕0⊕0⊕1 = 1)

Receiver:

C'= 1⊕1⊕0⊕0⊕0⊕0⊕1⊕1=0

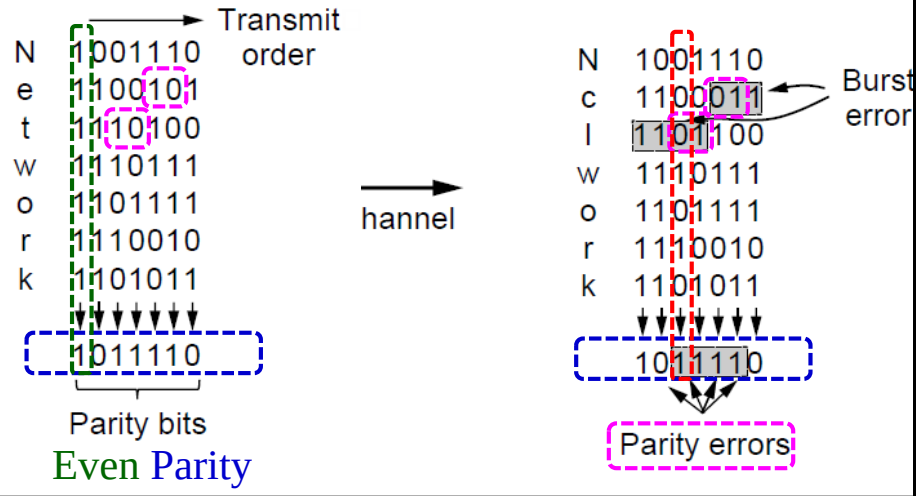
Note: Even parity = XOR = Modulo 2 sum

https://projectiot123.com/2019/05/26/introduction-to-xor-gate/

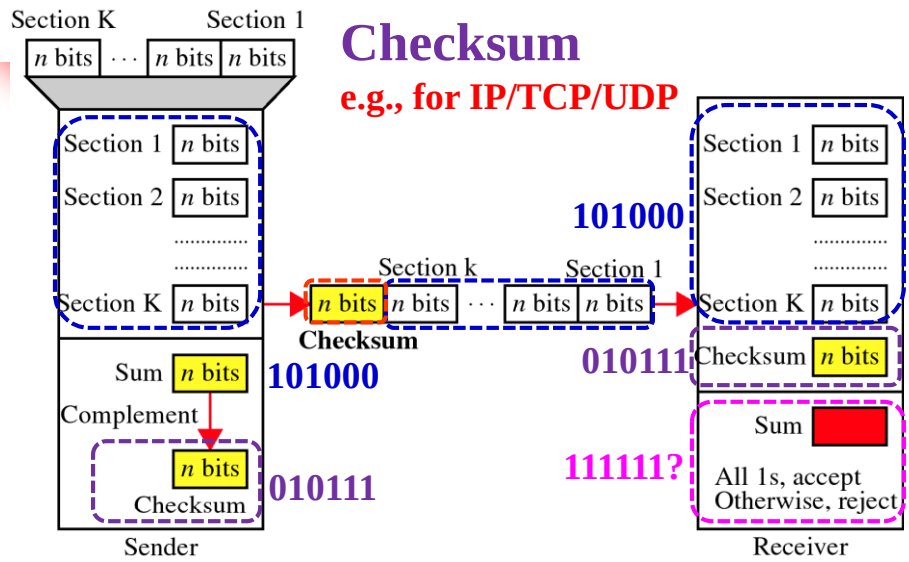
22

Error-Detecting Codes (1)

Interleaving of parity bits to detect a **burst error**.



Error-Detecting Codes (2)



WCB/McGraw-Hill

The checksum is the 16 bit one's complement of the one's complement sum of all 16 bit words.

24

- Suppose that a message 1001 1100 1010 0011 is transmitted using Internet Checksum(4-bit word).

The 1's complement sum of words:

Receiver checking:

```
  1001
  1100
  1010
  0011
+ 1011 (checksum)
------
  1111 = 0100 + 1011
```

Sender

```
  1001
  1100
  1010
+ 0011
------
  0010
     1
     1
------
  0100
```

Then 1's complement of 0100 is 1011.

Thus the Internet checksum is 1011.

```
  1001
+ 1100
------
  0101
     1
------
  0110
+ 1010
------
  0000
     1
------
  0001
+ 0011
------
  0100
     0
------
  0100
```

25

Error Detection Codes (3)

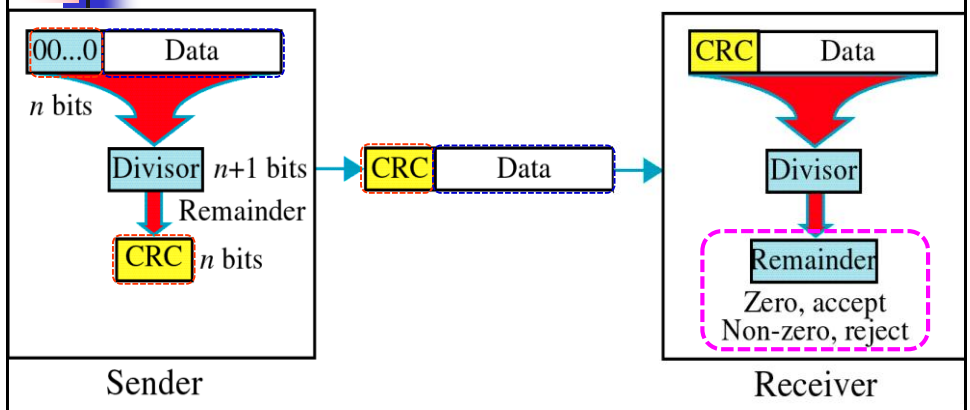
- Cyclic Redundancy Check (CRC)
  - treat frame as one long binary number
  - attach remainder to frame
  - 16-bit (32-bit) CRC

26

Error-Detecting Codes (3)

Error-Detecting Codes -- **CRC**

CRC (Cyclic Redundancy Check)



E.g., FCS(Frame Check Sequence)
in Ethernet/ WiFi frame format

Error-Detecting Codes (3)

Frame: 1 1 0 1 0 1 1 1 1 1 = $x^9 + x^8 + x^6 + x^4 + x^3 + x^2 + x^1 + 1$ (Dividend)
Generator: 1 0 0 1 1 = $1x^4 + 0x^3 + 0x^2 + 1x^1 + 1x^0 = x^4 + x + 1$ (Divisor)

```
                1 1 0 0 0 0 1 1 1 0   ← Quotient (thrown away)
          ┌─────────────────────────
1 0 0 1 1 │ 1 1 0 1 0 1 1 1 1 1 0 0 0 0   ← Frame with four zeros appended
          │ 1 0 0 1 1
          ├─────────
            1 0 0 1 1
          XOR 1 0 0 1 1
          ─────────
              0 0 0 0 1
              0 0 0 0 0
              ─────────
                0 0 0 1 1
                0 0 0 0 0
                ─────────
                  0 0 1 1 1
                  0 0 0 0 0
                  ─────────
                    0 1 1 1 1
                    0 0 0 0 0
                    ─────────
                      1 1 1 1 0
                    XOR 1 0 0 1 1
                    ─────────
                        1 1 0 1 0
                        1 0 0 1 1
                        ─────────
                          1 0 0 1 0
                          1 0 0 1 1
                          ─────────
                            0 0 0 1 0
                            0 0 0 0 0
                            ─────────
                              0 0 1 0   ← Remainder (n = 4 for n-bit CRC)
```

Divisor needs $n + 1 = 5$ bits

CRC uses Modulo 2 division
Subtraction rules:
1-1=0，0-0=0，1-0=1，0-1=1
(subtraction with no borrow)
➔ XOR operation

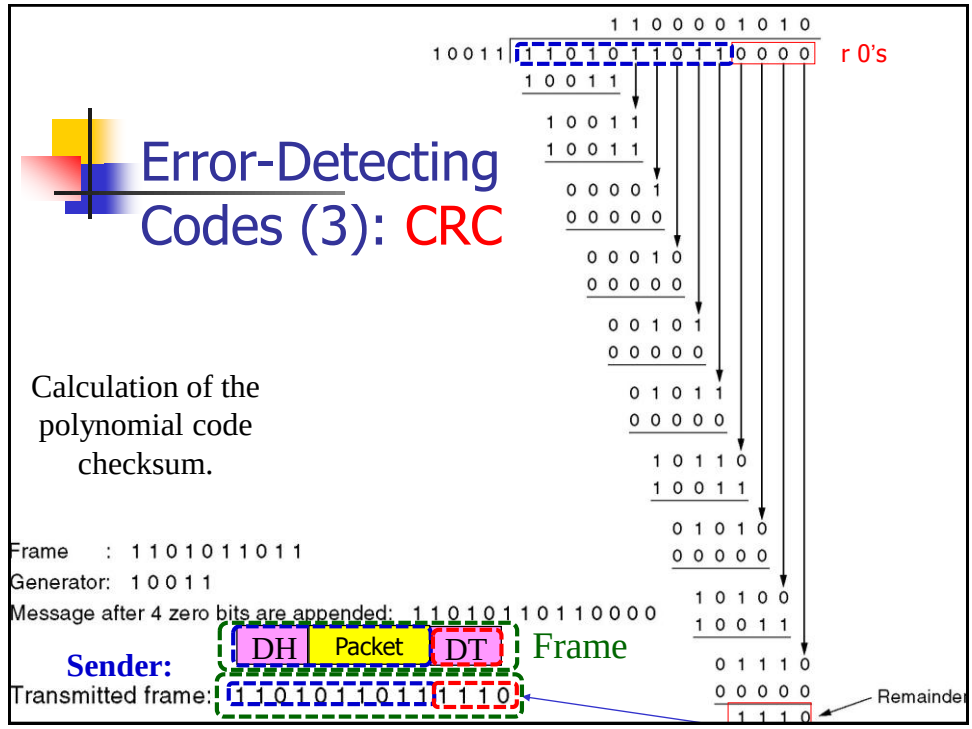
A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

Sender:
Transmitted frame: 1 1 0 1 0 1 1 1 1 1 0 0 1 0 ← Frame with four zeros appended minus remainder

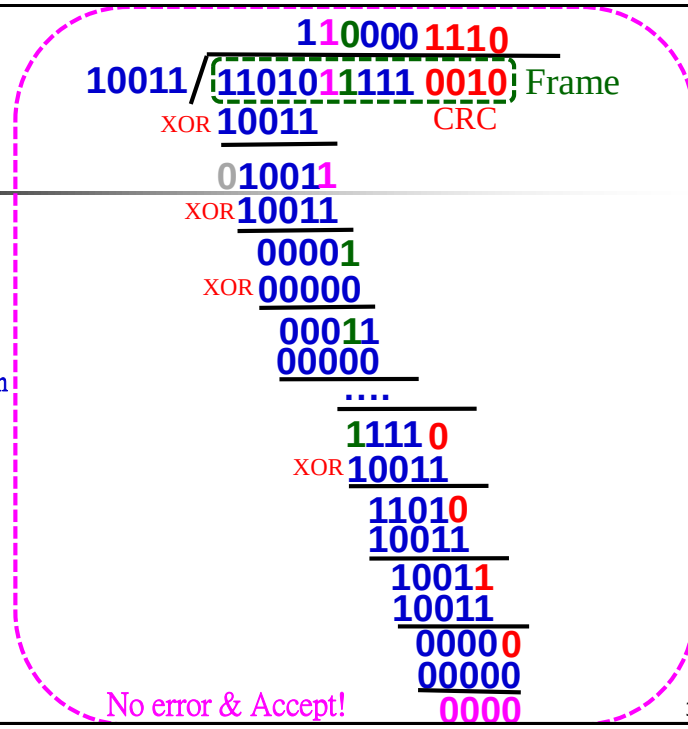
Example calculation of the CRC (CRC-4)

Error-Detecting Codes (3): CRC

Calculation of the polynomial code checksum.



Receiver CRC checking:



Modulo 2 Division (subtraction with no borrow) by XOR

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

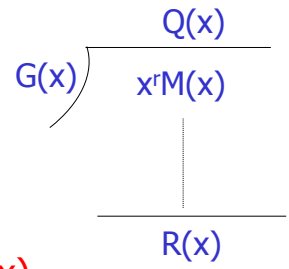
30

Error-Detecting Codes (3)

CRC (Cyclic Redundancy Check)

CRC Code = Polynomial Code
110001 = $x^5 + x^4 + x^0$
Original Message (dividend): M(x)
Generator (divisor): G(x)
Degree of G(x): r
Quotient: Q(x)
Remainder (CRC): R(x)
Transmitted Message: T(x)

Modulo 2 division
(add/sub = XOR)



$T(x) = x^r M(x) - R(x) = M(x)\&R(x)$

$T(x)/G(x) = 0$

If $T'(x)/G(x) = 0$ then Correct, otherwise Error

31

Standard Polynomials

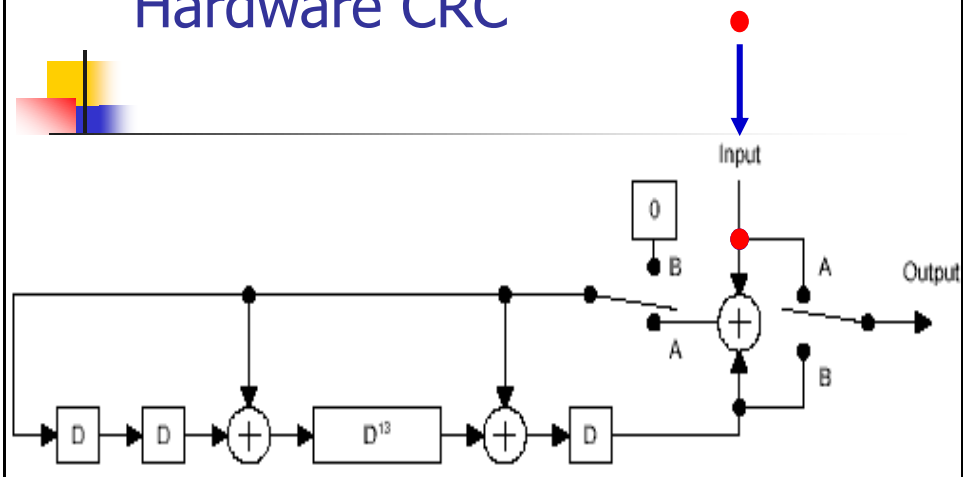
CRC-12	CRC-16	CRC-ITU
$x^{12}+x^{11}+x^3+x+1$	$x^{16}+x^{15}+x^2+1$	$x^{16}+x^{12}+x^5+1$
	(used in HDLC)	

CRC-32

$$x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$$

(used in Ethernet)

Hardware CRC



Source: http://www.erg.abdn.ac.uk/users/gorry/course/dl-pages/crc.html

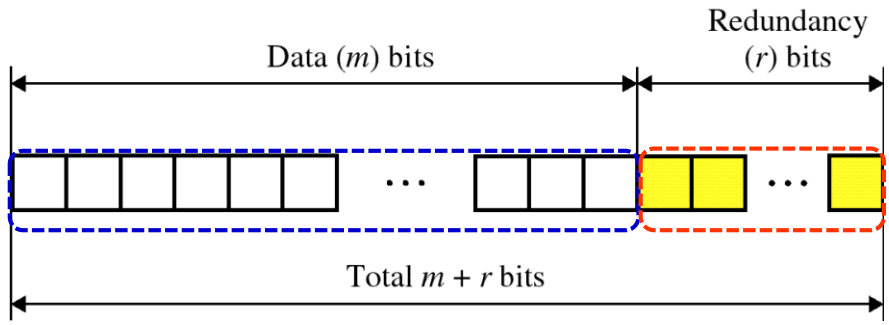
34

Error Correction (1)

- Two types of errors
  - Lost frame - never arrives
  - Damaged frame - error in bits
- Automatic Repeat reQuest (ARQ)
  - Error detection
  - Positive acknowledgment (ACK) if received
  - Retransmission after time-out if not ACK
  - Negative ACK (NACK)and retransmission if error
- Error-Correcting Codes

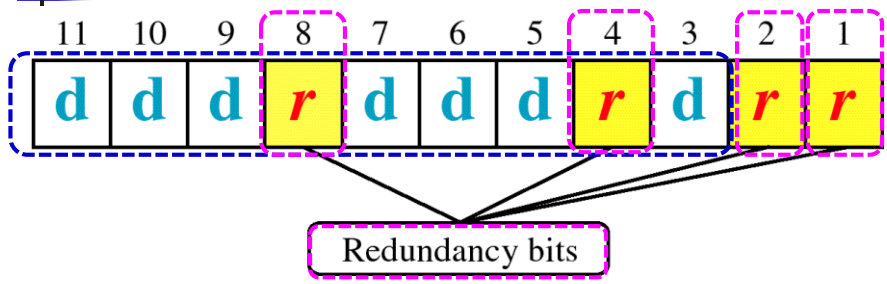
35

Error Correction (2)



Error Correction (3): **Hamming Code**

Hamming code with 7 data bits and 4 check bits

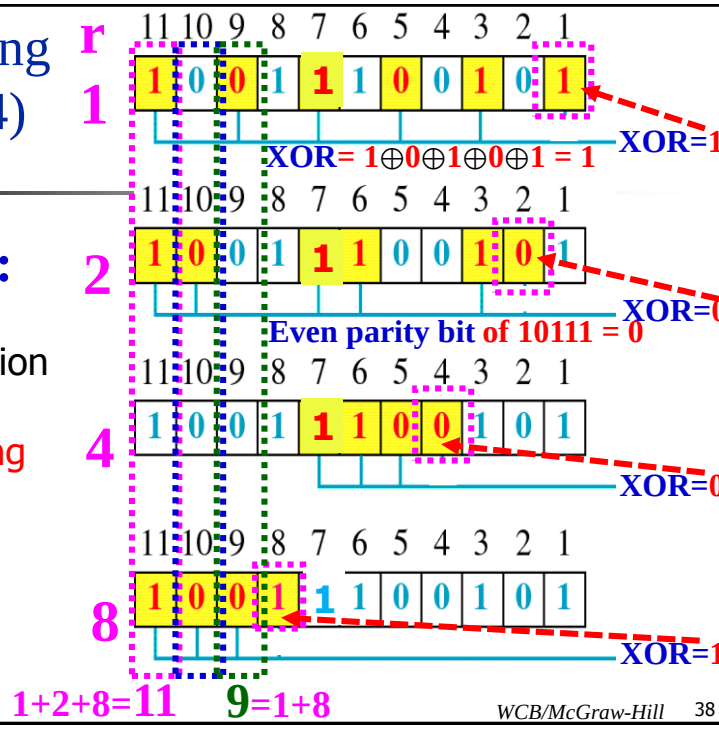


The bits that are powers of 2 (1, 2, 4, 8, 16, etc.) are check bits.
The rest (3, 5, 6, 7, 9, 11, etc.) are filled up with the *m* data bits.

Hamming Code (4)

Sender:

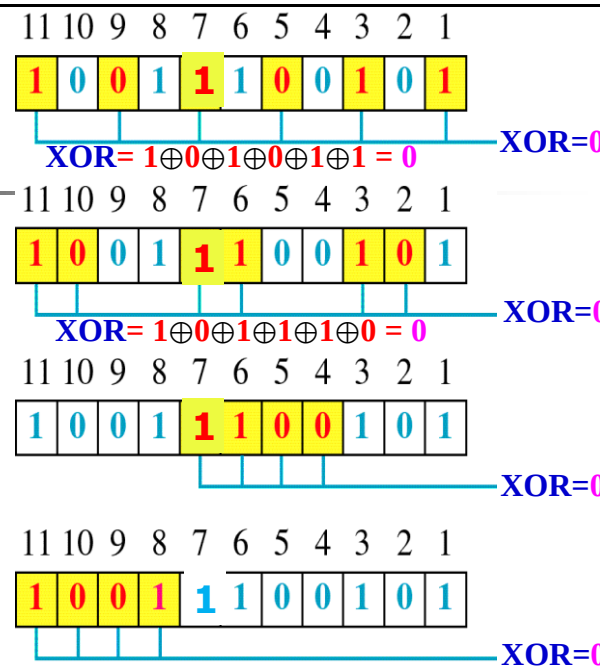
Generation of Hamming Code (**Even Parity** Check)



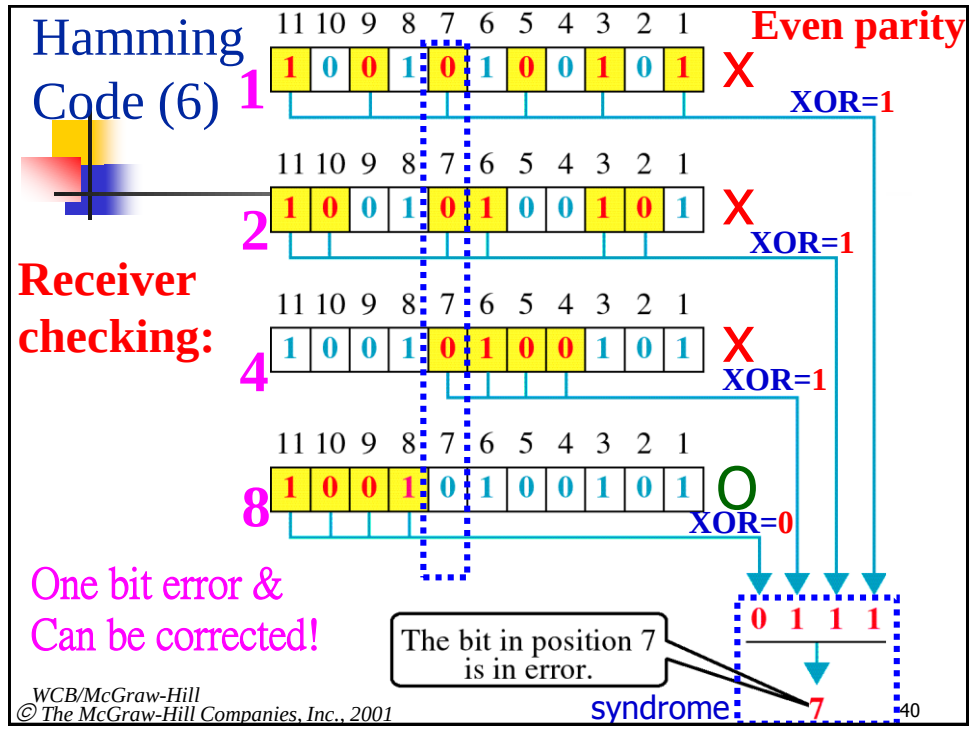
Hamming Code (5)

Receiver checking:

(**Even Parity** Check)



No error & Accept!



Error-Correcting Codes (7)

Use of a Hamming code to correct 1-bit errors.

Starting with bit 1 at the left end, bit 2 to its immediate right, and so on.

Char.	ASCII	Check bits (1,2,4,8)
H	1001000	00110010000
a	1100001	10111001001
m	1101101	11101010101
m	1101101	11101010101
i	1101001	01101011001
n	1101110	01101010110
g	1100111	01111001111
	0100000	10011000000
c	1100011	11111000011
o	1101111	10101011111
d	1100100	11111001100
e	1100101	00111000101

Order of bit transmission

Check bits may start with bit 1 at the left end

與上頁左右顛倒(也是可以)

41

Example of an (11, 7) Hamming code correcting a single-bit error

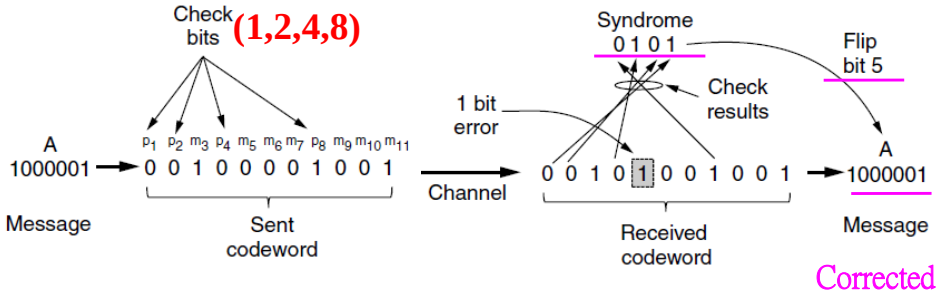


Figure 3-6. Example of an (11, 7) Hamming code correcting a single-bit error.

42

Hamming code for 8-bit data

Codeword:	1	2	3	4	5	6	7	8	9	10	11	12
			D0		D1	D2	D3		D4	D5	D6	D7
	H0	H1		H2				H3				

Hamming code:

$H0 = D0 \oplus D1 \oplus D3 \oplus D4 \oplus D6$

$H1 = D0 \oplus D2 \oplus D3 \oplus D5 \oplus D6$

$H2 = D1 \oplus D2 \oplus D3 \oplus D7$

$H3 = D4 \oplus D5 \oplus D6 \oplus D7$

Note: $\oplus$ = XOR

43

Hamming code (example)

An 8-bit byte with binary value 10101111

			D0		D1	D2	D3		D4	D5	D6	D7
	1	2	3	4	5	6	7	8	9	10	11	12
Codeword:			1		0	1	0		1	1	1	1
	H0 1	H1 0		H2 0				H3 0				

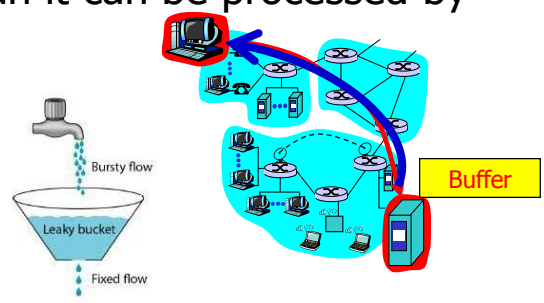
H0 = 1 ⊕ 0 ⊕ 0 ⊕ 1 ⊕ 1 = 1
H1 = 1 ⊕ 1 ⊕ 0 ⊕ 1 ⊕ 1 = 0
H2 = 0 ⊕ 1 ⊕ 0 ⊕ 1 = 0
H3 = 1 ⊕ 1 ⊕ 1 ⊕ 1 = 0

The encoded value (codeword) is 101001001111.

44

Flow Control

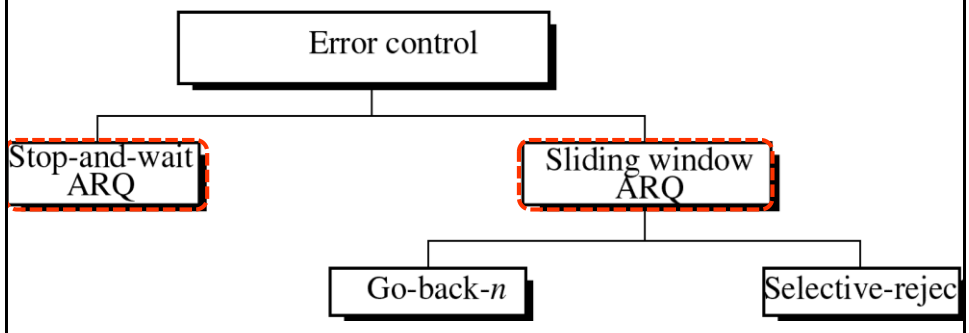
- Necessary when data is being sent faster than it can be processed by receiver



- Usually a buffer is filled, and transfer is stopped until buffer is emptied

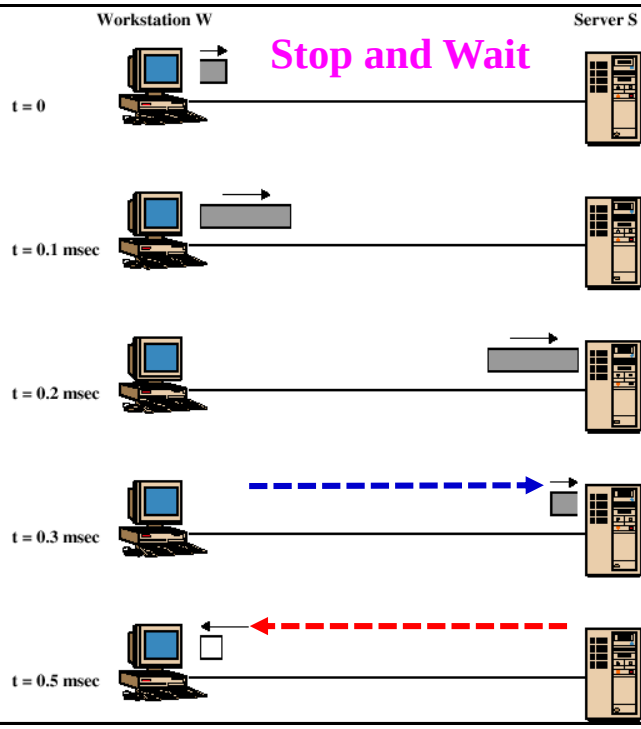
45

Flow Control with Error Control



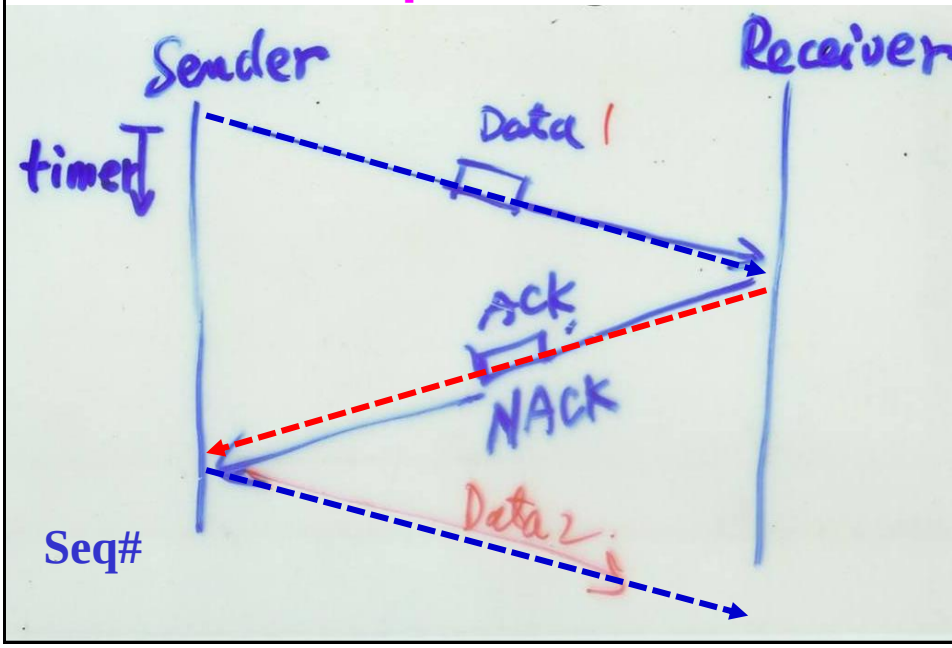
46

Stop and Wait

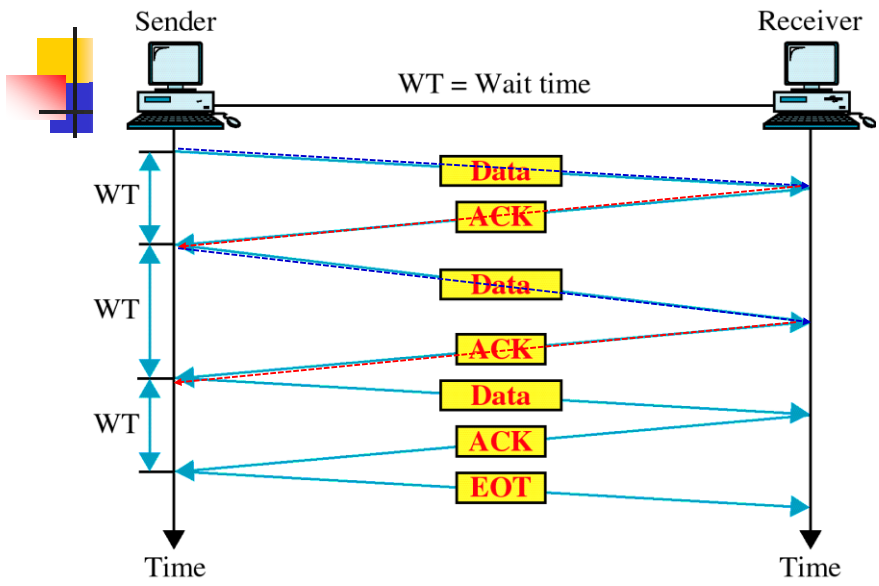


49

Stop-And-Wait

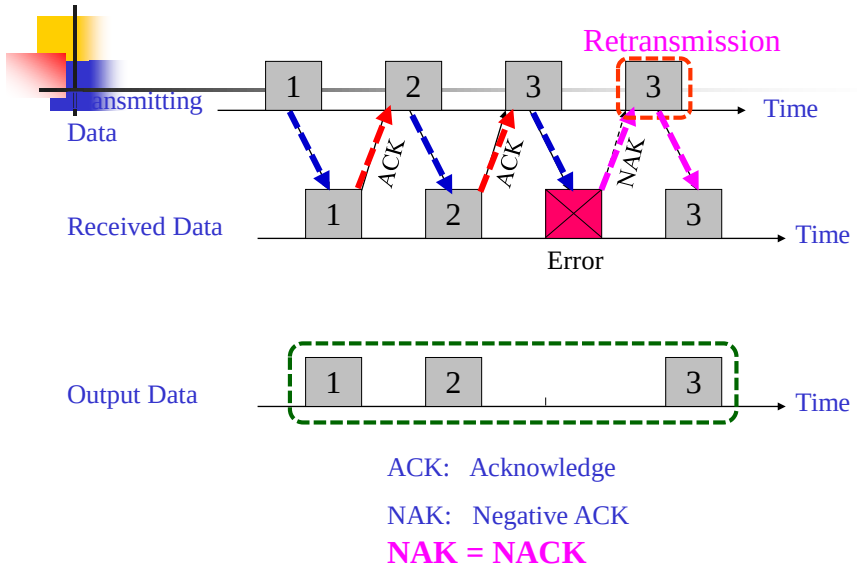


Stop and Wait ARQ



51

Stop-And-Wait ARQ



ACK: Acknowledge

NAK: Negative ACK

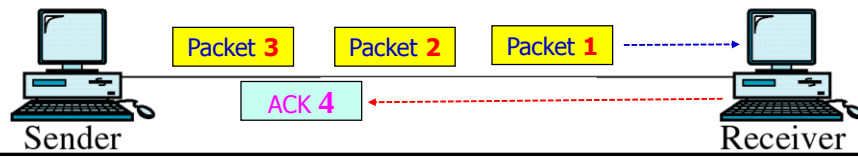
NAK = NACK

Stop-and-Wait Flow Control

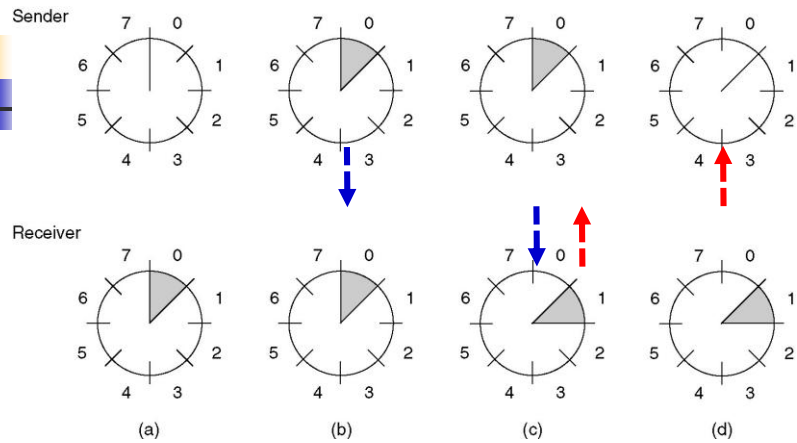
- Simplest form of Frame-Oriented Control
- Source may not send new frame until receiver acknowledges previous one
- Very inefficient, especially when a single message is broken into many small frames
  - buffer size of receiver is limited
  - if error, detected sooner and less data need be retransmitted

Sliding-Window Flow Control

- Allows multiple frames to be in transit
- Receiver sends acknowledgement with sequence number of next frame
- Sender maintains list of sequence numbers it can send, receiver maintains list of sequence numbers it can receive



Sliding Window Protocols (2)



A sliding window of size 1, with a **3-bit sequence number**.

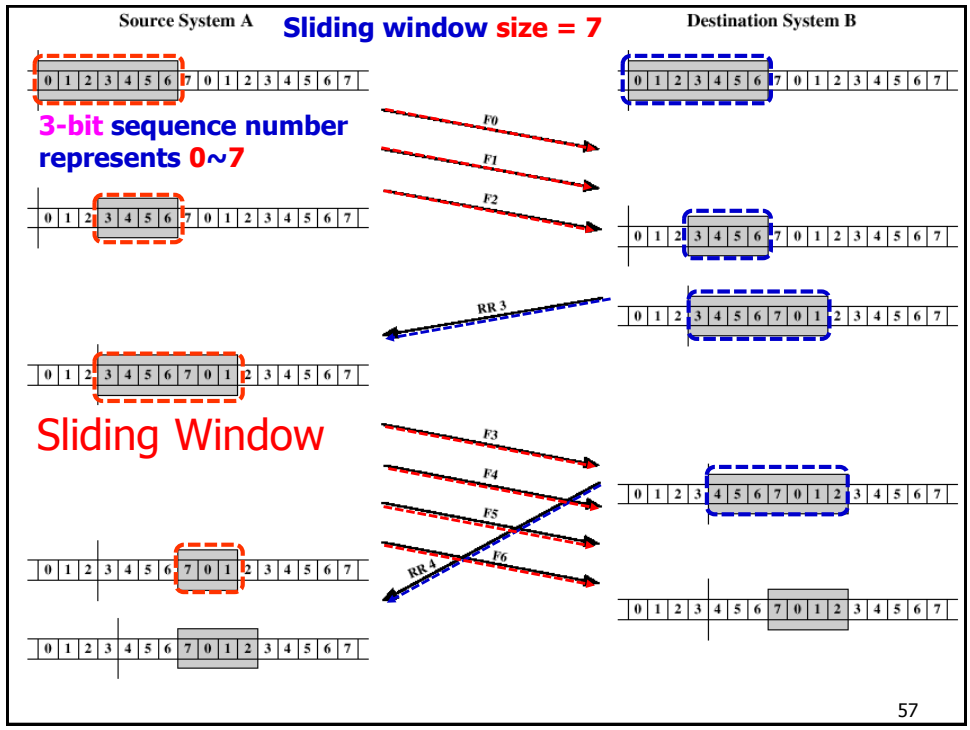
(a) Initially. **(size = 1 相當於Stop-And-Wait ARQ)**

(b) After the first frame has been sent.

(c) After the first frame has been received.

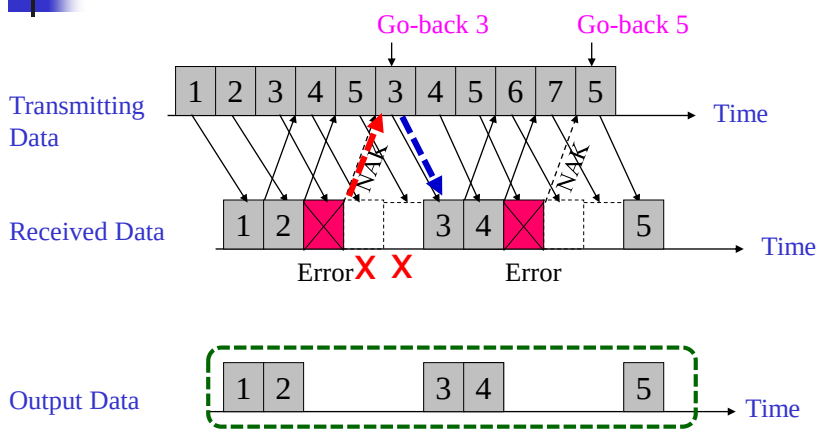
(d) After the first acknowledgement has been received.

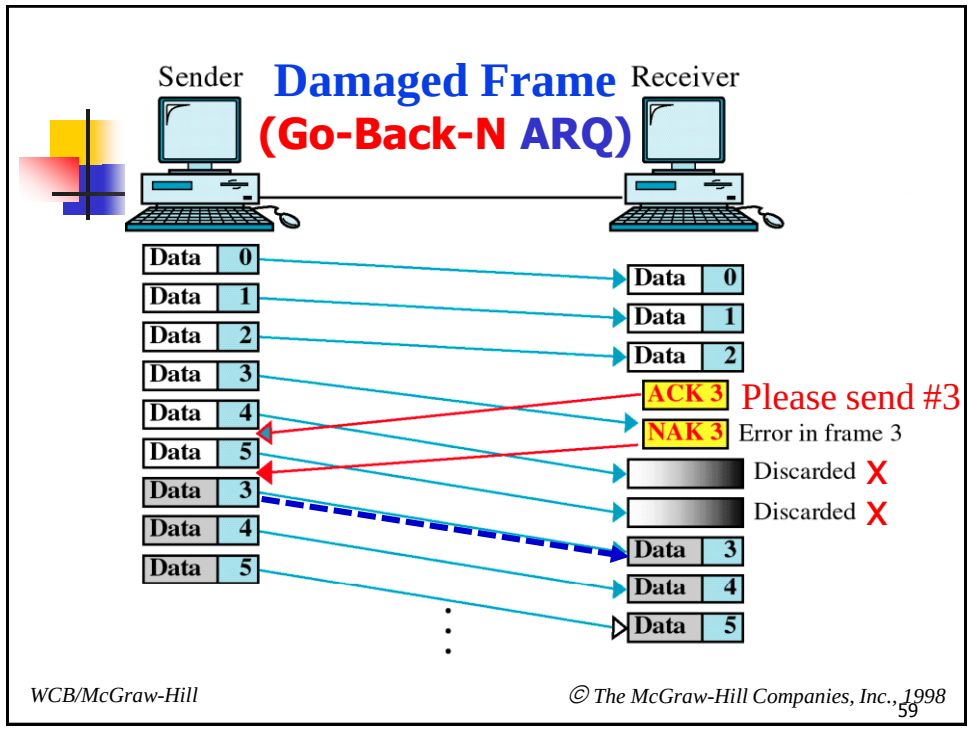
55



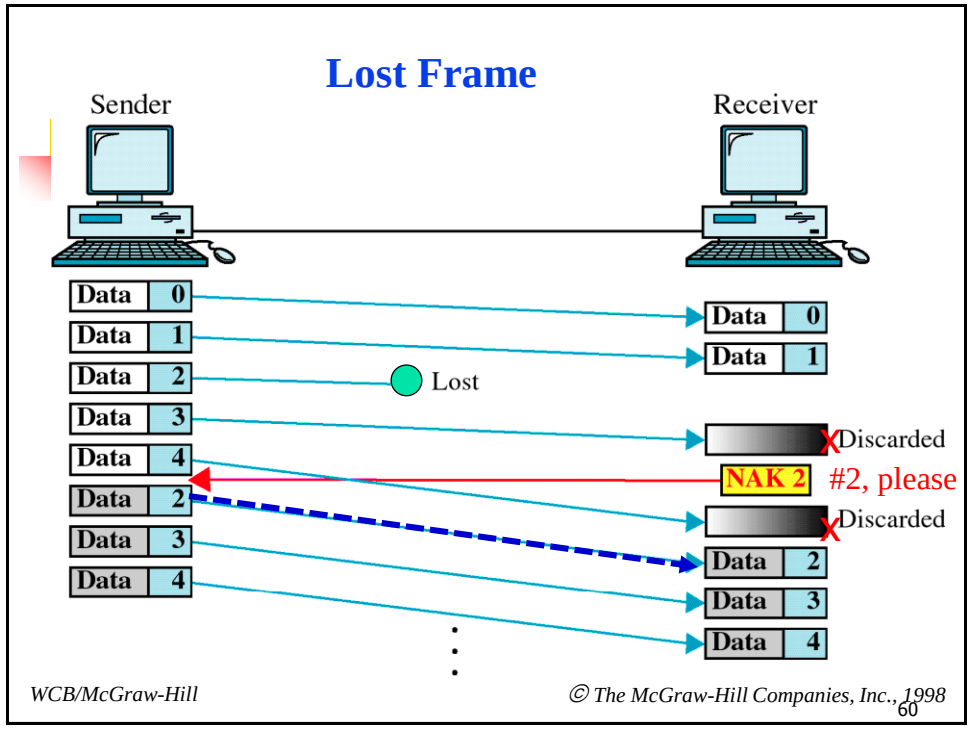
Sliding-Window Flow Control (1)

Go-Back-N ARQ (GBN ARQ)



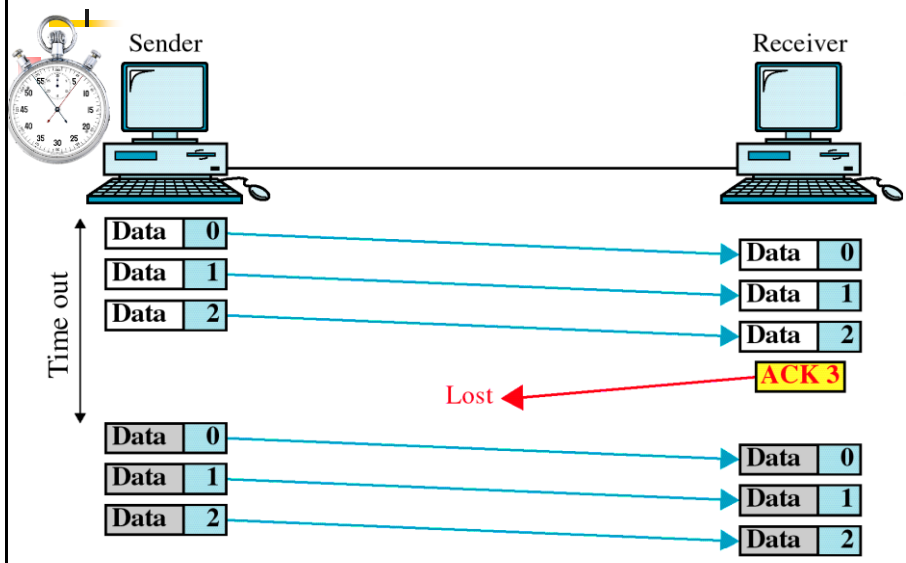


59



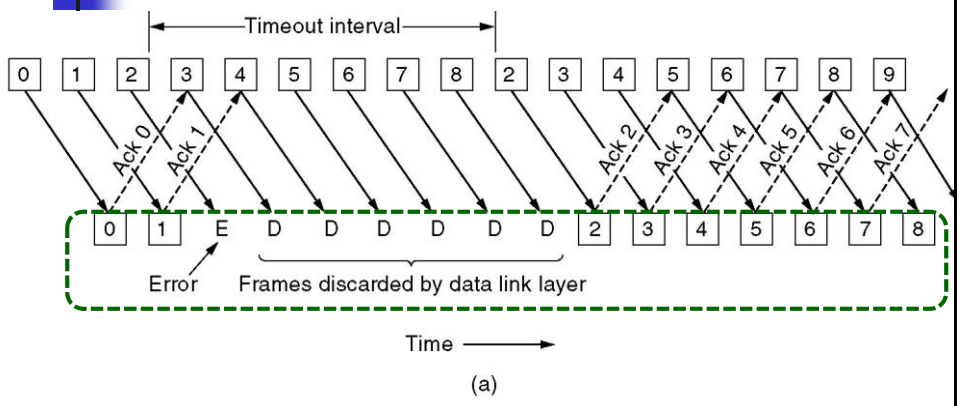
60

Lost ACK



61

A Protocol Using Go Back N

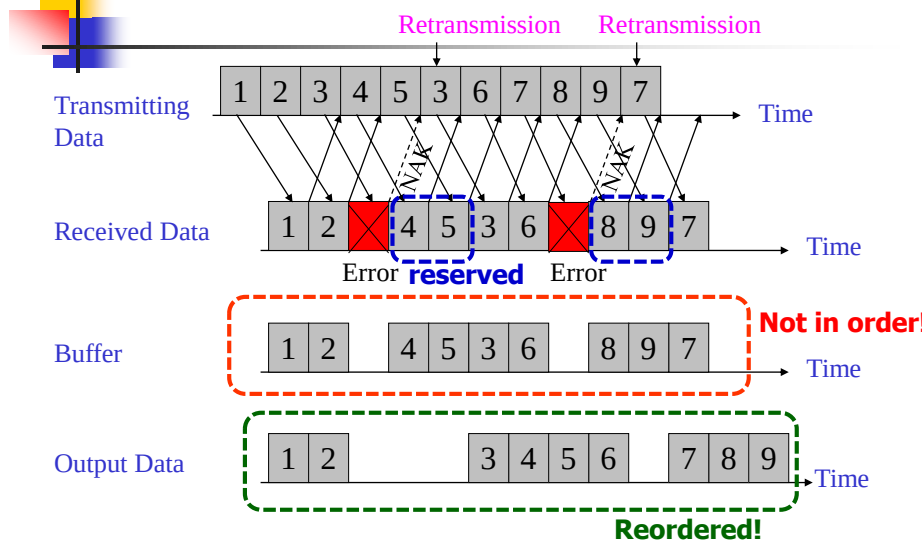


Pipelining and error recovery.

62

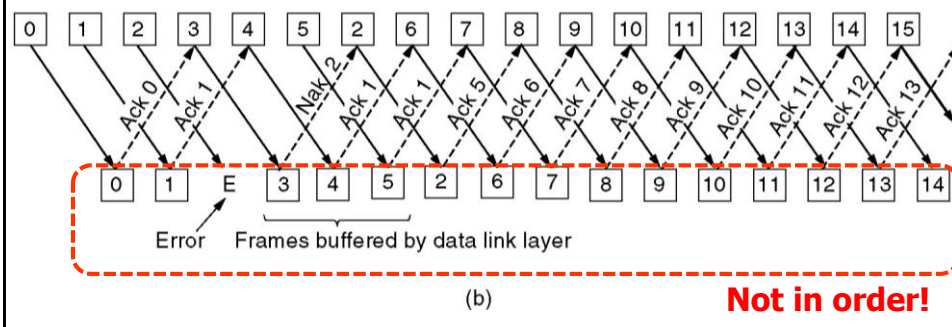
Sliding-Window Flow Control (2)

Selective-Repeat ARQ (SR ARQ)



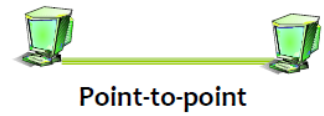
64

A Protocol Using Selective Repeat



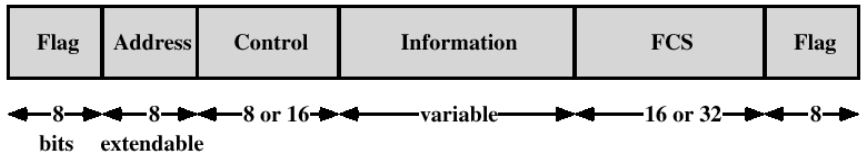
65

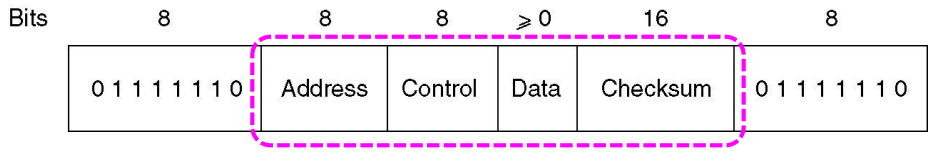
High-Level Data Link Control (HDLC)



Frame format for bit-oriented protocols.

(Bit stuffing in hardware)

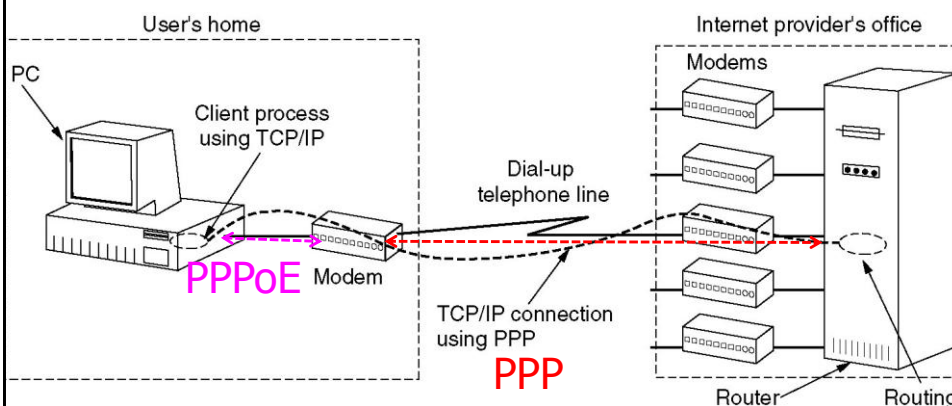




73

The Data Link Layer in the Internet: PPP

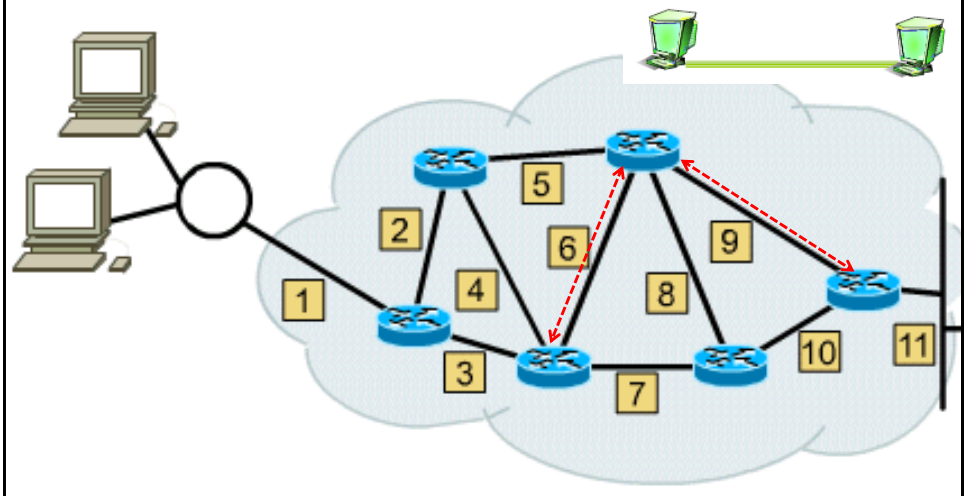
A home personal computer acting as an internet host.



Byte stuffing in software/ for modem

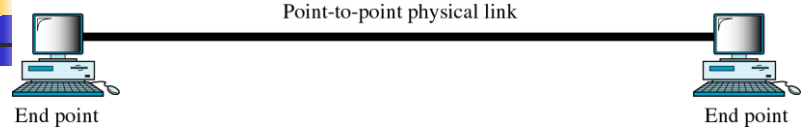
79

PPP (Point to Point Protocol)

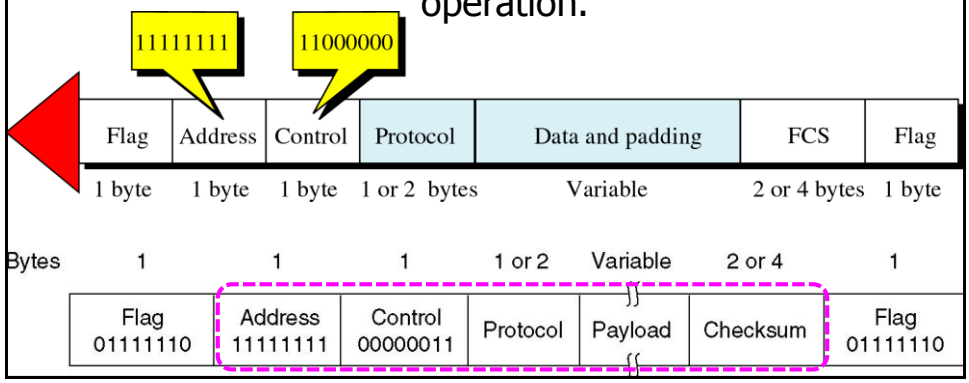


PPP – Point to Point Protocol

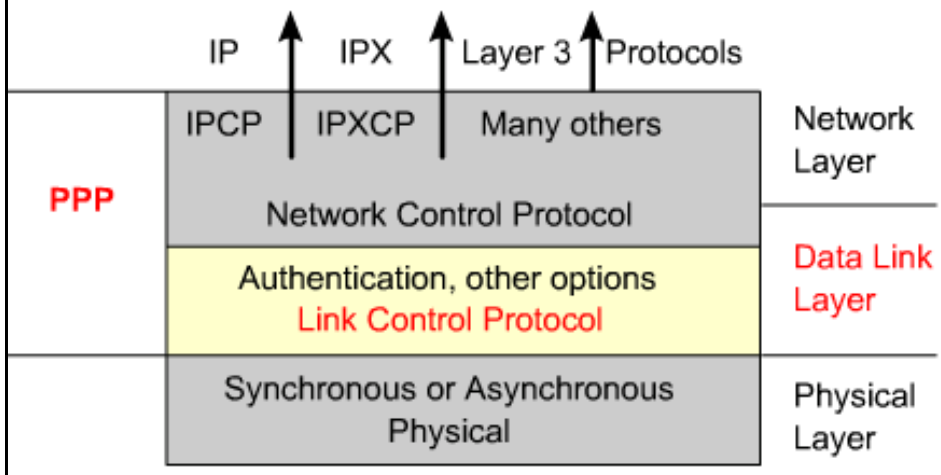
(Byte stuffing in software)



The PPP full frame format for unnumbered mode operation.



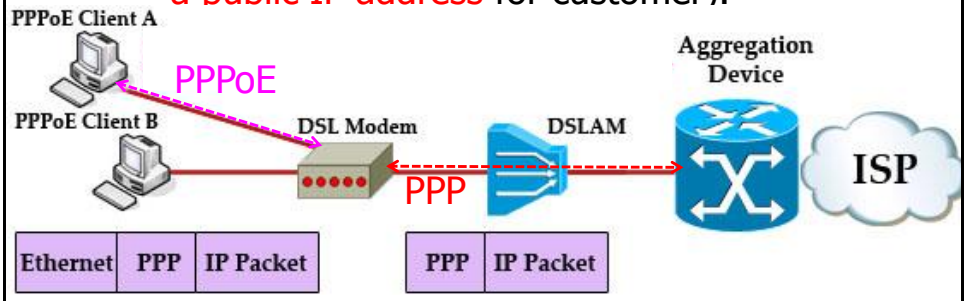
PPP – Point to Point Protocol (2)



82

PPP over Ethernet (PPPoE)

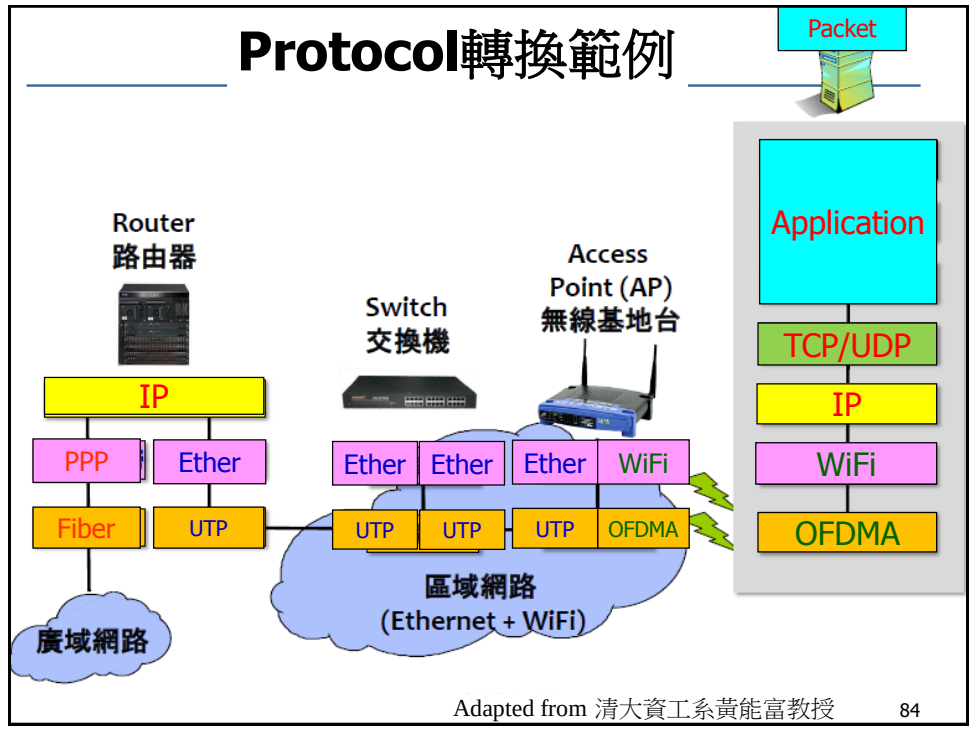
- Internet Service Provider (ISP) still like PPP because of authentication (PPP supports CHAP), accounting (checking customer's bill), link management (ISP can use PPP to assign a public IP address for customer).

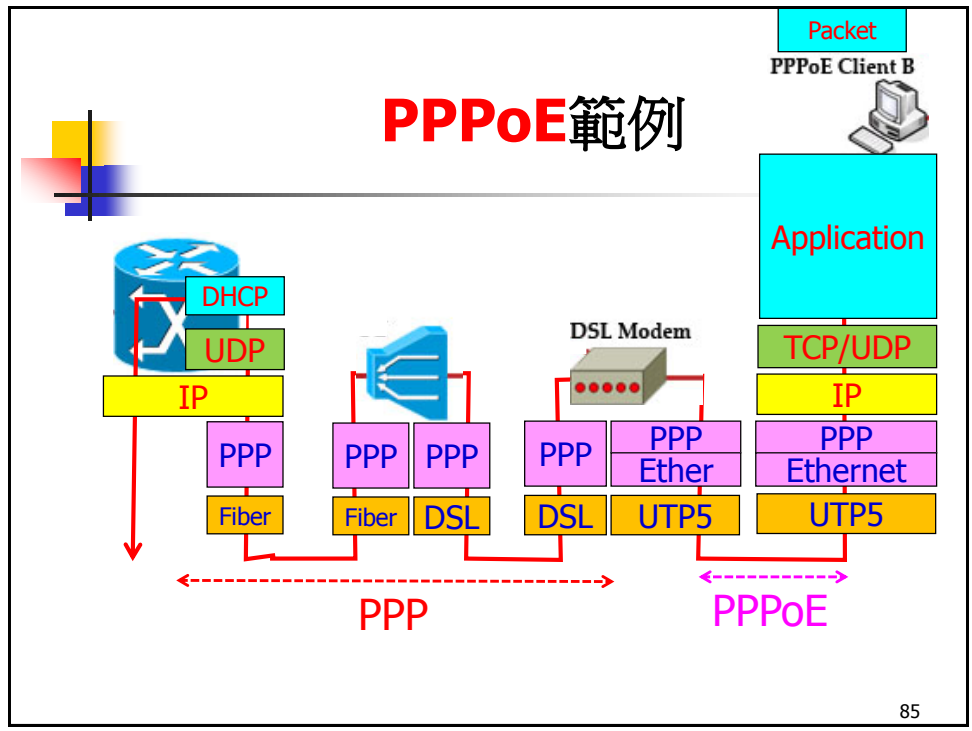


DSLAM: Digital Subscriber Line Access Multiplexer

Source: http://www.digitaltut.com/ppp-over-ethernet-pppoe-tutorial

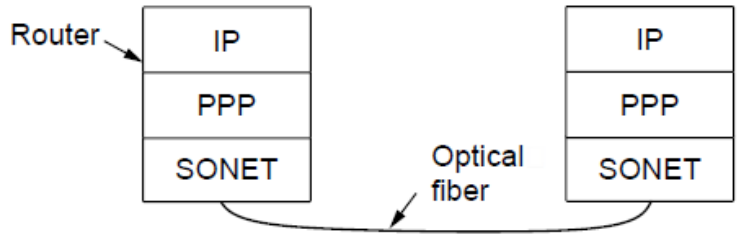
83

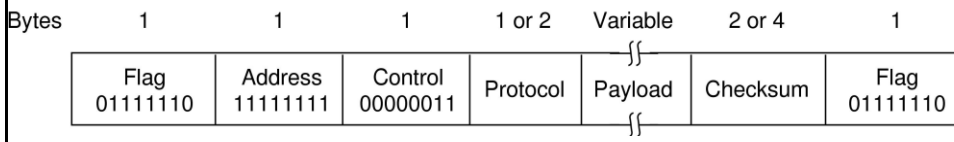

Protocol轉換範例
Packet
Application
TCP/UDP
IP
WiFi
OFDMA
Router
路由器
Switch
交換機
Access
Point (AP)
無線基地台
IP
PPP
Ether
Fiber
UTP
Ether
Ether
Ether
WiFi
UTP
UTP
UTP
OFDMA
區域網路
(Ethernet + WiFi)
廣域網路
Adapted from 清大資工系黃能富教授
84


PPPoE範例
Packet
PPPoE Client B
Application
TCP/UDP
IP
PPP
Ethernet
UTP5
DHCP
UDP
IP
PPP
Fiber
DSL Modem
PPP
Fiber
PPP
DSL
PPP
DSL
PPP
Ether
UTP5
PPP
PPPoE
85

Packet over SONET

(a) A protocol stack. (b) Frame relationships





The PPP full frame format for unnumbered mode operation

86

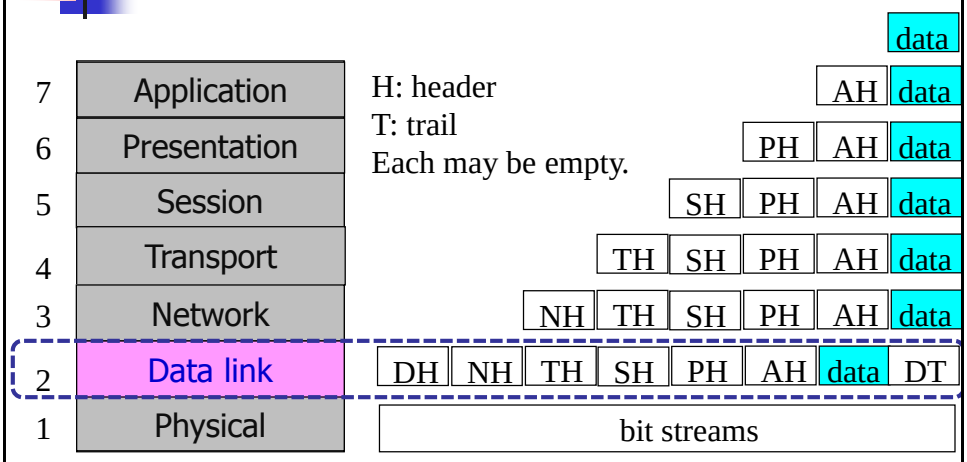
各個電腦自行利用PPPoE撥接ADSL上網

PPP over WLAN
PPP over WiFi

Source: http://www.im.isu.edu.tw/faculty/jbsuen/www/net/wireless.ppt

87

The Data Link Layer



OSI Reference Model

92

End of Chapter 3

Questions?

Thank you!