

Chapter 11 Data Link Control and Protocols

- Flow control: เป็นเงื่อนไข (เทคนิค, โปรโตคอล) เพื่อควบคุมปริมาณข้อมูลที่สามารถแลกเปลี่ยนระหว่างตัวส่ง กับตัวรับ
- Error control: การจัดการเมื่อเกิด error ขึ้น เช่น ให้ส่ง frame ใหม่; error ที่เกิดขึ้นเช่น frame error, frame หาย, delay, duplicate frame
- Acknowledgment คือการตอบสนองของตัวรับเมื่อได้รับข้อมูล แบ่งเป็น
 - positive acknowledgment (ACK)
 - negative acknowledgment (NAK) แจ้งว่าข้อมูลที่ได้รับมี error เกิดขึ้น

⇒ Protocol

1. Simplest

- ตัวส่ง ส่งอย่างเดียว ตัวรับก็รับอย่างเดียว ไม่มีการส่ง ACK กลับไป
- ไม่มี Error control
- unidirectional transmission
- โครงสร้าง frame

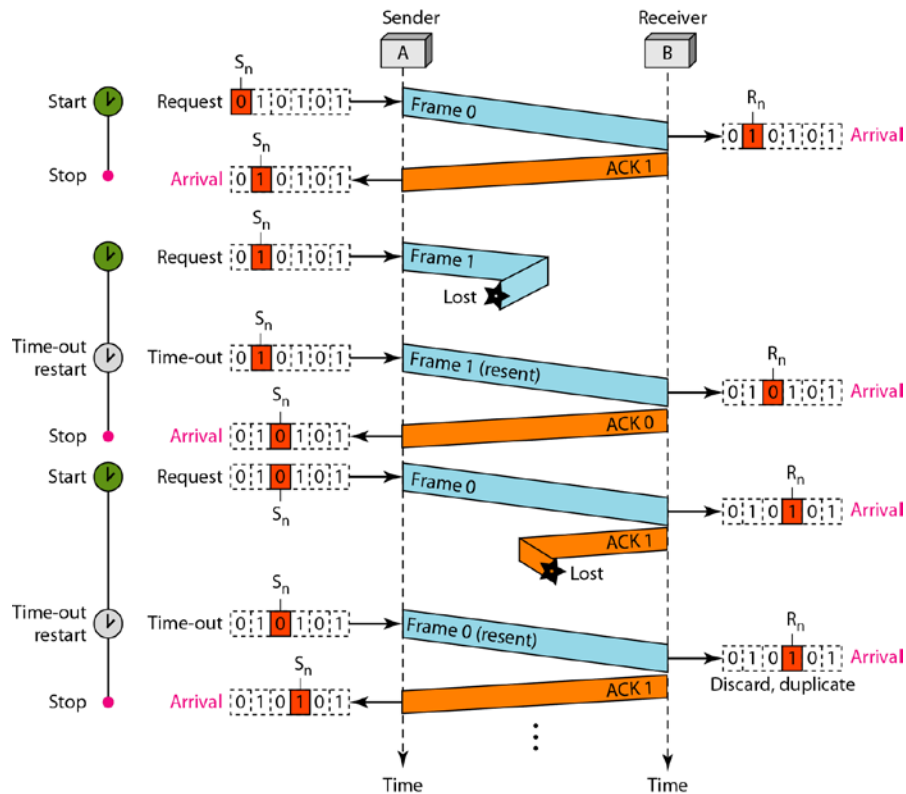
Flag <start frame>	Header <address>	Data	Trailer <error detection>	Flag <end frame>
-----------------------	---------------------	------	------------------------------	---------------------

2. Stop-and-Wait ARQ

- มีการจัดการเกี่ยวกับ error control แล้ว
- ตัวส่งไป frame ไป จะส่งต่อได้ต้อง
 - ได้รับ ACK frame# ถัดไป จากตัวรับ -> ส่ง frame ถัดไป
 - ได้รับ ACK/NAK frame# เดิม หรือ time out(ไม่ได้รับ ACK) -> ส่ง frame เดิม
- ตัวรับเมื่อได้รับ frame แล้ว
 - ตรงกับ frame # ที่รอรับ -> รับแล้ว ส่ง ACK frame# ถัดไป
 - ไม่ตรง -> ทิ้ง frame แล้วส่ง ACK หรือ NAK frame# ที่มันต้องการ
- ความกว้าง window ของตัวส่ง และตัวรับ ความกว้าง = 1
- ดังนั้นหมายเลข frame จึงมีได้แค่ 0 กับ 1
- การจัดการ error ใน case ต่างๆ

Frame error	ตัวรับ ส่ง ACK / NAK หรือรอจนตัวส่งเกิด time out และส่งมาใหม่
Lost frame	ตัวรับไม่ได้รับ frame -> ตัวส่งเกิด time out แล้วส่งใหม่
Lost ACK	ACK หาย -> ตัวส่งเกิด time out แล้วส่งใหม่

Duplicate frame	ตัวรับทั้ง frame ที่ได้รับซ้ำ และส่ง ACK frame# ที่ต้องการ
Delay ACK	ตัวส่ง time out ก่อนที่จะได้รับ ACK จึงส่ง frame เดิมซ้ำ (เป็น Duplicate frame) ตัวรับจึงจัดการทิ้งไป และส่ง ACK frame# ที่ต้องการ ตัวส่งได้รับ ACK (อันที่ delay) ก็ส่ง frame ใน ACK ไป



• Piggybacking

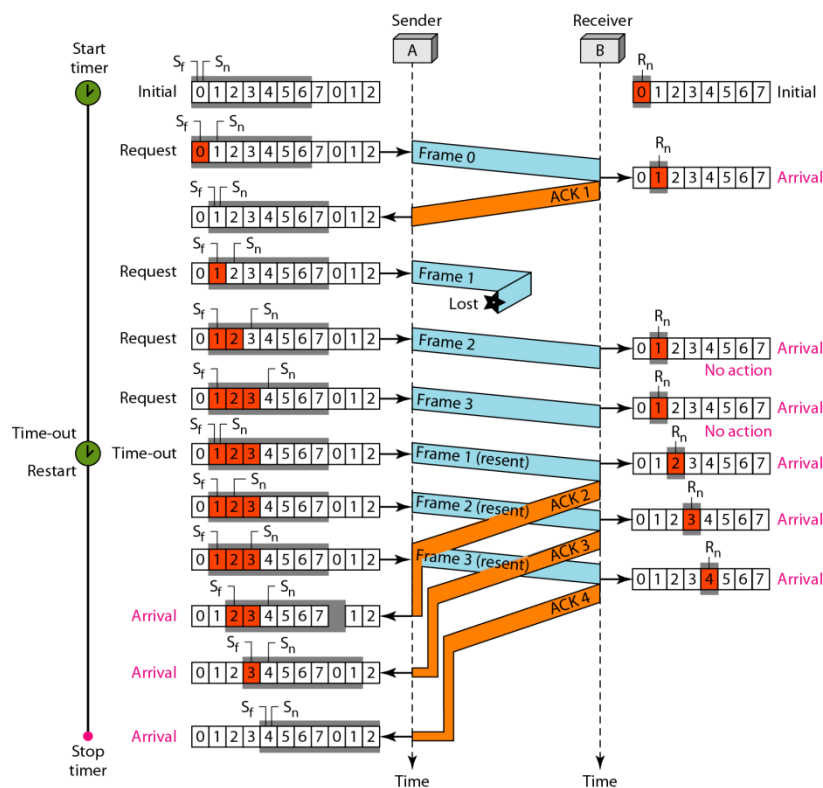
- สามารถส่ง Data frame + ACK ไปพร้อมกันได้

3. Go-Back-N ARQ

- ข้อดีกว่า stop-and-wait คือส่ง frame ที่เหลือใน window ได้เลย ไม่ต้องรอ ACK
- ความกว้าง window ของตัวส่ง $\leq 2^m - 1$ เช่นถ้าบอกว่าใช้หมายเลข frame 0 ถึง 15 (ใช้ 16 บิต คือ 2^4) ดังนั้น window size มากที่สุดคือ $16 - 1 = 15$
- ความกว้าง window ของตัวรับ = 1
- window ตัวส่งจะเลื่อนต่อเมื่อได้รับ ACK ถ้าไม่ได้ ACK ก็จะส่งต่อจนสุด แล้วเมื่อเกิด time out จะส่งไปใหม่โดยเริ่มตั้งแต่ frame แรกใน window
- frame# มีตั้งแต่เลข 0 ไปจนถึง window size
- การจัดการ error

Frame error	- ตัวรับส่ง NAK -> ตัวรับได้รับ NAK ก็เลื่อน window แล้วส่ง frame ใหม่
-------------	--

	(โดยเริ่มจากframeแรกใน window) - ตัวรับไม่ทำอะไร -> ตัวส่ง time out จึงส่งซ้ำตั้งแต่ frame แรกใน window
Lost frame, Out-of-order frame	- ตัวรับทั้ง frame ที่ผิด -> ตัวส่งก็ส่งframe ที่เหลือใน window ต่อไปจนหมด window หรือเกิด time out - เมื่อเกิด time out ตัวส่งจะส่งใหม่ตั้งแต่ frame แรกใน window
Lost ACK	- ตัวส่งก็ส่งframe ที่เหลือใน window ต่อไปจนหมด window - ถ้าเกิด time out ก็จะส่งใหม่ตั้งแต่ frame แรกใน window - ถ้ามี ACK เข้ามาก่อน time out ก็จะเลื่อน window



4. Selective Repeat ARQ

- ความกว้าง window ของตัวส่ง และตัวรับ $\leq 2^{m-1}$ เช่นถ้าบอกว่าใช้หมายเลข frame 0 ถึง 15 (ใช้ 16 ตัว คือ 2^4) ดังนั้น window size มากที่สุดคือ $2^3 = 8$
- มีข้อดีกว่า go-back-n คือเมื่อเกิด frame error ก็จะส่งใหม่เฉพาะ frame ที่ error ไม่ต้องส่งใหม่ทั้ง window

L2 Protocol Requirement

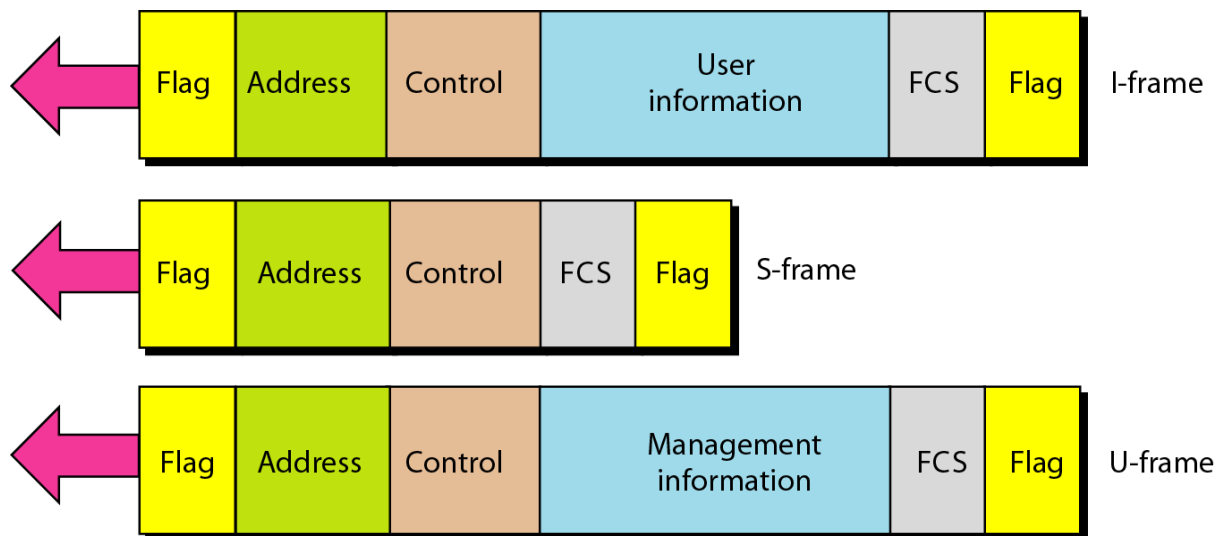
⇒ HDLC: High-level Data Link Control

- Configurations and Transfer Mode

1. Normal Response Mode (NRM) แบ่งเป็น Master/Slave (Primary/Secondary)
 - Slave จะส่งข้อมูลได้ก็ต่อเมื่อได้รับคำสั่ง Poll จาก Master
 - มีคำสั่ง Poll: ขอข้อมูลจาก Slave Select: ส่งข้อมูลให้ Slave
2. Asynchronous Response Mode (ARM) แบ่งเป็น Master/Slave เช่นเดียวกับ NRM
 - Slave ส่งข้อมูลได้โดยไม่ต้องรอ Master แต่การส่งข้อมูลจะต้องผ่าน Master (Slave คุยกันเองไม่ได้ ต้องผ่าน Master)
3. Asynchronous Balanced Mode (ABM) ไม่แบ่ง Master/Slave ทุกเครื่องเท่าเทียมกัน คุยกันได้โดยตรง

- Transmission mode: ได้ทั้ง point-to-point และ multipoint

- Frame: แบ่ง 3 กลุ่มคือ I-frame (Information frame), S-frame (Supervisory Frame) และ U-frame (Unnumbered frame)



1. Flag: เป็นจุดเริ่มต้น และสิ้นสุดของ frame มีค่าเป็น "01111110"
 - ถ้าในส่วน Data มี "01111110" จะทำการ bit stuffing คือเจอ 1 ติดกัน 5 ตัว จะยัด 0 เพื่อป้องกัน pattern ที่ซ้ำกับ flag

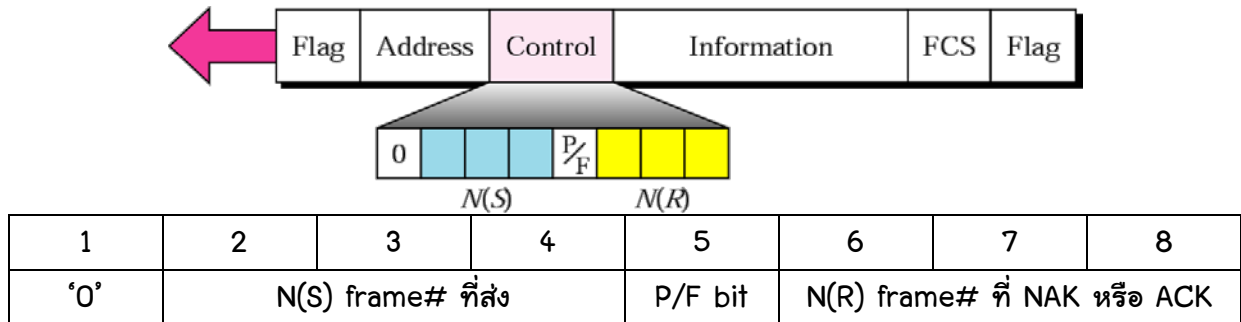
2. Address: มีได้ ≥ 1 byte โดยถ้าเป็น byte สุดท้ายต้องจบด้วย 1 เช่น

11001000	01100110	10010011
----------	----------	----------

3. FCS: เป็นตัว error detection ซึ่งใช้วิธี CRC

4. Control:

- I-frame

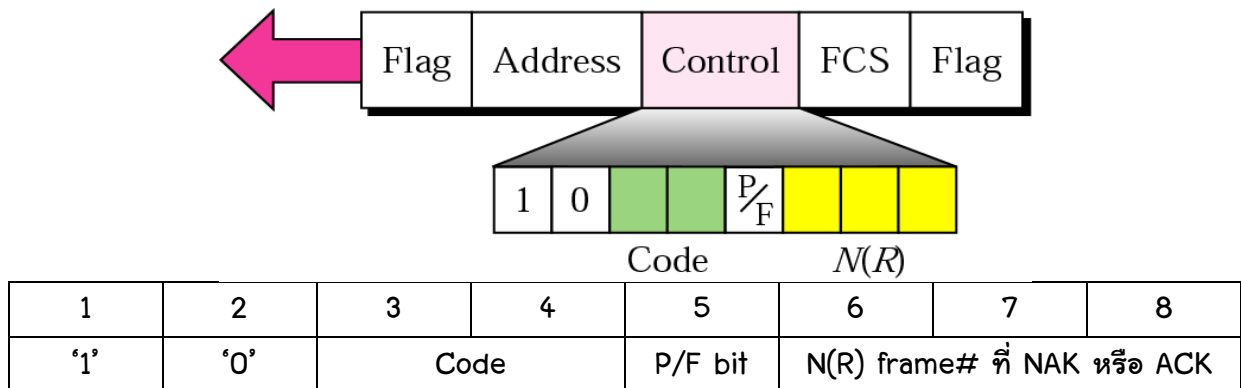


- P/F bit

P (Poll) = '1' เมื่อ Master สั่งให้ Slave ส่งข้อมูลให้ (คำสั่ง poll)
 '0' เมื่อ Master ส่งข้อมูลให้ Slave (คำสั่ง select)

F (Final) = '1' เมื่อเป็น frame สุดท้ายที่ส่งให้ Master
 '0' เมื่อยังมี frame ต่อไปอีก

- S-frame



- Code

'00': RR (Receive Ready) พร้อมรับ (ไม่พร้อมส่ง) เหมือนกับการ ACK
 '01': REJ (Reject) -> NAK for go-back-N
 '10': RNR (Receive Not Ready) ไม่พร้อมรับ (พร้อมส่ง)
 '11': SREJ (Selective Reject) -> NAK for Selective-reject

- U-frame

