การตรวจจับข้อผิดพลาด การควบคุมการไหลของข้อมูลและการควบคุมข้อผิดพลาด (ERROR DETECTION, FLOW CONTROL AND ERROR CONTROL)

วัตถุประสงค์

1. เปรียบเทียบความแตกต่างระหว่างข้อผิดพลาดแบบบิตเดียวกับแบบหลายบิตได้
2. สามารถคำนวณเพื่อตรวจจับข้อผิดพลาดตามวิธี Parity Checks, Checksum และ CRC ได้
3. บอกสาเหตุที่ต้องมีการควบคุมการไกลของข้อมูลและการควบคุมข้อผิดพลาดได้
4. เข้าใจหลักการทำงานของวิธีการควบคุมการไหลของข้อมูลแบบ Stop-and-Wait และ Sliding Windows
5. บอกวิธีการจัดการกับข้อผิดพลาดในกรณีต่างๆ ได้
6. เข้าใจกระบวนการควบคุมข้อผิดพลาดด้วยวิธี Stop-and-Wait ARQ และ Continuous ARQ

การตรวจจับข้อผิดพลาด การควบคุมการไหลของข้อมูล และการควบคุมข้อผิดพลาด

การทำงานของชั้นสื่อสารดาต้าลิงค์ คือ ทำการแปลงสัญญาณข้อมูลในรูปแบบเฟรมจากชั้นสื่อสารฟิสิคัลเพื่อส่งผ่านไปยังลิงค์จากโหนดหนึ่งไปยังโหนดถัดไป

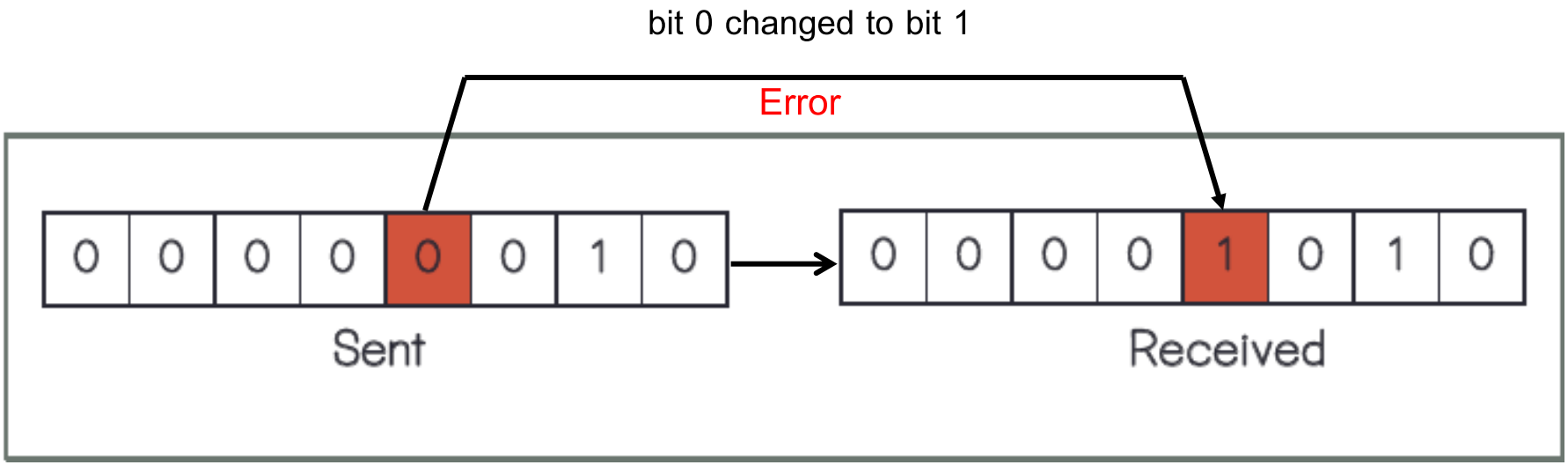
ชั้นสื่อสารดาต้าลิงค์จะต้องสร้างความน่าเชื่อให้กับชั้นสื่อสารฟิสิคัล ด้วยวิธีการดังนี้

- เพิ่มกลไกในการตรวจจับข้อผิดพลาด
- ทำการส่งเฟรมข้อมูลซ้ำ กรณีที่เฟรมข้อมูลเดิมเสียหาย
- การกำจัดเฟรมข้อมูลซ้ำซ้อน
- การจัดการกับเฟรมข้อมูลที่สูญหาย

ชนิดของข้อผิดพลาด (Types of Errors)

สัญญาณรบกวนที่เกิดขึ้นในระหว่างการสื่อสาร อาจทำให้เกิดข้อผิดพลาดขึ้น โดยข้อผิดพลาดสามารถแบ่งได้เป็น 2 ชนิด คือ

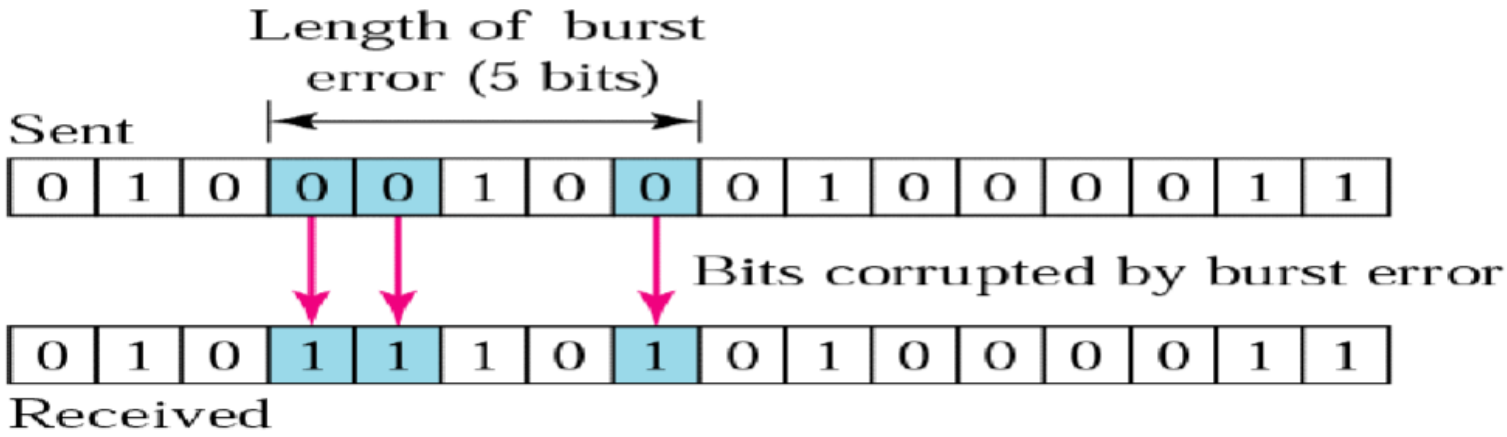
1. **ข้อผิดพลาดแบบบิตเดียว** (Single-Bit Error) ข้อผิดพลาดชนิดนี้จะมีเพียงบิตเดียวเท่านั้นที่ผิดพลาด เช่น มีการเปลี่ยนค่าจากบิต 1 เป็นบิต 0 หรือจากบิต 0 เป็นบิต 1



ชนิดของข้อผิดพลาด (Types of Errors)

2. **ข้อผิดพลาดแบบหลายบิต** (Burst Error)

เป็นข้อผิดพลาดที่จะมีจำนวนบิตตั้งแต่ 2 บิตขึ้นไปที่เกิดข้อผิดพลาด



วิธีตรวจจับข้อผิดพลาด (Error Detection Methods)

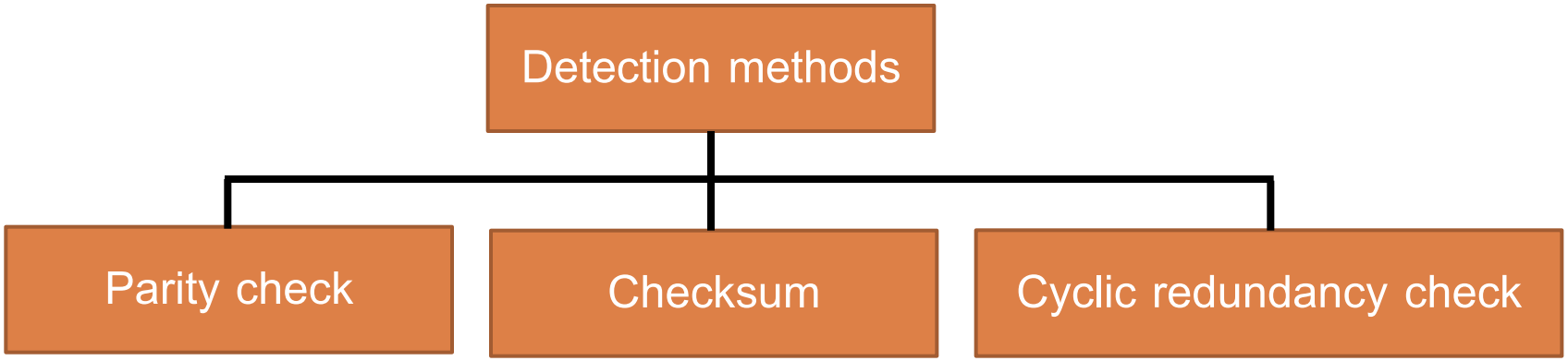
ตามหลักการ เครือข่ายจะต้องสามารถถ่ายโอนข้อมูลจากอุปกรณ์หนึ่งไปยังอุปกรณ์อื่นๆ ได้อย่างถูกต้องและสมบูรณ์ แต่ไม่สามารถรับประกันได้ว่าข้อมูลที่ส่งไปถึงปลายทางจะเป็นข้อมูลที่เหมือนกับที่ส่งมาจากต้นทางหรือข้อมูลอาจเกิดข้อผิดพลาดไปจากเดิมในระหว่างการเดินทาง เนื่องมาจากปัจจัยแวดล้อมต่างๆ เช่น การลดทอนสัญญาณ การถูกสัญญาณรบกวน ซึ่งมีผลกระทบต่อข้อมูลโดยตรง

ดังนั้นเครือข่ายที่ดีต้องมีความน่าเชื่อถือและไว้วางใจได้ จึงจำเป็นต้องมีกลไกในการตรวจจับข้อผิดพลาดและการแก้ไข (Detection and Correction) ข้อผิดพลาดที่เกิดขึ้น

วิธีตรวจจับข้อผิดพลาด (Error Detection Methods)

การตรวจสอบข้อผิดพลาดข้อมูลโดยวิธีการใช้บิตตรวจสอบนั้น สามารถทำได้ 3 วิธีการ ดังนี้

1. **การใช้บิตตรวจสอบ** (Parity check)
2. **การหาผลรวม** (Checksum)
3. **การใช้วิธี** CRC (Cyclic redundancy check)



วิธีตรวจจับข้อผิดพลาด (Error Detection Methods)

1. **การใช้บิตตรวจสอบ** (Parity Checks) เป็นวิธีตรวจจับข้อผิดพลาดอย่างง่าย โดยจะใช้บิตพาริตี้ซึ่งประกอบด้วยเลขไบนารี 0 หรือ 1 ปะท้ายเพิ่มเข้ามาอีกหนึ่งบิต เพื่อใช้เป็นบิตตรวจสอบวิธีนี้สามารถแบ่งได้เป็น 2 วิธี คือ

- การตรวจสอบบิตภาวะ**คู่** (Even Parity) เพิ่มบิตตรวจสอบ (0 หรือ 1) รวมกับบิตข้อมูล แล้วนับจำนวนบิต “1” ทั้งหมดให้ได้จำนวน**คู่**
- การตรวจสอบบิตภาวะ**คี่** (Odd Parity) เพิ่มบิตตรวจสอบ(0 หรือ 1) รวมกับบิตข้อมูล แล้วนับจำนวนบิต “1” ทั้งหมดให้ได้จำนวน**คี่**

Original Data	Even Parity	Odd Parity
0 0 0 0 0 0 0 0	0	1
0 1 0 1 1 0 1 1	1	0
0 1 0 1 0 1 0 1	0	1
1 1 1 1 1 1 1 1	0	1
1 0 0 0 0 0 0 0	1	0
0 1 0 0 1 0 0 1	1	0

วิธีตรวจจับข้อผิดพลาด (Error Detection Methods)

Ex. กรณีตรวจสอบบิตแบบภาวะคู่ (Even Parity)

- หากบิตข้อมูลมีค่าเท่ากับ 0100110 บิตพาริตี้ที่เพิ่มเข้าไปคือ บิต 1

  01001101 **<-- ที่เติม bit 1 เพื่อให้นับบิต 1 แล้วเป็นเลขคู่**

- หากบิตข้อมูลมีค่าเท่ากับ 0100110 บิตพาริตี้ที่เพิ่มเข้าไปคือ บิต 0

  01000100 **<-- ที่เติม bit 0 เพื่อให้นับบิต 1 แล้วเป็นเลขคู่**

Original Data	Sender Parity Bit	Transmitted Information	Received Calculated Parity Bit	Agree
0100110	1	01001101	1	Yes
0100110	1	01001001	0	No

แสดงการตรวจสอบบิตภาวะคู่ ซึ่งมีบิตหนึ่งเกิดการเปลี่ยนแปลงทำให้ตรวจจับข้อผิดพลาดพบ

วิธีตรวจจับข้อผิดพลาด (Error Detection Methods)

การตรวจจับข้อผิดพลาดด้วยวิธีการใช้บิตตรวจสอบนั้น มีข้อเสียตรงที่หากข้อมูลมีการเปลี่ยนแปลงหลายๆ บิตพร้อมกันเป็นคู่ๆ จะทำให้ไม่สามารถตรวจพบข้อผิดพลาดใดๆ

Original Data	Sender Parity Bit	Transmitted Information	Received Calculated Parity Bit	Agree
0100110	1	01001101	1	Yes
0100110	1	01000001	1	Yes

แสดงการตรวจสอบบิตภาวะคู่ ซึ่งมี 2 บิต เปลี่ยนแปลงทั้งคู่ ทำให้ตรวจจับข้อผิดพลาดไม่พบ

วิธีตรวจจับข้อผิดพลาด (Error Detection Methods)

2. **การหาผลรวม** (Checksum) เป็นอีกวิธีหนึ่งในการตรวจจับข้อผิดพลาดที่มีประสิทธิภาพสูงกว่าการใช้บิตตรวจสอบแต่จะมีการใช้โอเวอร์เฮดมากกว่า โดยฝั่งส่งจะคำนวณหาผลรวมข้อมูลและส่งไปพร้อมกับข้อมูล เมื่อฝั่งรับได้รับข้อมูลแล้ว จะนำผลรวมนั้นไปตรวจสอบกับผลรวมของข้อมูลที่ได้รับเข้ามาว่าตรงกันหรือไม่ ในการหาผลรวมทำได้โดยนำค่าตัวเลขของข้อมูลมารวมกัน เช่น ค่าตัวเลขของรหัส ASCII ของคำว่า Networks มีการใช้ Checksum ขนาด 16 บิต

N	e	t	w	o	r	k	s
4E	65	74	77	6F	72	6B	73

สามารถหาผลรวมของข้อมูลคำว่า Networks ได้ด้วยการนำค่าตัวเลขของรหัสแอสกีมารวมกัน ดังนี้

4E65 + 7477 + 6F72 + 6B73 = 19DC1

วิธีตรวจจับข้อผิดพลาด (Error Detection Methods)

การตรวจจับข้อผิดพลาดด้วยวิธี Checksum นี้ ยังมีข้อเสียตรงที่หากค่าข้อมูลมีการเปลี่ยนแปลงหลายตัวแล้วเกิดมีผลรวมตรงกัน จะส่งผลให้ตรวจไม่พบข้อผิดพลาด

Data Item	Checksum Value	Data Item	Checksum Value
0101	5	0100	4
0110	6	0111	7
0100	4	0101	5
0001	1	0000	0
Total	16	Total	16

จะพบว่าข้อมูลมีการเปลี่ยนแปลงที่บิตสุดท้าย แต่ผลรวมที่ได้กลับมีค่าเท่ากัน ทำให้ไม่สามารถตรวจพบข้อผิดพลาดได้ โดยตำแหน่งของบิตที่เกิดการเปลี่ยนแปลงและก่อให้เกิดข้อผิดพลาดในรูปแบบนี้ เรียกว่า Vertical Errors

วิธีตรวจจับข้อผิดพลาด (Error Detection Methods)

Ex. **ฝั่งส่ง**

สมมติว่าต้องการที่จะส่งบิตข้อมูลจำนวน 16 บิตออกไป และใช้วิธีการตรวจสอบแบบ checksum โดยมีบิตตรวจสอบ 8 บิต ซึ่งบิตข้อมูลที่ต้องการส่งมีดังนี้

10101001 00111001

วิธีทำ นำข้อมูลของทุกเซ็กเมนต์มาบวกกันด้วยวิธีแบบ 1's complement ได้ดังนี้

	10101001
	00111001
sum	11100010
Checksum	00011101

10101001 00111001 00011101

0 + 1 = 1

1 + 0 = 1

1 + 1 = 0 ทด 1

1'S Complement คือ การกลับสถานะของตัวเลขฐานสองเป็นสถานะตรงกันข้ามจากสถานะเดิม เช่น เดิมเป็นสถานะ 0 จะเปลี่ยนเป็นสถานะ 1 และถ้าเดิมเป็นสถานะ 1 จะเปลี่ยนเป็นสถานะ 0

วิธีตรวจจับข้อผิดพลาด (Error Detection Methods)

Ex. **ฝั่งรับ**

จากตัวอย่างการส่งที่ผ่านมาเมื่อฝั่งรับได้รับข้อมูลดังข้างล่างนี้ จงตรวจสอบข้อมูลดังกล่าวว่าถูกต้องหรือไม่ 10101001 00111001 00011101

วิธีทำ นำข้อมูลของทุกเซ็กเมนต์มาบวกกันด้วยวิธีแบบ 1's complement ได้ดังนี้

	10101001
	00111001
	00011101
sum	11111111
complement	00000000

0 + 1 = 1

1 + 0 = 1

1 + 1 = 0 ทด 1

วิธีตรวจจับข้อผิดพลาด (Error Detection Methods)

3. CRC (Cyclic Redundancy Checksum) การตรวจสอบแบบ CRC เป็นวิธีที่มีประสิทธิภาพสูงกว่าวิธีตรวจสอบแบบพาริตี้และเป็นที่นิยมในการนำมาใช้งานบนเครือข่ายแลนอีเทอร์เน็ต ใช้หลักการทำงานที่แตกต่างกันดังนี้

- วิธีตรวจสอบแบบพาริตี้ จะใช้กับการบวก แต่แบบ CRC จะใช้การหาร
- แบบพาริตี้สามารถแทรกบิตตรวจสอบลงในข้อมูลได้ แต่แบบ CRC จะต้องนำบิตไปตรวจสอบไปต่อท้ายบิตข้อมูล

หลักการของ CRC จะใช้รหัสโพลีโนเมียล ซึ่งมีข้อกำหนดว่าบิตซ้ายสุดและบิตขวาสุดจะต้องมีค่าเป็น 1 เสมอ อยู่ในลักษณะ Exclusive-OR โดยบิตที่นำมาบวกหรือลบกัน หากบิตตรงกันผลที่ได้จะเท่ากับ 0 ถ้าบิตต่างกันผลที่ได้จะเท่ากับ 1

วิธีตรวจจับข้อผิดพลาด (Error Detection Methods)

First Input	Second Input	XOR Output
0	0	0
0	1	1
1	0	1
1	1	0

ในการหาบิตตรวจสอบจะใช้กระบวนการหารแบบโมดูโล2 (modulo-2) ซึ่งจะไม่มีการทดเหมือนการบวกและไม่มีการยืมสำหรับการลบ เพราะฉะนั้นแล้วการบวกลบแบบนี้จะเหมือนกับการนำบิตข้อมูลมา XOR กัน ดังนั้นจะได้ว่า

0-0 = 0

0-1 = 1

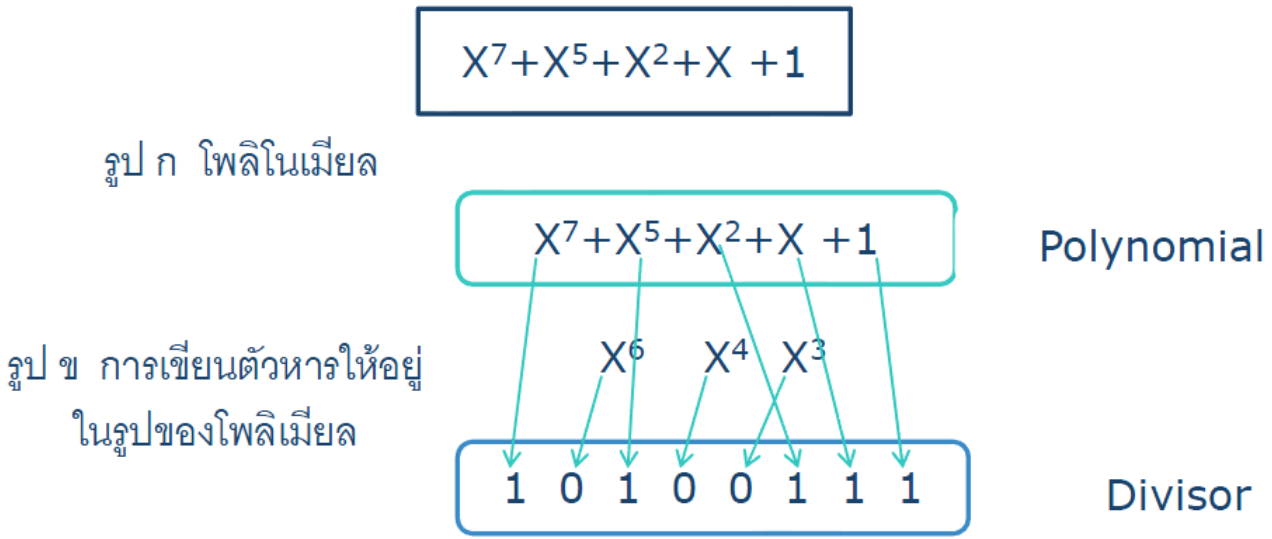
1-0 = 1

1-1= 0

วิธีตรวจจับข้อผิดพลาด (Error Detection Methods)

รหัสโพลีโนเมียล (Polynomial)

ปกติการแทนบิตข้อมูลของตัวหาร จะไม่ใช้รูปของเลขฐานสอง เนื่องจากค่อนข้างยาวและจำได้ยาก แต่จะต้องเขียนให้อยู่ในรูปของโพลิโนเมียล เช่น ถ้าตัวหารมีค่าเป็น 10100111 จะสามารถเขียนอยู่ในรูป โพลิโนเมียลได้ ดังรูป ก. ส่วนความสัมพันธ์กันระหว่างโพลิโนเมียลกับเลขฐานสองจะเป็นดังรูป ข.



วิธีตรวจจับข้อผิดพลาด (Error Detection Methods)

สำหรับตัวหารที่เป็นมาตรฐานทั่วไป ที่มีการนำไปใช้งานในโปรโตคอลต่างๆ มีดังนี้

Name	Polynomial	Application
CRC-8	x^8+x^2+x+1	ATM Header
CRC-10	$x^{10}+x^9+x^5+x^4+x^2+1$	ATM AAL
ITU-16	$x^{16}+x^{12}+x^5+1$	HDLC
ITU-32	$x^{32}+x^{26}+x^{23}+x^{22}+x^8+x^7+x^5+x^4+x^2+x+1$	LANs

วิธีตรวจจับข้อผิดพลาด (Error Detection Methods)

สรุปรายละเอียด CRC (Cyclic Redundancy Checksum)

M(X) คือเฟรมข้อมูลที่ต้องการส่ง

G(X) คือมาตรฐานของโพลีเมียลที่นำมาใช้ (Generator Polynomial)

n คือบิตศูนย์ที่นำมาผนวกเพิ่มเติม ด้วยการนำไปปะท้ายเฟรม M(X) โดยจำนวนของบิตศูนย์จะมีจำนวนเท่ากับ (Degree) ของ G(X)

R(X) คือเศษที่ได้จากการคำนวณ (Remainder) (เกิดจากการนำ M(X) ที่ได้ผนวกกับบิตศูนย์เพิ่มเติม แล้วหารด้วย G(X) โดยเศษที่ได้จากการหารก็คือ R(X))

T(X) คือเฟรมที่ส่งไป ซึ่งเกิดจากการนำ M(X) ปะด้วย R(X) โดยที่ $T(X) = M(X) + R(X)$

วิธีตรวจจับข้อผิดพลาด (Error Detection Methods)

การหาบิตตรวจสอบของเทคนิค CRC (Cyclic Redundancy Checksum)

1. ถ้าตัวหารมีจำนวนบิตเท่ากับ n+1 บิตแล้วจะต้องเติมบิต 0 จำนวน n ตัวที่ส่วนท้ายของข้อมูล
2. ใช้บิตข้อมูลลบด้วยตัวหาร (ใช้วิธี XOR) เมื่อลบแล้วผลที่ได้จากการลบ ถ้า
    2.1 บิตซ้ายสุดของเศษเป็น 1 ให้นำตัวหารมาเป็นตัวลบอีกครั้ง
    2.2 บิตซ้ายสุดของเศษเป็น 0 ให้นำ 0000 มาเป็นตัวลบ
3. ทำในข้อ 2 จนกระทั่งไม่สามารถลบกันได้อีกแล้ว (จำนวนบิตของเศษน้อยกว่าจำนวนบิตของตัวหาร) จะถือได้ว่าเศษที่ได้จากการหารนั้นคือ บิตตรวจสอบ
4. นำบิตตรวจสอบที่ได้ไปแทนที่บิต 0 จำนวน n ตัวที่ส่วนท้ายของข้อมูล

วิธีตรวจจับข้อผิดพลาด (Error Detection Methods)

EX.

1. สมมุติว่าในการรับส่งข้อมูลใช้การตรวจสอบข้อผิดพลาดแบบ CRC4 ถ้าต้องการส่งข้อมูล $X^9 + X^7 + X^4 + X^3 + X^2 + X$ โดยใช้ตัวหาร คือ $X^4 + X^2 + X + 1$ จงหาว่าข้อมูลทั้งหมด (รวม CRC) ที่ฝั่งส่งให้ฝั่งรับคืออะไร (พร้อมแสงวิธีทำทั้งฝั่งส่งฝั่งรับ)

แปลงรูปแบบของ Polynomial Codes ให้อยู่ในรูปฐานสอง

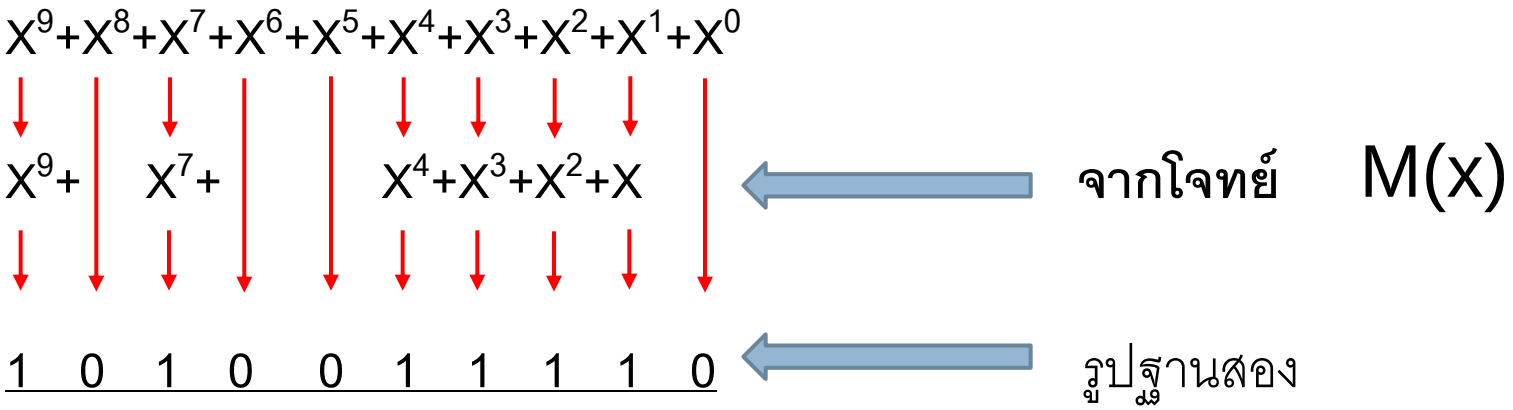
$M(x) = X^9 + X^7 + X^4 + X^3 + X^2 + X = 1010011110$

หลักการแปลง ให้เขียนดีกรีเลขทั้งหมดตั้งแต่ดีกรีสูงสุดจนถึงดีกรีต่ำสุด

จากโจทย์ เขียนจากดีกรีสูงสุดไปดีกรีต่ำสุด คือ $X^9+X^8+X^7+X^6+X^5+X^4+X^3+X^2+X^1+X^0$

วิธีตรวจจับข้อผิดพลาด (Error Detection Methods)

แปลงข้อมูลที่จะทำการส่ง หลักการแปลง ให้เขียนดีกรีเลขทั้งหมดตั้งแต่ดีกรีสูงสุดจนถึงดีกรีต่ำสุด จากโจทย์ เขียนจากดีกรีสูงสุดไปดีกรีต่ำสุด คือ $X^9+X^8+X^7+X^6+X^5+X^4+X^3+X^2+X^1+X^0$



มีใส่ 1 ไม่มีใส่ 0

ดังนั้นค่า M(x) = 1010011110

วิธีตรวจจับข้อผิดพลาด (Error Detection Methods)

แปลงตัวหาร หลักการแปลง ให้เขียนดีกรีเลขทั้งหมดตั้งแต่ดีกรีสูงสุดจนถึงดีกรีต่ำสุด

จากโจทย์ เขียนจากดีกรีสูงสุดไปดีกรีต่ำสุด คือ $X^4+X^3+X^2+X^1+X^0$

$X^4+X^3+X^2+X^1+X^0$

X^4+ X^2+X+1 ← **จากโจทย์** $G(x)$

1 0 1 1 1 ← รูปฐานสอง

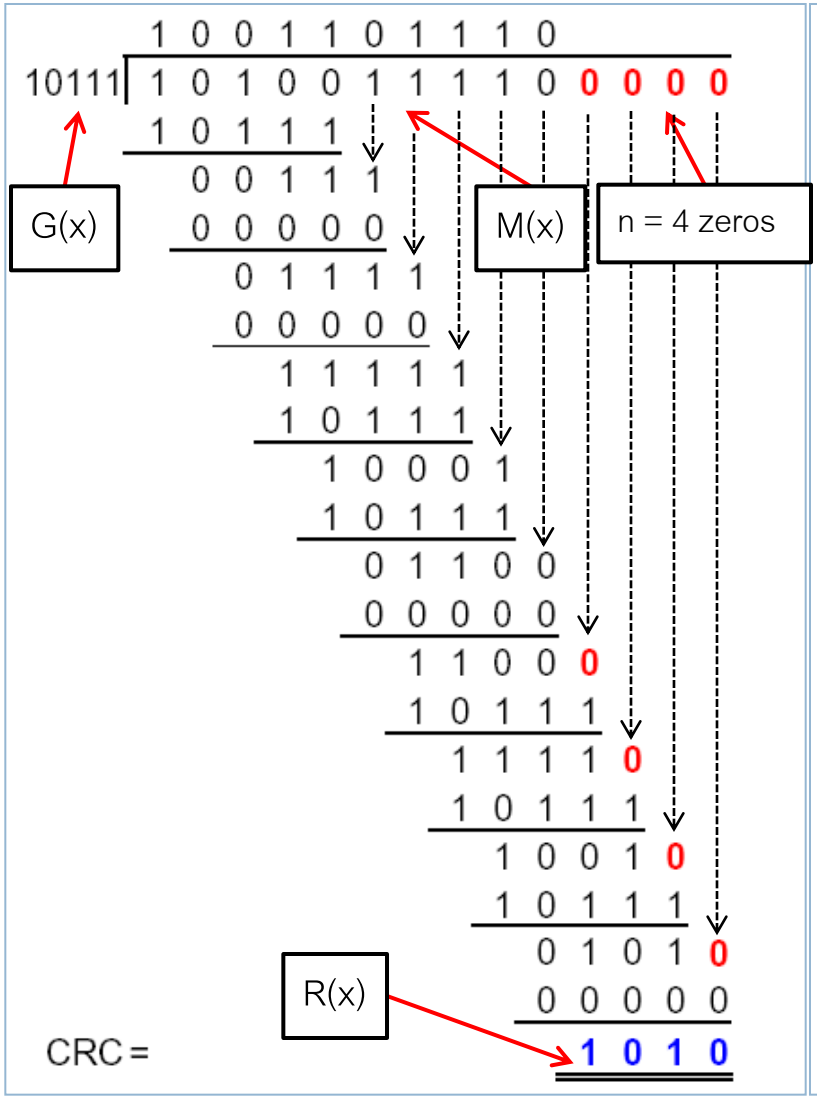
มีใส่ 1 ไม่มีใส่ 0

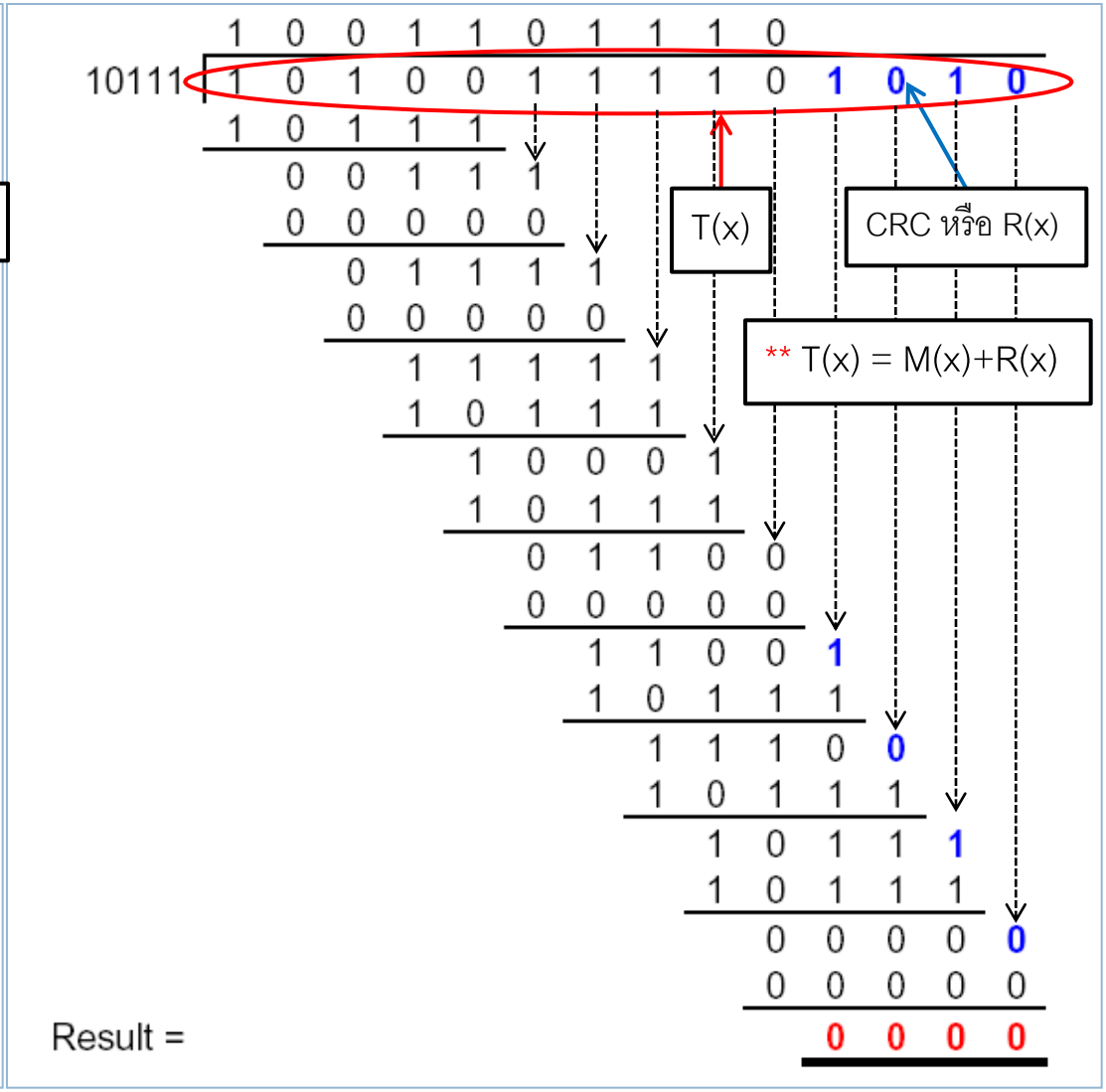
n คือจำนวนดีกรีของ $G(x)$ จากโจทย์คือ 4

$X^4 + X^2 + X + 1$

ดังนั้นค่า G(x) = 10111 และค่า n = 0000

วิธีตรวจจับข้อผิดพลาด (Error Detection Methods)





วิธีตรวจจับข้อผิดพลาด (Error Detection Methods)

จากรูป เฟรมที่ส่งไปยังปลายทางคือ

T(x) = M(x)+R(x)

= 1010011110 1010

หลังจากคำนวณ CRC หรือ R(x) เป็นที่เรียบร้อยแล้ว ฝั่งส่งก็จะส่ง T(x) ไปยังปลายทาง เมื่อปลายทางได้รับเฟรมดังกล่าว ก็จะนำ T(x) ไปหารด้วย G(x) ที่เป็นรหัสเดียวกันกับฝั่งส่ง โดยผลลัพธ์จากการคำนวณ T(x)/ G(x) จะต้องหารลงตัว หรือมีเศษเป็นศูนย์เสมอ ในกรณีที่ตรวจจับไม่พบข้อผิดพลาด ดังรูปขวามือ

การควบคุมการไหลของข้อมูล (Flow Control)

การสื่อสารบนเครือข่าย อาจมีข้อผิดพลาดเกิดขึ้นจากปัจจัยต่างๆ เช่น สัญญาณรบกวน ทำให้เกิดความไม่สมบูรณ์ระหว่างการสื่อสาร จึงต้องมีการควบคุมการไหลของข้อมูล และควบคุมข้อผิดพลาด เนื่องจาก

1. ในกรณีที่ฝั่งส่งและฝั่งรับสื่อสารอยู่บนความเร็วที่แตกต่างกัน
2. หากเฟรมข้อมูลที่ส่งเกิดความเสียหายหรือสูญเสียจะทำการสื่อสารโต้ตอบกันได้อย่างไร
3. จะเกิดอะไรขึ้น หากฝั่งรับไม่ได้รับข่าวสารที่ส่งมา
4. จะเกิดอะไรขึ้น หากเฟรมข้อมูลของฝั่งส่งเกิดความเสียหาย

การควบคุมการไหลของข้อมูล (Flow Control)

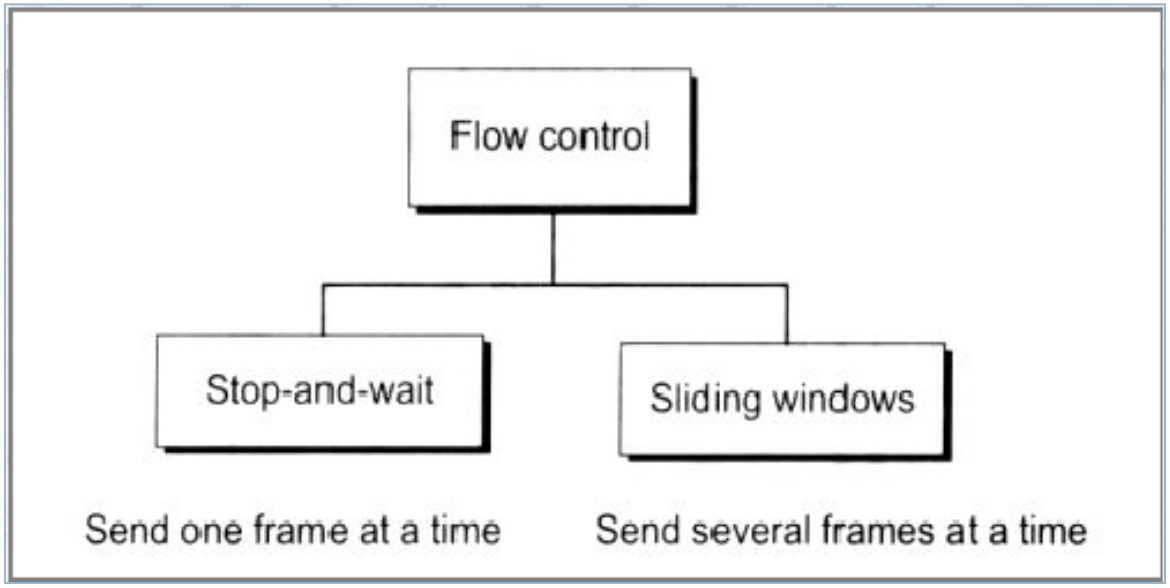
การควบคุมการไหลของข้อมูล เป็นวิธีการควบคุมการส่งข้อมูลจากผู้ส่งไปยังผู้รับ เพื่อไม่ให้ข้อมูลถูกส่งออกไปมากเกินกว่าที่ผู้รับจะสามารถรับได้ เนื่องมาจาก ผู้ส่งและผู้รับมีอัตราเร็วในการรับส่งข้อมูลไม่เท่ากัน อาจทำให้ข้อมูลสูญหายได้

ตัวอย่างเช่น การติดต่อระหว่างคอมพิวเตอร์กับเครื่องพิมพ์ ซึ่งเครื่องพิมพ์จะมีบัฟเฟอร์สำหรับเก็บข้อมูลชั่วคราว หากผู้ส่ง (คอมพิวเตอร์) ทำการส่งข้อมูลไปยังผู้รับ (เครื่องพิมพ์) โดยข้อมูลมีมากเกินกว่าขนาดของบัฟเฟอร์จะรองรับได้ จะทาให้บัฟเฟอร์เต็ม และข้อมูลเกิดการสูญหายได้

การควบคุมการไหลของข้อมูล (Flow Control)

การควบคุมการไหลของข้อมูล แบ่งได้เป็น 2 ประเภท คือ

- การควบคุมการไหลของข้อมูลด้วยวิธีหยุดและรอ
- การควบคุมการไกลของข้อมูลด้วยวิธีเลื่อนหน้าต่าง



การควบคุมการไหลของข้อมูล (Flow Control)

1. **การควบคุมการไหลของข้อมูลด้วยวิธีหยุดและรอ** (Stop-and-Wait Flow Control)

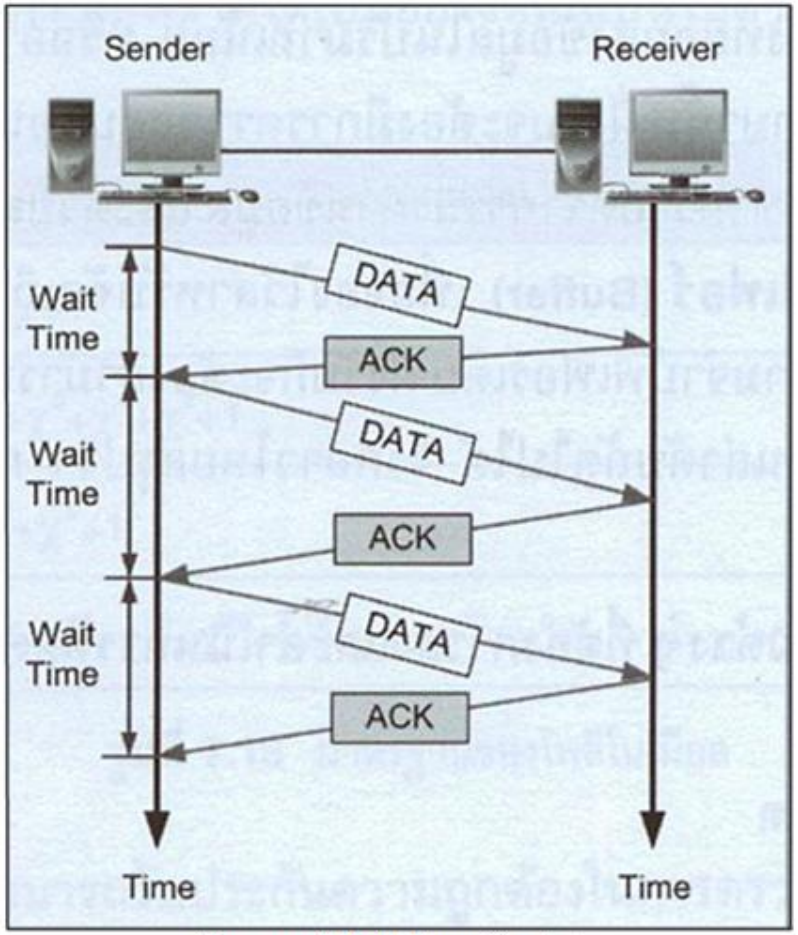
- วิธีนี้ฝั่งส่งจะส่งเฟรมข้อมูลมาให้ 1 เฟรมก่อน และรอการตอบ Acknowledge (ACK) จากฝั่งรับ
- เมื่อฝั่งส่งได้รับสัญญาณ ACK จากฝั่งรับ ก็จะดำเนินการส่งเฟรมในลำดับถัดไป
- ดังนั้นแต่ละเฟรมที่ฝั่งส่งได้ส่งไป จะต้องได้รับการ ACK จากฝั่งรับเสมอ
- ในกรณีที่ฝั่งรับต้องการหยุดการรับข้อมูลชั่วคราว ก็จะไม่ส่ง ACK กลับไป

การควบคุมการไหลของข้อมูล (Flow Control)

ข้อเสียของวิธีนี้คือ ความล่าช้า เนื่องจากเฟรมทุกเฟรมที่เดินทางไปยังฝั่งรับ ต้องได้รับการ ACK ก่อนเสมอ ฝั่งส่งจึงจะสามารถส่งเฟรมลำดับถัดไปได้ และหากระยะทางของฝั่งส่งและฝั่งรับอยู่ไกลกัน จะต้องเสียเวลารอคอยการตอบรับ ACK ในแต่ละเฟรม

ได้มีการปรับปรุงประสิทธิภาพของวิธีนี้ ด้วยการ**เพิ่มรหัส** NAK โดยเป็นสัญญาณที่ผู้รับจะส่งให้กับผู้ส่ง ในกรณีที่พบข้อผิดพลาดของข้อมูล ซึ่งเมื่อผู้ส่งได้รับสัญญาณ NAK จะส่งข้อมูลกลับมาให้อีกครั้งทันที

การควบคุมการไหลของข้อมูล (Flow Control)



Stop-and-Wait Flow Control

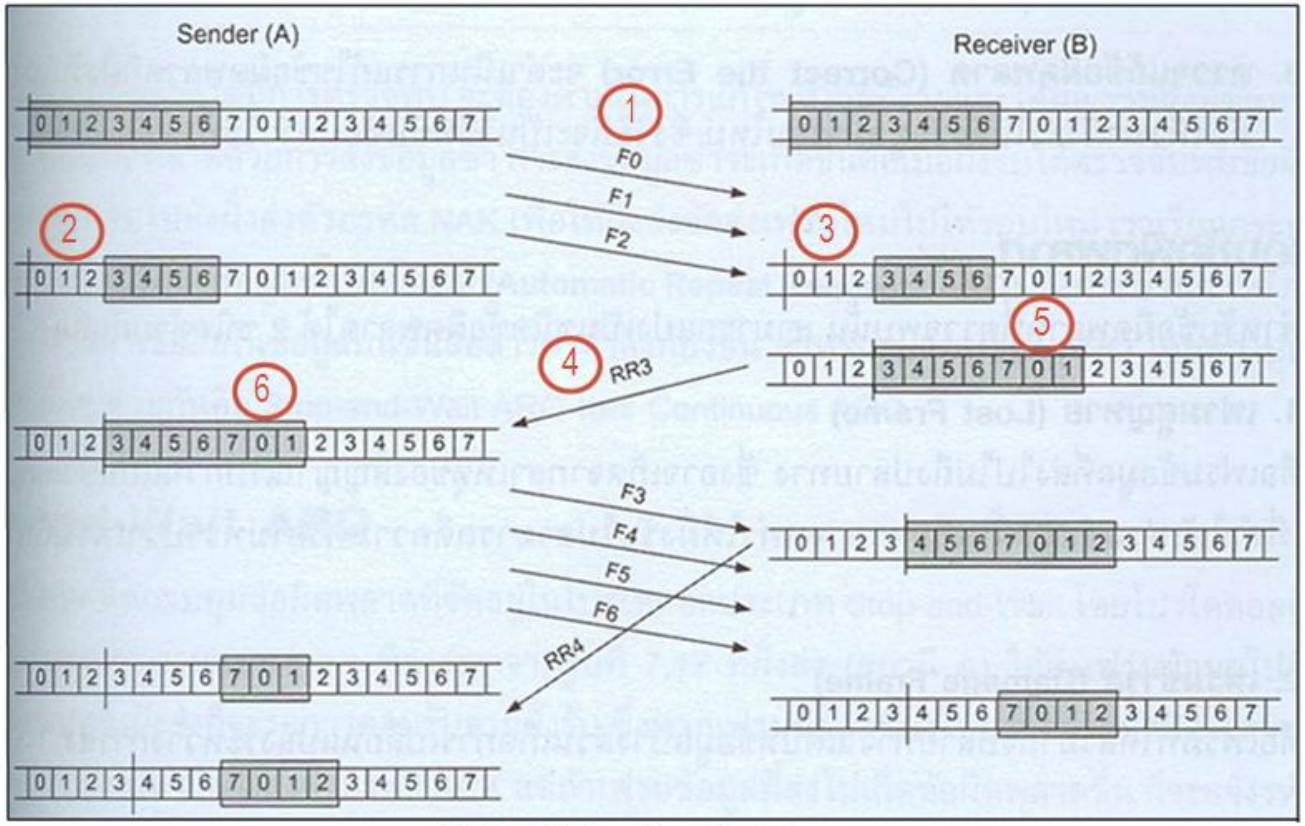
การควบคุมการไหลของข้อมูล (Flow Control)

2. **การควบคุมการไหลของข้อมูลด้วยวิธีเลื่อนหน้าต่าง** (Sliding-Window Flow Control)

วิธีนี้ฝั่งส่งสามารถส่งเฟรมข้อมูลหลายๆ เฟรมก่อนที่จะได้รับการตอบรับ (ACK) ดังนั้นการตอบรับ ACK ในหนึ่งครั้งจะหมายถึง การได้รับเฟรมมาแล้วหลายเฟรม

- ฝั่งส่งและฝั่งรับมีเฟรมในการรับและส่งข้อมูลได้เท่ากัน
- เมื่อฝั่งส่งส่งข้อมูลออกไป กรอบจำนวนเฟรมจะลดลง และเมื่อฝั่งรับได้รับข้อมูลกรอบจำนวนเฟรมของผู้รับจะลดลง
- เมื่อผู้รับได้รับข้อมูลจะส่งสัญญาณ ACK กลับไปพร้อมกับหมายเลขเฟรมถัดไปที่ฝั่งรับจะรับได้ พร้อมกับเพิ่มกรอบจำนวนเฟรมให้เท่าเดิม
- เมื่อฝั่งส่งได้รับสัญญาณตอบรับ และหมายเลขเฟรมถัดไปแล้ว จะทำการขยายขนาดกรอบจำนวนเฟรมให้เท่าเดิม

การควบคุมการไหลของข้อมูล (Flow Control)



Sliding-Window Flow Control

เป็นการรับส่งข้อมูลระหว่างสถานี A และ B ซึ่งสามารถบรรจุเฟรมได้สูงสุด 7 เฟรม

1. ฝั่งส่งส่งเฟรม F0,F1,F2 ไปยังฝั่งรับ โดยยังไม่ได้รับการตอบรับ Acknowledge สถานี A ก็จะหดหน้าต่างเหลือ 4 เฟรม
2. ฝั่งส่งสามารถส่งเฟรมทั้งสี่ด้วยการเริ่มต้นที่เฟรม F3
3. ฝั่งรับได้ตอบรับรหัส RR3 หมายความว่าได้รับเฟรมจนถึง F2 พร้อมที่จะรับเฟรมตั้งแต่ F3 จนถึงอีก 7 เฟรมถัดไป
4. ฝั่งส่งเมื่อได้รับ Acknowledge ก็จะเลื่อนหน้าต่าง โดยเริ่มต้นที่เฟรมหมายเลข 3 ถัดไปจนครบ 7 เฟรม (F3,F4,F5,F6,F7,F0,F1)
5. ฝั่งส่งส่งเฟรม F3,F4,F5,F6 ไปอย่างรวดเร็ว
6. ขณะนั้นฝั่งรับได้รับเฟรม F3 และมีการตอบรับ RR4 บอกว่าพร้อมที่จะรับเฟรม F4 และถัดไปอีก 7 เฟรม (F4,F5,F6,F7,F0,F1,F2)
7. ช่วงที่มีการตอบรับ RR4 ฝั่งส่งได้ส่งเฟรม F4,F5,F6 ไปก่อนแล้ว ดังนั้นฝั่งส่งก็เปิดหน้าต่างเพื่อส่ง 4 เฟรมถัดไปโดยเริ่มจากเฟรม F7

RR คือ Receive Ready

การควบคุมข้อผิดพลาด (Error Control)

การควบคุมข้อผิดพลาดจะเกี่ยวข้องกับวิธีการตรวจสอบข้อผิดพลาดของเฟรม และหากเกิดข้อผิดพลาดขึ้น จะต้องดำเนินการอย่างไร เพื่อให้มั่นใจได้ว่า เฟรมข้อมูลทั้งหมดที่ส่งไปยังปลายทางไม่มีข้อผิดพลาดใดๆ

การดำเนินการกับข้อผิดพลาด

เมื่อฝั่งรับตรวจพบข้อผิดพลาดจากข้อมูลที่ส่งมา ฝั่งรับสามารถดำเนินการกับข้อผิดพลาดที่เกิดขึ้นได้ 3 กรณี คือ

1. **ไม่ต้องดำเนินการใดๆ** (Do nothing) จะปล่อยเฟรมข้อมูลที่ผิดพลาดไป แล้วให้ชั้นสื่อสารที่อยู่เหนือกว่าไปจัดการแทน
2. **แจ้งกลับไปให้ฝั่งส่งรับทราบ** (Return a message) เพื่อให้ฝั่งส่งทำการส่งข้อมูลส่วนที่เสียหายมาให้อีกครั้ง
3. **ตรวจแก้ข้อผิดพลาด** (Correct the Error) จะดำเนินการแก้ไขข้อผิดพลาดที่ฝั่งรับเองโดยไม่ต้องให้ฝั่งส่งส่งข้อมูลมาใหม่ ซึ่งเป็นวิธีที่ซับซ้อนกว่าวิธีทั้งหมด

การควบคุมข้อผิดพลาด (Error Control)

ชนิดของข้อผิดพลาด

สำหรับข้อผิดพลาดที่ตรวจพบนั้น สามารถแบ่งเป็นชนิดของข้อผิดพลาด 2 ชนิด

1. **เฟรมสูญหาย** (Lost Frame) คือ เฟรมข้อมูลที่ส่งไปไม่ถึงปลายทาง ซึ่งอาจเกิดจากสาเหตุของสัญญาณรบกวนที่ทำให้เฟรมข้อมูลเสียหาย จนทำให้ฝั่งรับไม่สามารถตีความหรือไม่ทราบว่าเฟรมนั้นส่งมาถึง

2. **เฟรมชำรุด** (Damage Frame) คือ เฟรมสามารถส่งไปถึงปลายทาง แต่บิตของข้อมูลบางส่วนเกิดการเปลี่ยนแปลงระหว่างการส่ง

การควบคุมข้อผิดพลาด (Error Control)

เทคนิคการควบคุมข้อผิดพลาด จะอยู่บนพื้นฐานของส่วนประกอบต่างๆ ดังนี้

การตรวจจับข้อผิดพลาด ปลายทางจะมีการนำเฟรมที่ได้รับมาทำการตรวจจับข้อผิดพลาด ด้วยเทคนิควิธีการต่างๆ

การตอบรับ ACK ปลายทางจะตอบรับ ACK เมื่อได้รับข้อมูลอย่างสมบูรณ์ โดยไม่มีข้อผิดพลาดใดๆ

การส่งข้อมูลรอบใหม่หลังจากรอจนหมดเวลา (Timeout) ฝั่งส่งจะทำการส่งเฟรมข้อมูลรอบใหม่ทันที ในกรณีที่ปลายทางไม่ตอบรับกลับมาภายในเวลาที่กำหนด ก็คือเกิด Timeout

การควบคุมข้อผิดพลาด (Error Control)

การตอบรับ NAK และการส่งข้อมูลรอบใหม่ ปลายทางจะมีการตอบรับ NAK (Negative Acknowledgement) กลับมาที่ฝั่งส่ง ในกรณีที่เฟรมที่ได้รับนั้นเกิดข้อผิดพลาด เมื่อฝั่งส่งได้รับการตอบรับ NAK ก็จะทราบว่าข้อมูลที่ส่งไปนั้นมีข้อผิดพลาด จะดำเนินการส่งเฟรมข้อมูลไปอีกครั้ง

การแก้ไขข้อผิดพลาดโดยการส่งข้อมูลซ้ำ (Error Correction via Retransmission)

- เมื่อมีการตรวจพบข้อผิดพลาด จะต้องดำเนินการแก้ไขโดยวิธีที่ง่ายที่สุด คือ การส่งข้อมูลซ้ำ
- การส่งข้อมูลซ้ำจะเกิดขึ้นเมื่อฝั่งรับตรวจพบข้อผิดพลาด แล้วตอบกลับฝั่งส่งด้วยรหัส NAK เพื่อให้ฝั่งส่งส่งเฟรมข้อมูลมาให้ใหม่ เรียกกระบวนการนี้ว่า **การร้องขอเพื่อส่งข้อมูลซ้ำอัตโนมัติ** (Automatic Repeat Request: ARQ) ซึ่งช่วยสร้างความน่าเชื่อถือของข้อมูลในชั้นสื่อสารดาต้าลิงค์

สำหรับรูปแบบของ ARQ มี 2 วิธีหลัก คือ

- Stop-and-Wait ARQ
- Continuous ARQ

การแก้ไขข้อผิดพลาดโดยการส่งข้อมูลซ้ำ (Error Correction via Retransmission)

Stop-and-Wait ARQ

• เป็นเทคนิคควบคุมข้อผิดพลาดที่มีกระบวนการทำงานแบบง่ายๆ จัดอยู่ในโปรโตคอลประเภท Stop-and-Wait โดยมีการทำงานดังนี้

- ฝั่งส่งส่งเฟรมข้อมูลไปยังฝั่งรับ และรอการตอบรับจากฝั่งรับ
- หากเฟรมข้อมูลที่ส่งไปไม่มีข้อผิดพลาด ฝั่งรับจะส่งสัญญาณมายังฝั่งส่งด้วยรหัส ACK
- แต่ถ้าเฟรมข้อมูลเกิดข้อผิดพลาดขึ้น ฝั่งรับจะส่งสัญญาณกลับมาด้วยรหัส NAK หรือ REJ (Reject) กลับไป

การแก้ไขข้อผิดพลาดโดยการส่งข้อมูลซ้ำ (Error Correction via Retransmission)

Stop-and-Wait ARQ

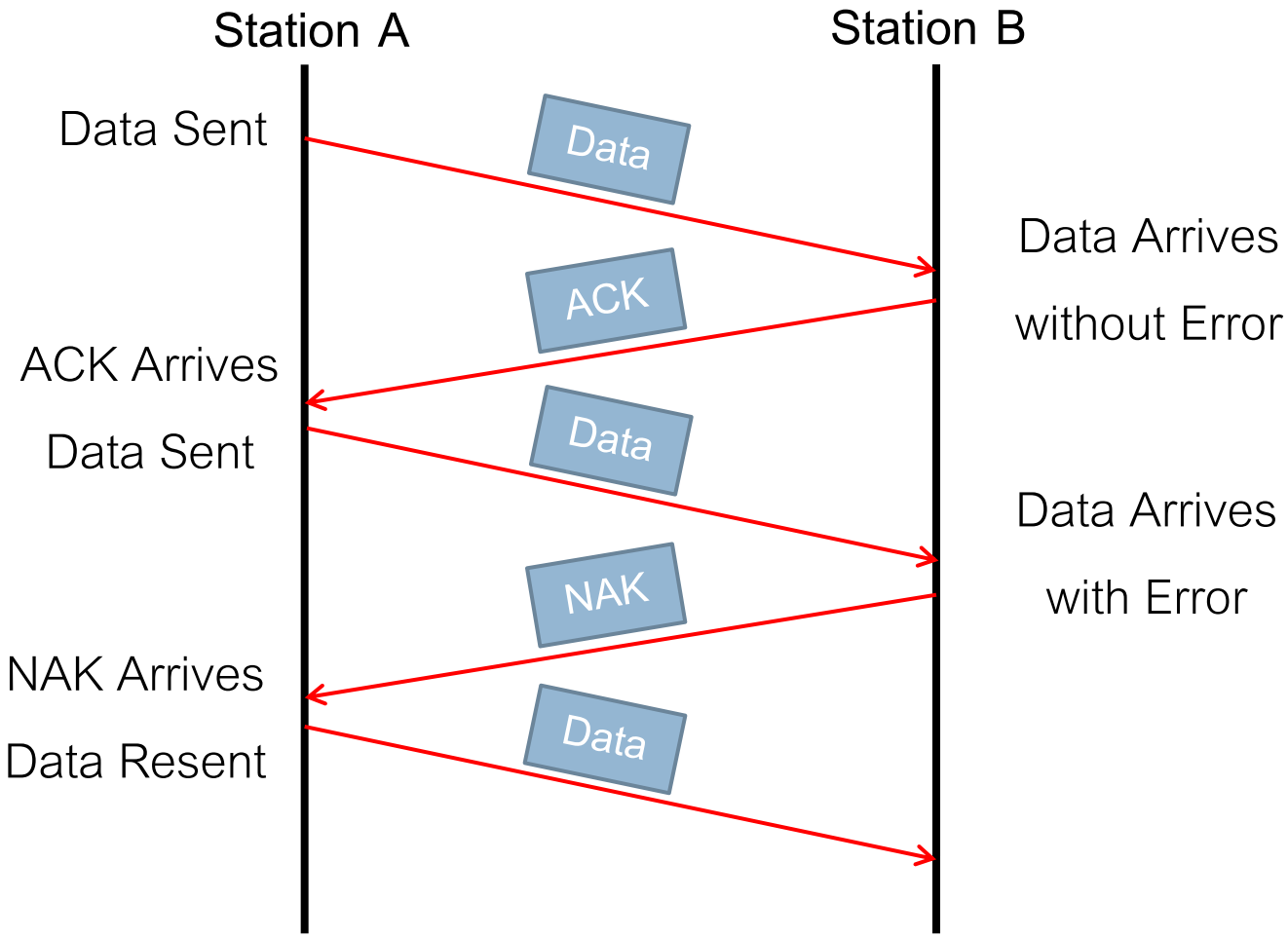
- เมื่อฝั่งส่งได้รับสัญญาณเป็นรหัส ACK ก็จะส่งเฟรมข้อมูลในลำดับถัดไป
- แต่หากได้รับสัญญาณเป็นรหัส NAK หรือ REJ ก็จะส่งเฟรมข้อมูลที่ส่งไปในขณะนั้นไปให้อีกครั้ง

• ข้อดีของวิธีนี้ คือ มีกระบวนการทำงานแบบง่าย ไม่ซับซ้อน

• ข้อเสีย คือ ทำให้เกิดการหน่วงเวลาค่อนข้างสูง เนื่องจากฝั่งส่งต้องได้รับการตอบรับจากฝั่งรับในทุกๆ ครั้งที่มีการส่งเฟรมแต่ละเฟรม

การแก้ไขข้อผิดพลาดโดยการส่งข้อมูลซ้ำ (Error Correction via Retransmission)

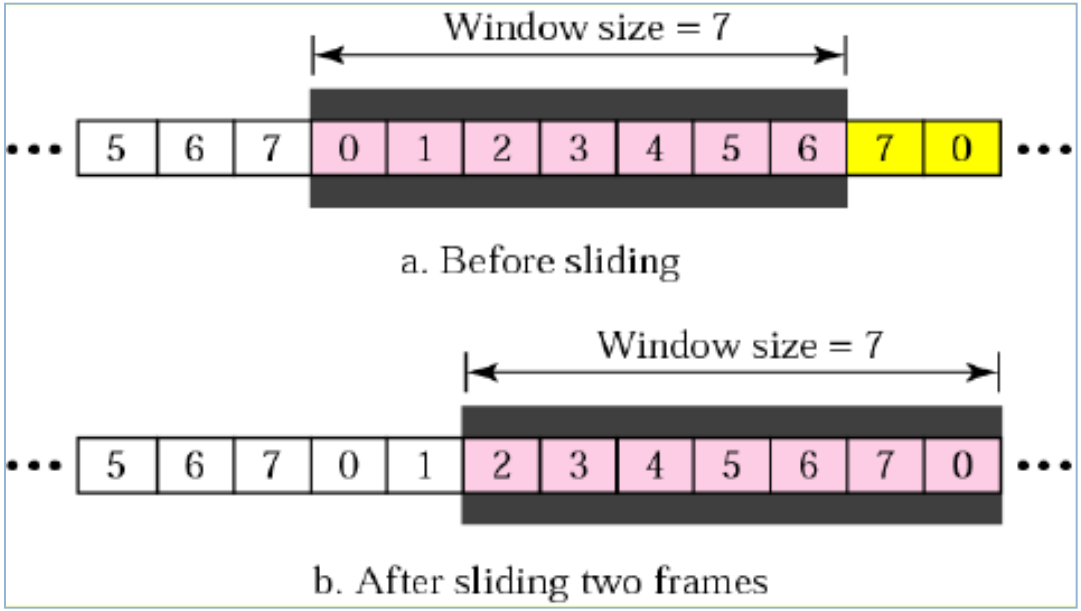
Stop-and-Wait ARQ



การแก้ไขข้อผิดพลาดโดยการส่งข้อมูลซ้ำ (Error Correction via Retransmission)

Continuous ARQ

เป็นวิธีที่จัดอยู่ในโปรโตคอลประเภท Sliding Window ที่ฝั่งส่งสามารถส่งข้อมูลไปยังฝั่งรับได้อย่างต่อเนื่อง แต่ละเฟรมที่ส่งไปไม่ต้องรอการตอบรับจากฝั่งรับโดยทันที ทำให้มีประสิทธิภาพดีกว่า



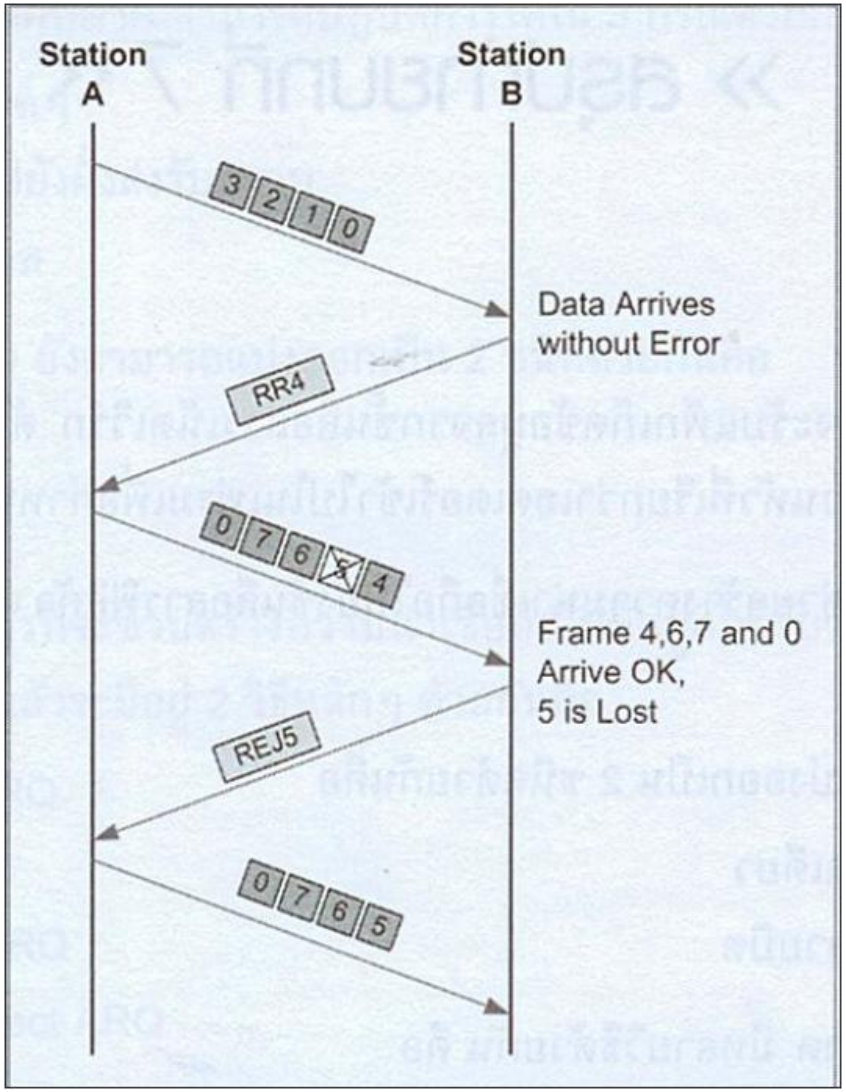
การแก้ไขข้อผิดพลาดโดยการส่งข้อมูลซ้ำ (Error Correction via Retransmission)

โดยวิธี Continuous ARQ **สามารถแบ่งได้เป็น** 2 **วิธีด้วยกัน คือ**

1. Go-Back-N ARQ

วิธีนี้ฝั่งส่งสามารถส่งเฟรมข้อมูลไปยังฝั่งรับได้อย่างต่อเนื่อง แต่ถ้าฝั่งรับตอบรับข้อผิดพลาดกลับมายังฝั่งส่ง ฝั่งส่งจะย้อนกลับไปยังตำแหน่งเฟรมที่ผิดพลาด และเริ่มต้นส่งใหม่ตั้งแต่เฟรมที่ผิดพลาดอีกครั้ง ถึงแม้ว่าจะมีเฟรมที่ส่งล่วงหน้าไปแล้วก็ตาม

การแก้ไขข้อผิดพลาดโดยการส่งข้อมูลซ้ำ (Error Correction via Retransmission)



เป็นการรับส่งข้อมูลระหว่างสถานีที่ข้อมูลตรงกลางเฟรมข้อมูลทั้งหมดเกิดสูญหาย

1. ฝั่งส่งส่งเฟรม 0,1,2,3 ไปยังฝั่งรับ
2. ฝั่งรับตอบรับ RR4 กลับมายังฝั่งส่ง
3. ฝั่งส่งเฟรมข้อมูลต่อเนื่องไปอีกคือ เฟรม 4,5,6,7,0
4. เฟรมหมายเลข 5 เกิดสูญหายระหว่างการส่งเมื่อฝั่งรับได้รับเฟรมข้อมูล และตรวจพบว่าไม่พบเฟรมหมายเลข 5 ดังนั้นจึงส่งรหัส REJ5 กลับไป
5. ฝั่งส่งได้รับรหัส Reject กลับมา จึงย้อนกลับไปยังตำแหน่งเฟรมหมายเลข 5 และเริ่มต้นส่งข้อมูลทั้งหมด คือ เฟรมหมายเลข 5,6,7 และ 0

Go-Back-N ARQ

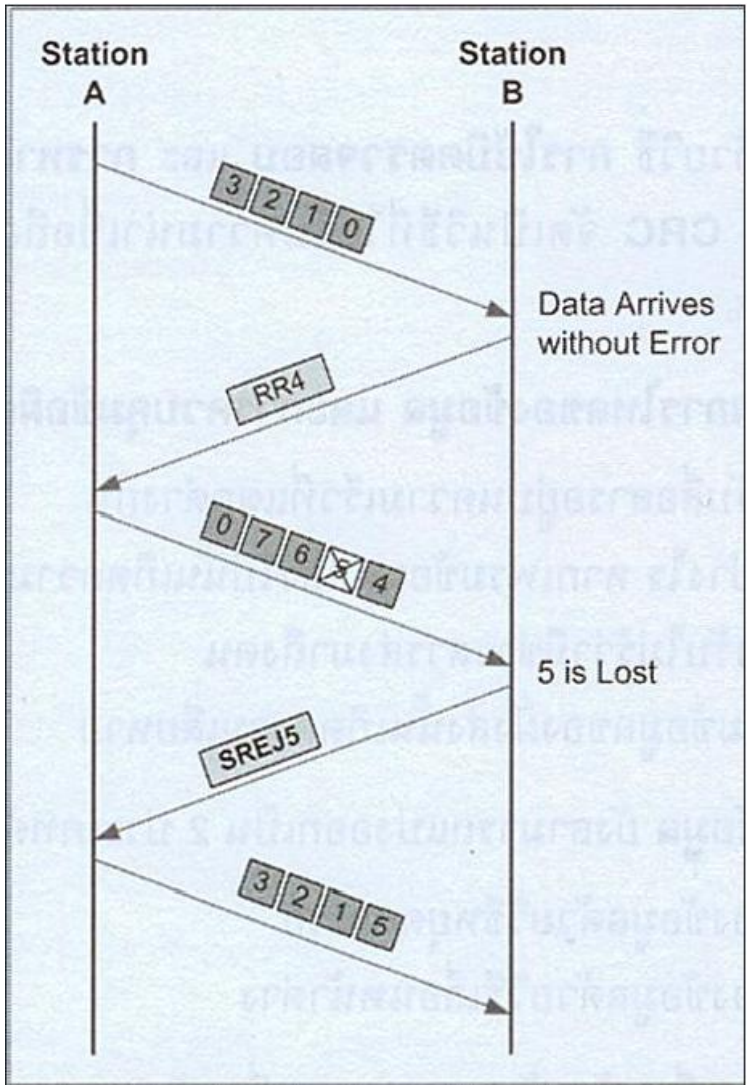
การแก้ไขข้อผิดพลาดโดยการส่งข้อมูลซ้ำ (Error Correction via Retransmission)

โดยวิธี Continuous ARQ สามารถแบ่งได้เป็น 2 วิธีด้วยกัน คือ

2. Selective-Reject ARQ

วิธีนี้คล้ายกับแบบ Go-Back-End ARQ แต่มีประสิทธิภาพมากกว่า โดยฝั่งส่งจะส่งเฉพาะเฟรมที่ผิดพลาดกลับไปเท่านั้น ส่วนเฟรมที่จะส่งในลำดับถัดไป สามารถเริ่มต้นถัดจากเฟรมที่ส่งล่วงหน้าไปได้ทันที

การแก้ไขข้อผิดพลาดโดยการส่งข้อมูลซ้ำ (Error Correction via Retransmission)

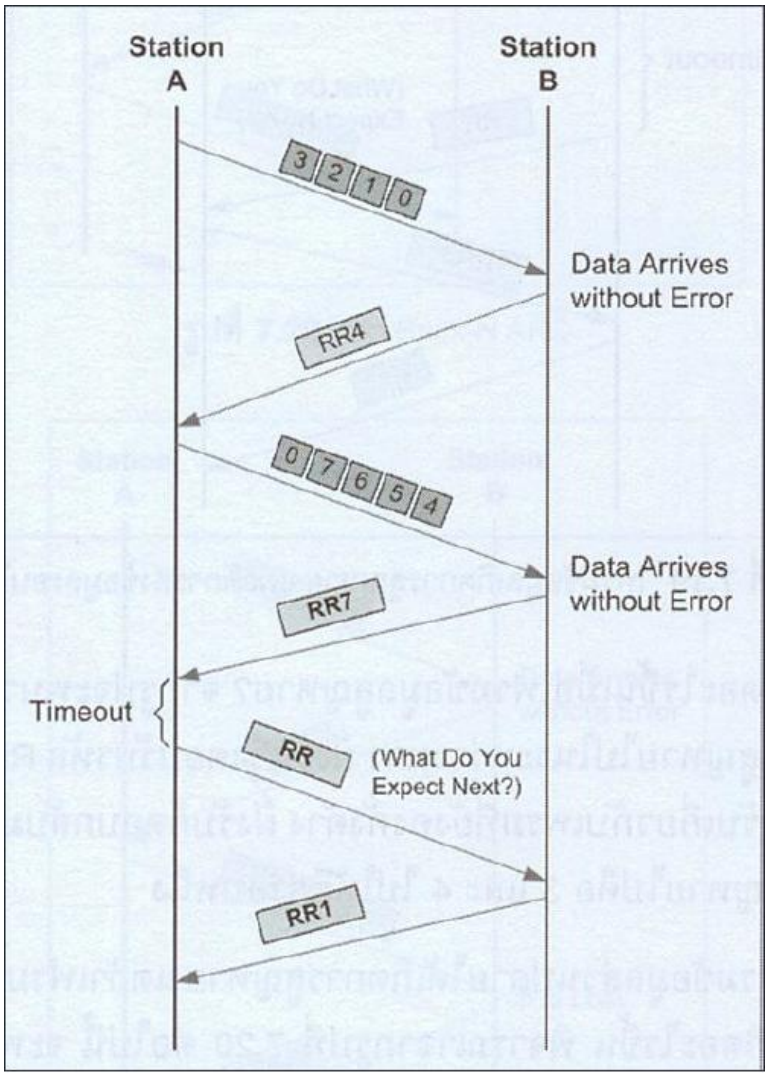


เป็นการรับส่งข้อมูลระหว่างสถานีที่ข้อมูลตรงกลาง เฟรมข้อมูลทั้งหมดเกิดสูญหาย

1. ฝั่งส่งส่งเฟรม 0,1,2,3 ไปยังฝั่งรับ
2. ฝั่งรับตอบรับ RR4 กลับมายังฝั่งส่ง
3. ฝั่งส่งเฟรมข้อมูลต่อเนื่องไปอีกคือ เฟรม 4,5,6,7,0
4. เฟรมหมายเลข 5 เกิดสูญหายระหว่างการส่งเมื่อฝั่งรับได้รับเฟรมข้อมูล และตรวจพบว่าไม่พบเฟรมหมายเลข 5 ดังนั้นจึงส่งรหัส REJ5กลับไป
5. ฝั่งส่งได้รับรหัส Reject กลับมา จะทำการส่งเพียงเฟรมข้อมูลที่สูญหายกลับไปให้เท่านั้น และเฟรมถัดไปที่ส่งก็คือ เฟรมหมายเลข 1,2,3

Selective-Reject ARQ

การแก้ไขข้อผิดพลาดโดยการส่งข้อมูลซ้ำ (Error Correction via Retransmission)

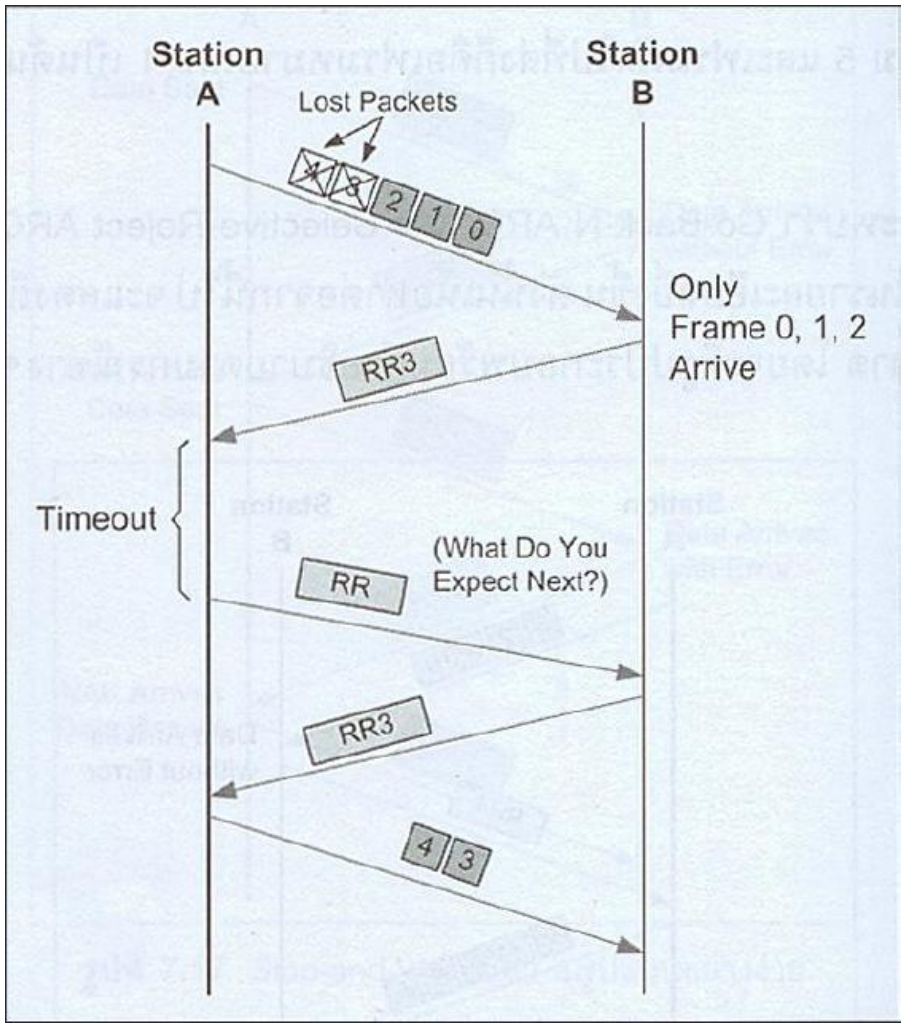


เป็นการรับส่งข้อมูลระหว่างสถานีที่ไม่พบข้อผิดพลาดในข้อมูลโดยเริ่มจาก

1. ฝั่งส่งส่งเฟรม 0,1,2,3 ไปยังฝั่งรับ
2. ฝั่งรับตอบรับ RR4 กลับมายังฝั่งส่ง
3. ฝั่งส่งเฟรมข้อมูลต่อเนื่องไปอีกคือ เฟรม 4,5,6,7,0
4. ฝั่งรับตอบรับรหัส RR7 กลับมาก่อน (ได้รับเฟรม 4,5,6)
5. มี timeout เกิดขึ้น
6. ฝั่งส่งทำการส่งรหัส RR (Receive Ready) ไปยังฝั่งรับเพื่อสอบถามว่าเฟรมที่ยังค้างอยู่ (7,0) เกิดสูญหายหรือได้รับช้าหรือไม่ และต้องการเฟรมหมายเลขใดถัดไป
7. ฝั่งรับตอบกลับมาเป็นรหัส RR1 แสดงว่าได้รับเฟรมที่ยังค้างเรียบร้อยแล้ว และต้องการเฟรมข้อมูลถัดไป คือ เฟรมหมายเลข 1

Selective-Reject ARQ

การแก้ไขข้อผิดพลาดโดยการส่งข้อมูลซ้ำ (Error Correction via Retransmission)



เป็นการรับส่งข้อมูลระหว่างสถานีที่มีเฟรมข้อมูลสูญหาย

1. ฝั่งส่งส่งเฟรม 0,1,2,3,4 ไปยังฝั่งรับแต่ 2 เฟรมสุดท้ายสูญหาย
2. ฝั่งรับตอบกลับรหัส RR3 กลับมา
3. หลังจาก Timeout ฝั่งส่งส่งรหัส RR เพื่อถามเกี่ยวกับเฟรมที่ยังคั่งค้าง
4. ฝั่งรับตอบกลับมาว่า ต้องการเฟรมหมายเลข 3 (RR3)
5. ฝั่งส่งจึงส่งเฟรมข้อมูลที่สูญหาย คือ เฟรม 3,4 ไปอีกรอบ

Selective-Reject ARQ

สรุป

ชั้นสื่อสารดาต้าลิงก์ จะรับแพ็กเก็ตข้อมูลจากชั้นสื่อสารเน็ตเวิร์ก ด้วยการแบ่งเป็นหน่วยข้อมูลที่เรียกว่าเฟรม จากนั้นก็จะปะส่วนหัวที่เรียกว่าเฮดเดอร์เข้าไปในเฟรมเพื่อกำหนดที่อยู่ของผู้ส่งและผู้รับ

ชั้นสื่อสารดาต้าลิงก์ยังช่วยสร้างความน่าเชื่อถือไปยังชั้นสื่อสารฟิสิคัล ด้วยการเพิ่มกลไกการตรวจจับข้อผิดพลาดของข้อมูล

ข้อผิดพลาด สามารถแบ่งออกเป็น 2 ชนิดด้วยกันคือ

1. ข้อผิดพลาดแบบบิตเดียว
2. ข้อผิดพลาดแบบหลายเดียว

การตรวจจับข้อผิดพลาด มีหลายวิธีด้วยกัน คือ

1. การใช้บิตตรวจสอบ
2. การหาผลรวม
3. การใช้วิธี CRC

สรุป

การตรวจสอบข้อผิดพลาดด้วยวิธี **การใช้บิตตรวจสอบ** และ **การหาผลรวม** เป็นวิธีที่ไม่รับรองความถูกต้องได้ทั้งหมด ทั้งนี้วิธีแบบ CRC จัดเป็นวิธีที่ได้รับความน่าเชื่อถือสูง เนื่องจากสามารถตรวจจับข้อผิดพลาดได้แม่นยำถึง 99.99%

สาเหตุที่ต้องมีการควบคุมการไหลของข้อมูล และการควบคุมข้อผิดพลาด ก็เนื่องมาจาก

1. ในกรณีที่ฝั่งส่งและฝั่งรับสื่อสารอยู่บนความเร็วที่แตกต่างกัน
2. จะทำการโต้ตอบกันอย่างไร หากเฟรมข้อมูลที่ส่งไปนั้นเกิดความเสียหาย หรือสูญเสีย
3. จะเกิดอะไรขึ้น หากฝั่งรับไม่รู้ว่ามีข่าวสารส่งมาถึงตน
4. จะเกิดอะไรขึ้น ถ้าเฟรมข้อมูลของฝั่งส่งนั้นเกิดความเสียหาย

การควบคุมการไหลของข้อมูล ยังสามารถแบ่งออกเป็น 2 ประเภทด้วยกัน คือ

1. การควบคุมการไหลของข้อมูลด้วยวิธีหยุดและรอ
2. การควบคุมการไหลของข้อมูลด้วยวิธีเลื่อนหน้าต่าง

สรุป

การควบคุมการข้อผิดพลาด จะเกี่ยวข้องกับกระบวนการที่จะต้องตรวจสอบข้อผิดพลาดของเฟรมอย่างไร และจะต้องทำอะไรบ้าง หากเกิดข้อผิดพลาดขึ้น เพื่อแสดงความมั่นใจได้ว่าเฟรมข้อมูลทั้งหมดที่ส่งไปยังปลายทางนั้น จะปราศจากข้อผิดพลาด

การดำเนินการกับข้อผิดพลาด สามารถปฏิบัติการได้ใน 3 กรณีด้วยกัน คือ

1. ไม่ต้องดำเนินการใดๆ
2. แจ้งข่าวสารกลับไปยังฝั่งส่งรับทราบ
3. ตรวจแก้ข้อผิดพลาด

ข้อผิดพลาดที่ตรวจพบ ยังสามารถแบ่งออกเป็น 2 ชนิดด้วยกันคือ

1. เฟรมสูญหาย
2. เฟรมชำรุด

สรุป

ARQ เป็นกระบวนการที่จะช่วยสร้างความน่าเชื่อถือในข้อมูลให้กับชั้นสื่อสารดาต้าลิงก์ยิ่งขึ้น โดยรูปของ ARQ โดยทั่วไปแล้วจะมีอยู่ 2 วิธีหลักๆ ด้วยกันคือ

1. Stop-and-Wait ARQ
2. Continuous ARQ
    - 2.1 Go-Back-N ARQ
    - 2.2 Selective-Reject ARQ

แบบฝึกหัดที่ 7

- จากโจทย์เฟรมข้อมูลที่ต้องการส่งคือ 101010 และใช้รหัสโพลีโนเมียล X^3+x^2+1 พร้อมบอกค่าดังต่อไปนี้
- M(x)
- G(x)
- R(x)
- n
- T(x)