

Data Communication

Data com -

Chapter 5

Error & Error Detection

ARTIFICIAL INTELLIGENCE

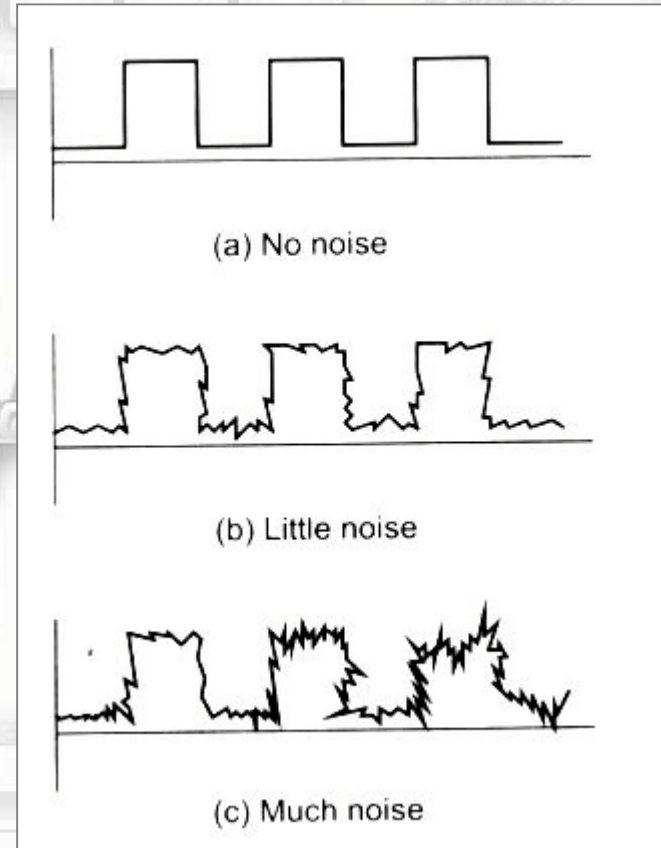


สัญญาณรบกวน(Noise)

- หมายถึงสัญญาณที่ไม่พึงประสงค์ ได้สอดแทรกเข้ามาในระบบ
- เกิดขึ้นได้หลายสาเหตุส่วนใหญ่เกิดจากปัญหาของระบบไฟฟ้า
 - Thermal Noise
 - Impulse Noise
 - Crosstalk
 - Echo
 - Jitter
 - Delay Distortion
 - Attenuation

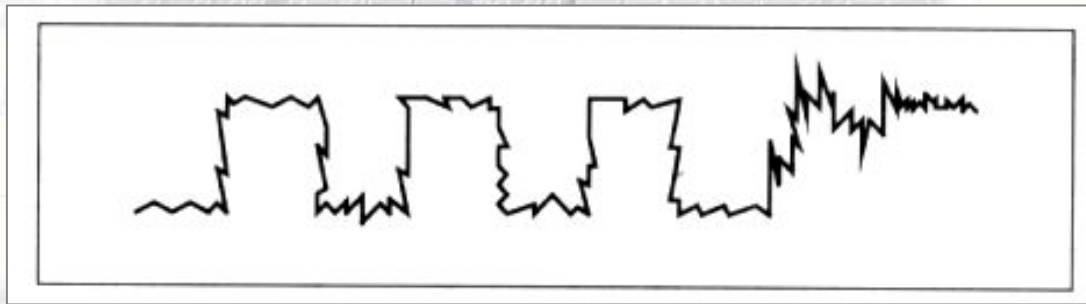
Thermal Noise

- เรียกอีกอย่างว่า **white Noise** ,
Gaussian Noise
- เกิดจากการเคลื่อนที่ของ อิเล็กตรอน
บนตัวนำ ซึ่งส่งผลให้เกิดความร้อนไป
รบกวนการเคลื่อนที่ ทำให้เกิดการ
เคลื่อนที่ ๆ ผิดปกติไป
- การป้องกัน
 - สัญญาณ Analog ใช้ Filters ช่วย
 - สัญญาณ Digital ใช้ Regenerate



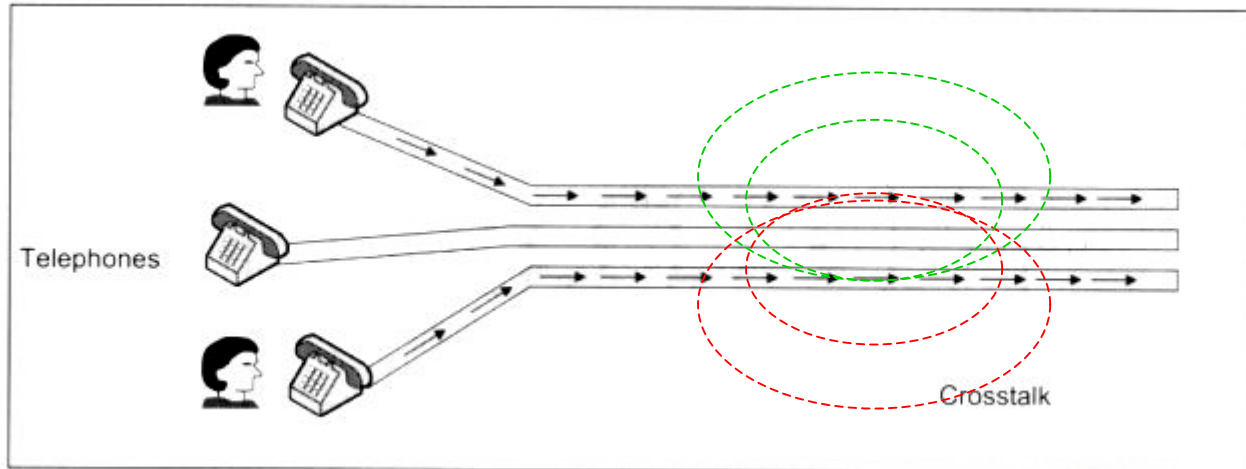
Impulse Noise

- เป็นเหตุการณ์ที่มีคลื่นสัญญาณโด่งขึ้นผิดปกติ(Spikes) อย่างรวดเร็ว และจะเกิดเป็นช่วงสั้นๆ
- มักเป็นสัญญาณที่มาจากปัจจัยภายนอก เช่น ไฟแลบ ไฟผ่า
- Noise ชนิดนี้ส่วนมากยากแก่การกู้คืน



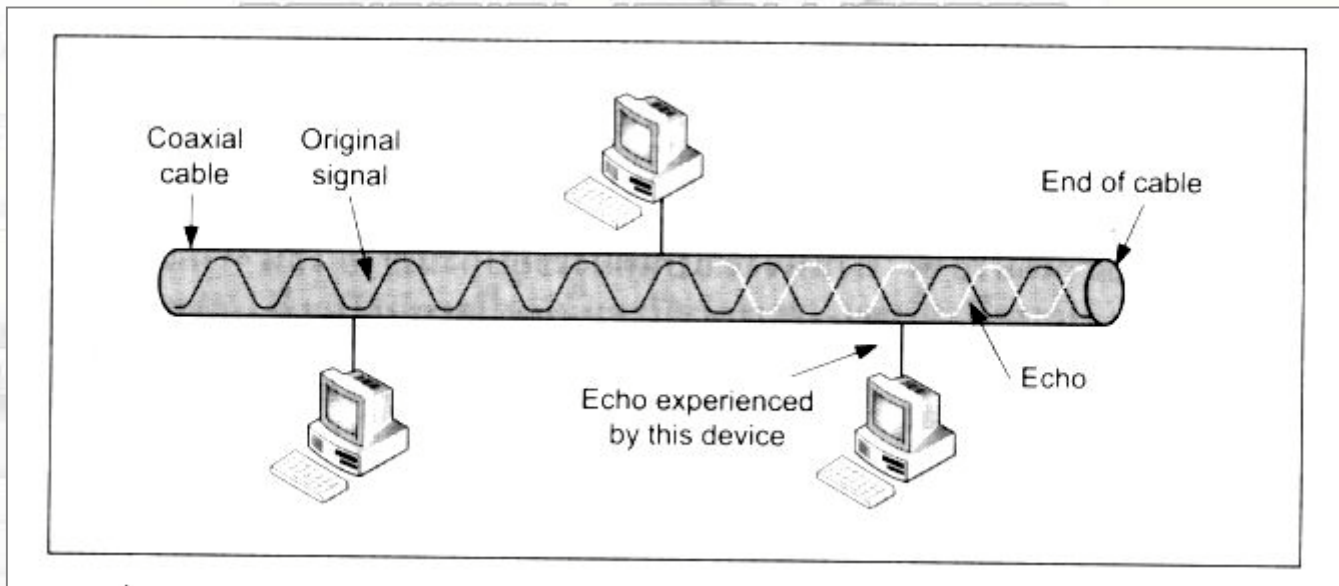
Crosstalk

- เกิดจากการเหนี่ยวนำของสนามแม่เหล็กไฟฟ้า
- เมื่ออิเล็กทรอนิกส์เคลื่อนที่ผ่านขดลวด จะทำให้เกิดสนามแม่เหล็ก
- ขดลวดเมื่อตัดผ่านสนามแม่เหล็กก็ทำให้เกิดกระแสไฟฟ้า
- สนามแม่เหล็กจากสายเส้นหนึ่งเข้าไปรบกวนสัญญาณในสายอีกเส้นหนึ่ง การแก้ไขคือต้องใช้สายที่มี Shield ป้องกัน



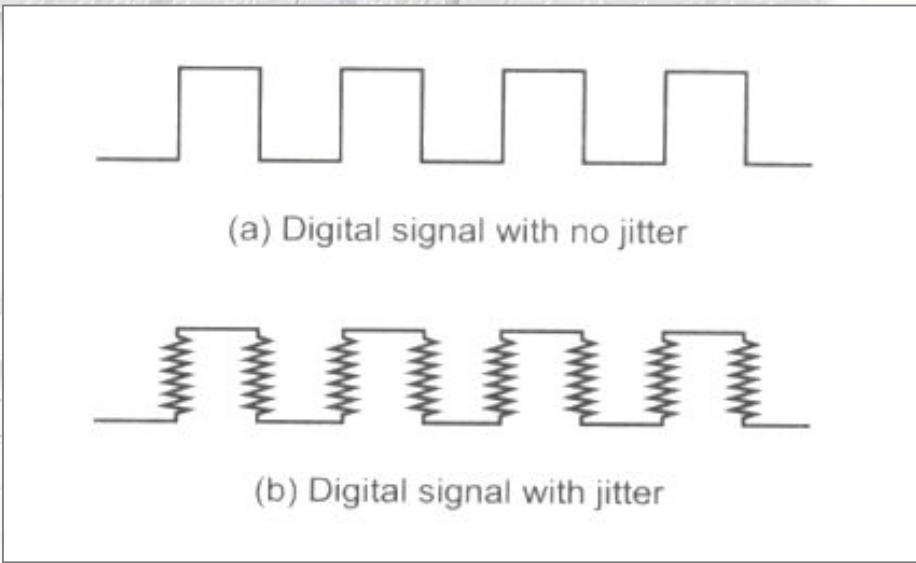
Echo

- หมายถึงสัญญาณสะท้อนกลับ(Reflection)
- ปัญหานี้มักเกิดกับสายชนิด Coaxial กล่าวคือเมื่อสัญญาณเดินทางไปสุดปลายสายจะเกิดการสะท้อนกลับตามธรรมชาติของสายชนิดนี้
- ทางแก้คือใช้ Terminator เพื่อดูดซับสัญญาณ



Jitter

- เป็นเหตุการณ์ที่ความถี่ของสัญญาณเกิดการเปลี่ยนแปลงตลอดเวลา ทำให้เฟสทางไฟฟ้าเปลี่ยนค่าจนตีความหมายผิด อาจเกิดจากปัญหาของอุปกรณ์อิเล็กทรอนิกส์เอง
- ทางแก้คือหาอุปกรณ์ที่มีคุณภาพ



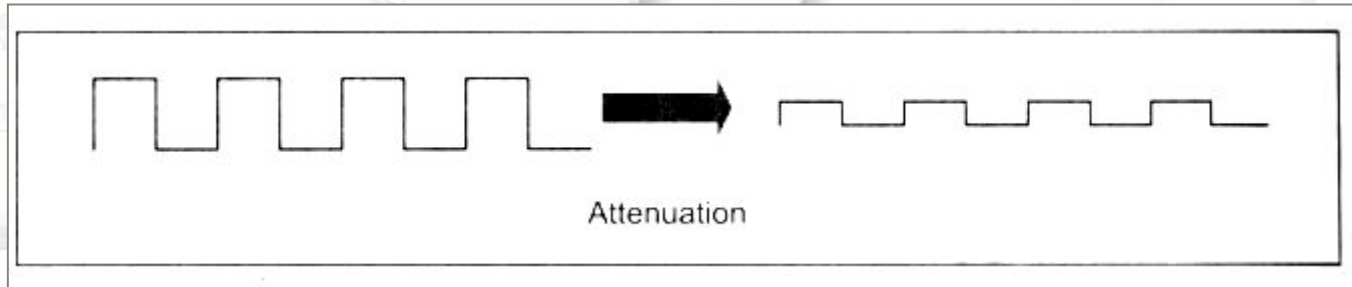
Delay Distortion

- สาเหตุจากการทำงานด้วยความเร็วที่แตกต่างกัน
- สัญญาณที่ส่งหลังเคลื่อนที่มาทันสัญญาณที่ส่งมาก่อนจนทับซ้อนกัน
- การป้องกันคือใช้วงจร Equalizer เพื่อควบคุมความถี่ของการทำงาน



Attenuation

- เกิดจากความต้านทานทางไฟฟ้าของสายสัญญาณ
- เมื่อสัญญาณเคลื่อนที่ออกไปในระยะที่ไกลขึ้น สัญญาณถูกลดทอนให้เขาลง หรือมีความเข้มลดลง จนไม่สามารถตีความได้
- ทางแก้คือใช้ Amplifier หรือ Repeater ช่วยทวนและขยายสัญญาณให้ชัดเจนขึ้นอีก



Error Detection การตรวจหาข้อผิดพลาด

- การเคลื่อนย้ายข้อมูลในแต่ละช่วงนั้น ปลายทางต้องทำการตรวจสอบว่าข้อมูลที่รับมานั้นถูกต้องหรือไม่
- ข้อผิดพลาดอาจไม่ได้มาจากปัญหาที่กล่าวมาแล้วแต่อาจเกิดจากอุปกรณ์ทำงานผิดพลาดเองก็ได้
- ชนิดของการตรวจสอบข้อผิดพลาด
 - Parity checks
 - Checksum
 - CRC : Cyclic Redundancy Checksum

Parity check

- ใช้วิธีเติมบิต 0 หรือ 1 แปะท้ายเข้าไป เพื่อบอกว่าผลรวมบิตเป็นเลขคู่หรือคี่
 - **Even Parity** กำหนด parity = 0 ถ้าผลรวมเป็น คู่ ถ้าไม่เป็น 1
 - **Odd Parity** กำหนด parity = 0 ถ้าผลรวมเป็น คี่ ถ้าไม่เป็น 1

Original data	Sender parity bit	Transmitted Information	Receiver calculated parity bit	Agree ?
0100110	1	01001101	1	Yes
0100110	1	01001001	0	No

Parity check ..

- บางครั้งหากเกิดข้อผิดพลาด 2 บิตพร้อมกัน ทำให้การตรวจสอบที่ปลายทางไม่สามารถทำได้เช่น

Original data	Sender parity bit	Transmitted Information	Receiver calculated parity bit	Agree ?
0100110	1	01001101	1	Yes
0100110	1	01000001	0	Yes

การหาผลรวม : Checksum

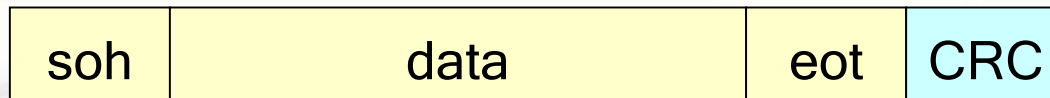
- ประสิทธิภาพในการตรวจหาข้อผิดพลาดดีกว่า Parity
- ฝ่ายส่งจะคำนวณหาผลรวม ASCII Code ของข้อมูล แล้วส่งไปพร้อมกับข้อมูล
- ฝ่ายรับจะทำแบบเดียวกันแล้วเทียบกับตัวเลขที่ฝ่ายส่งส่งมาให้
- เช่น ถ้าต้องการส่งคำว่า Networks ไปยังปลายทาง

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

- ผลรวมคือ $4E65+7477+6F72+6B73 = 19DC1$ ตัวเลขนี้จะถูกส่งไปกับข้อมูลเพื่อให้ฝ่ายรับตรวจสอบ

CRC : Cyclic Redundancy Checksum

- สองวิธีแรกจะไม่สามารถตรวจข้อผิดพลาดในกรณีเกิดการผิดพลาดตั้งแต่ 2 บิตขึ้นไป(Burst Error)
- CRC สามารถตรวจหาข้อผิดพลาดได้ 98.99999998%
- ใช้รหัสโพลีโนเมียล ซึ่งต้องมีบิตซ้ายและขวาสุดเป็น 1 และต้องมีจำนวนบิตน้อยกว่าบิตของข้อมูล ในลักษณะการคำนวณแบบ Modulo การคำนวณจะเป็นแบบ Exclusive-OR : xor



การคำนวณ

- กำหนดค่า

$M(x)$ คือ เฟรมข้อมูลที่ต้องการส่ง

$G(x)$ คือ โพลีโนเมียลที่ตั้งขึ้นมา

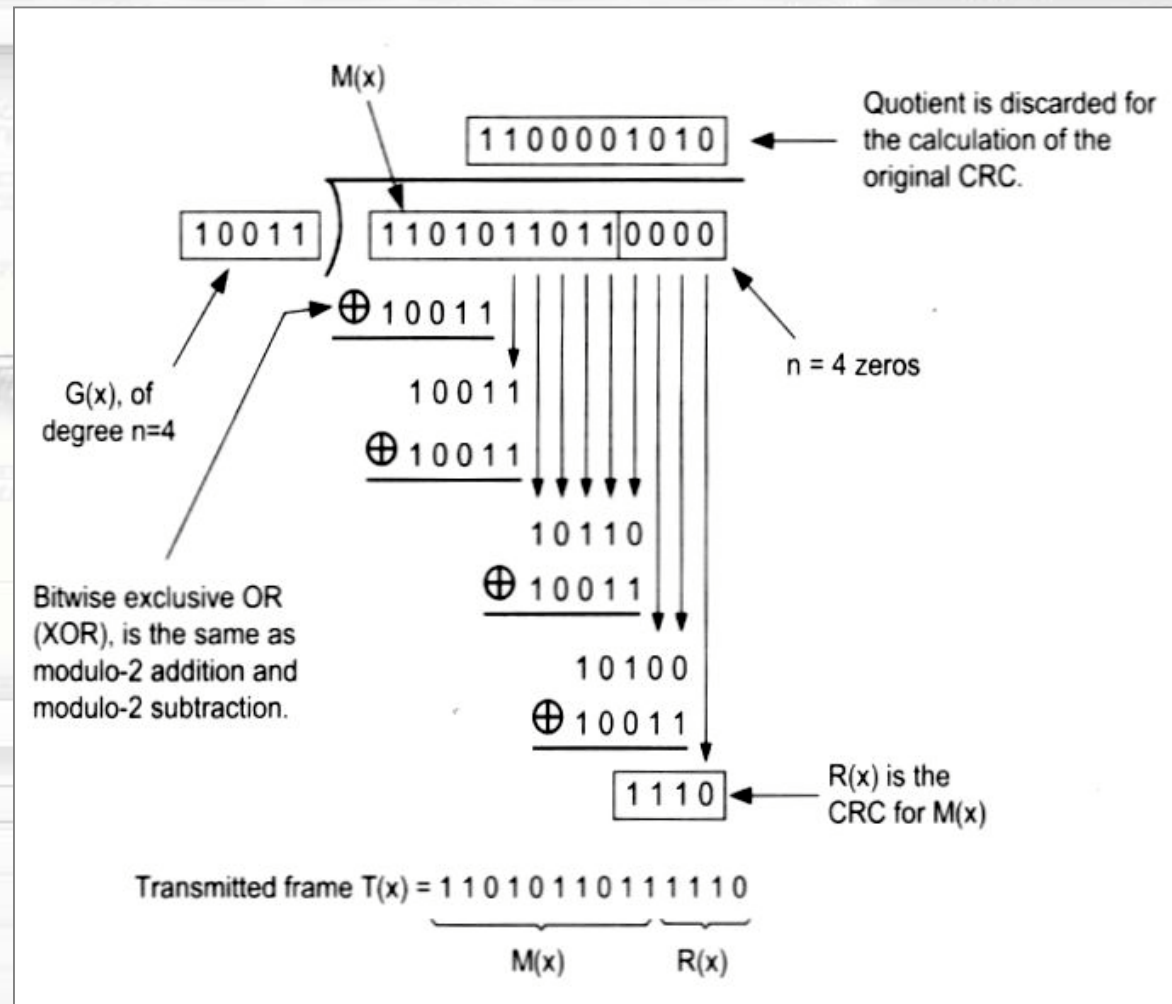
n คือ บิตศูนย์ที่เพิ่มเติมต่อท้าย $M(x)$

$R(x)$ คือ ผลของเศษที่ได้จากการคำนวณ

$T(x)$ คือ เฟรมที่ส่งออกไปมีค่าเท่ากับ $M(x)+R(x)$

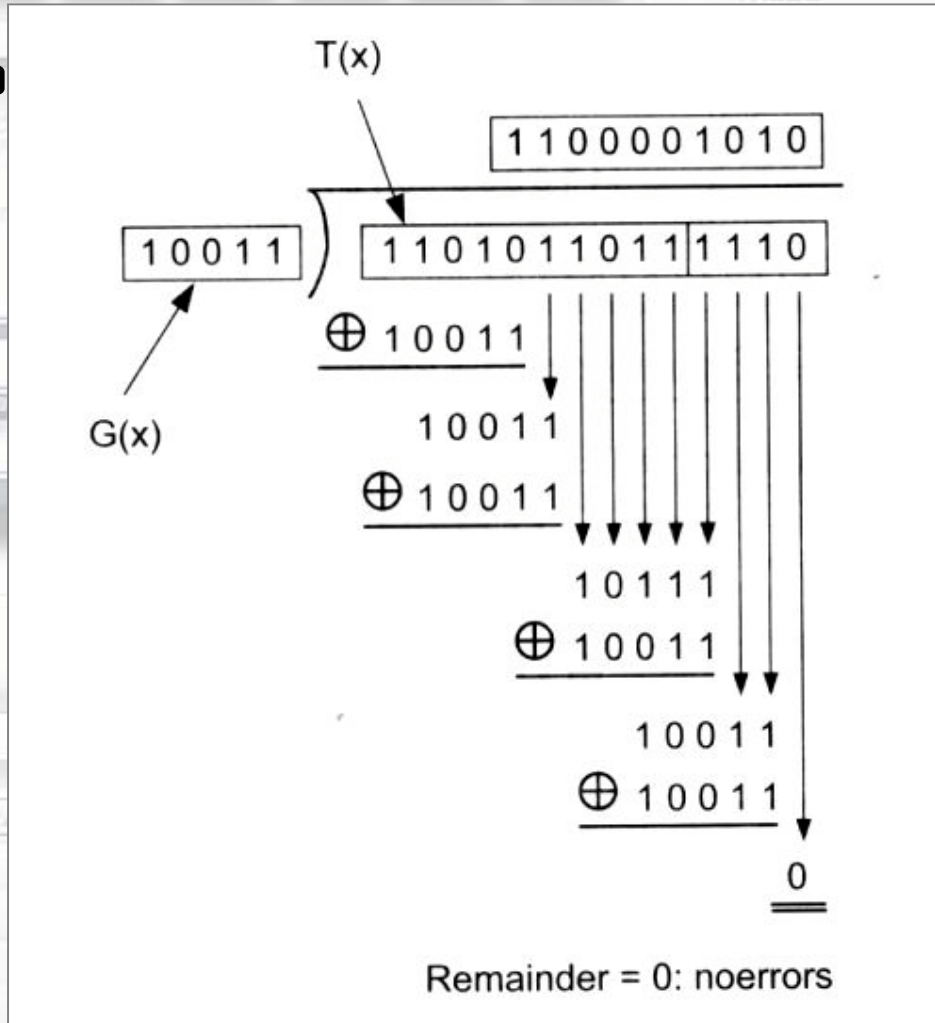
CRC ด้านส่ง

• Data = 1101011011



CRC ด้านรับ

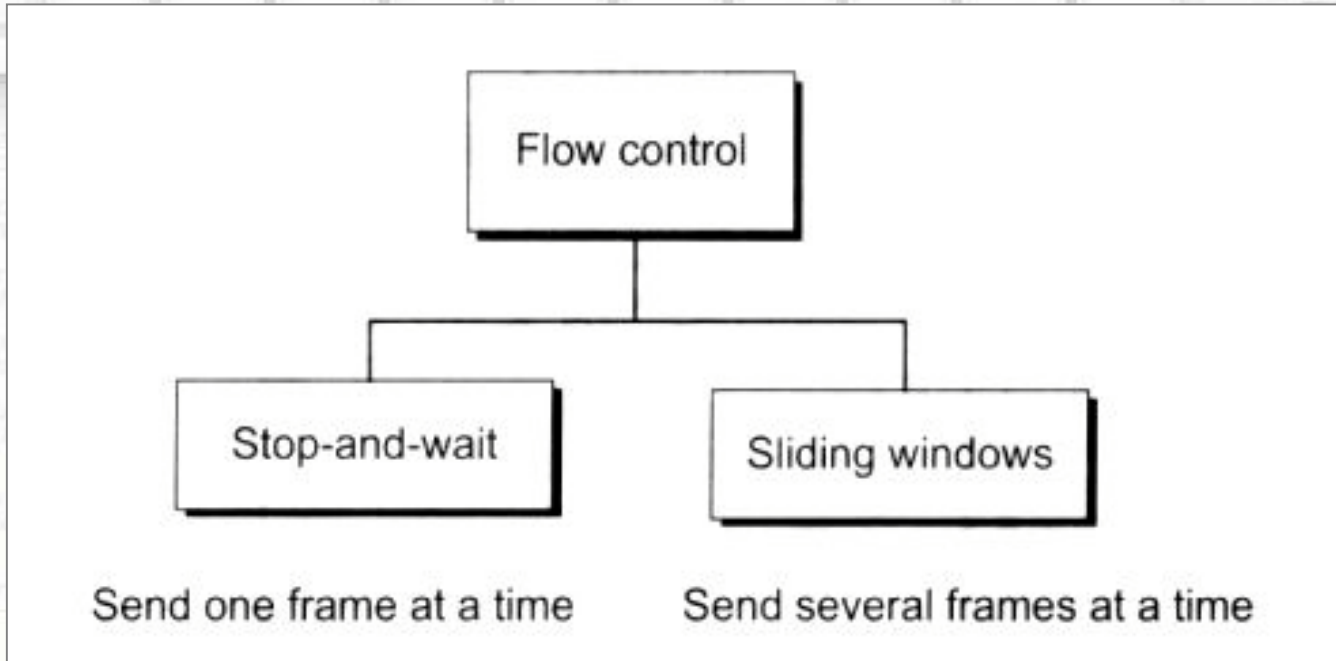
- ปลายทางจะนำ $T(x)$ ไปหารด้วย $G(x)$ ถ้าหารลงตัว แสดงว่าข้อมูลถูกต้อง



Flow Control and Error Control

- วิธีการที่จะบอกฝ่ายส่งว่าจะสามารถส่งข้อมูลจำนวนเท่าใดก่อนที่จะได้รับการรับรอง(Acknowledgment) จากฝ่ายรับ
- เป็นวิธีการที่จะบอกว่าเมื่อใดควรส่ง เมื่อใดควรรอ เมื่อใดควรส่งใหม่
- Buffer เป็นพื้นที่สำหรับจัดเก็บข้อมูลที่เคลื่อนเข้ามาไว้ชั่วคราว
- การกำหนด Flow Control
 - กำหนดแผนการส่งเฟรมต่างๆ ที่ต้องการส่งละทำการติดตาม
 - ส่งเฟรมเมื่อใด
 - เฟรมที่จะส่งจะจบหรือสิ้นสุดเมื่อใด

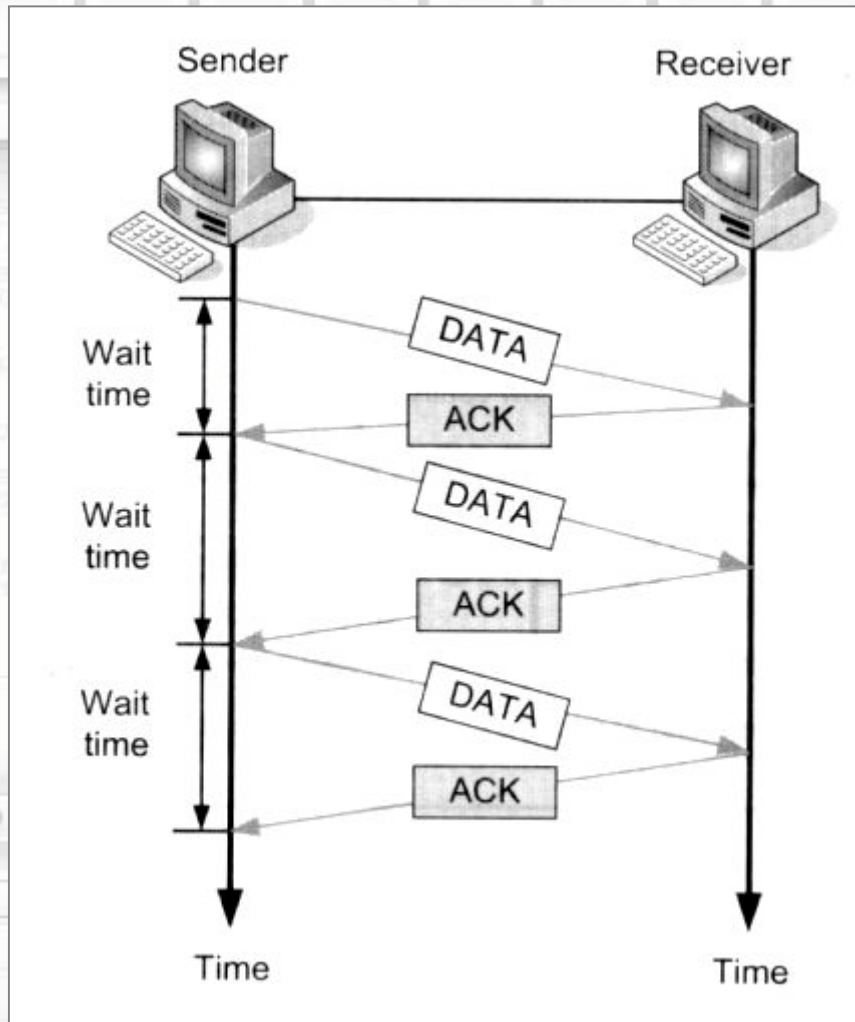
Flow control



Type of Flow Control

- Stop-and-Wait Flow control หรือหยุดและรอ
 - ง่ายส่งส่งเฟรมข้อมูลไปให้ฝ่ายรับหนึ่งเฟรม
 - รอการตอบรับ(Acknowledge:ACK)จากฝ่ายรับ
 - เมื่อฝ่ายส่งได้รับ ACK ก็จะทำการส่งเฟรมต่อไป
- ข้อดี ง่ายต่อการประยุกต์ใช้งาน
- ข้อเสีย ใช้เวลาในการรอคอยมากพอๆ กับเวลาที่ข้อมูลเดินทาง

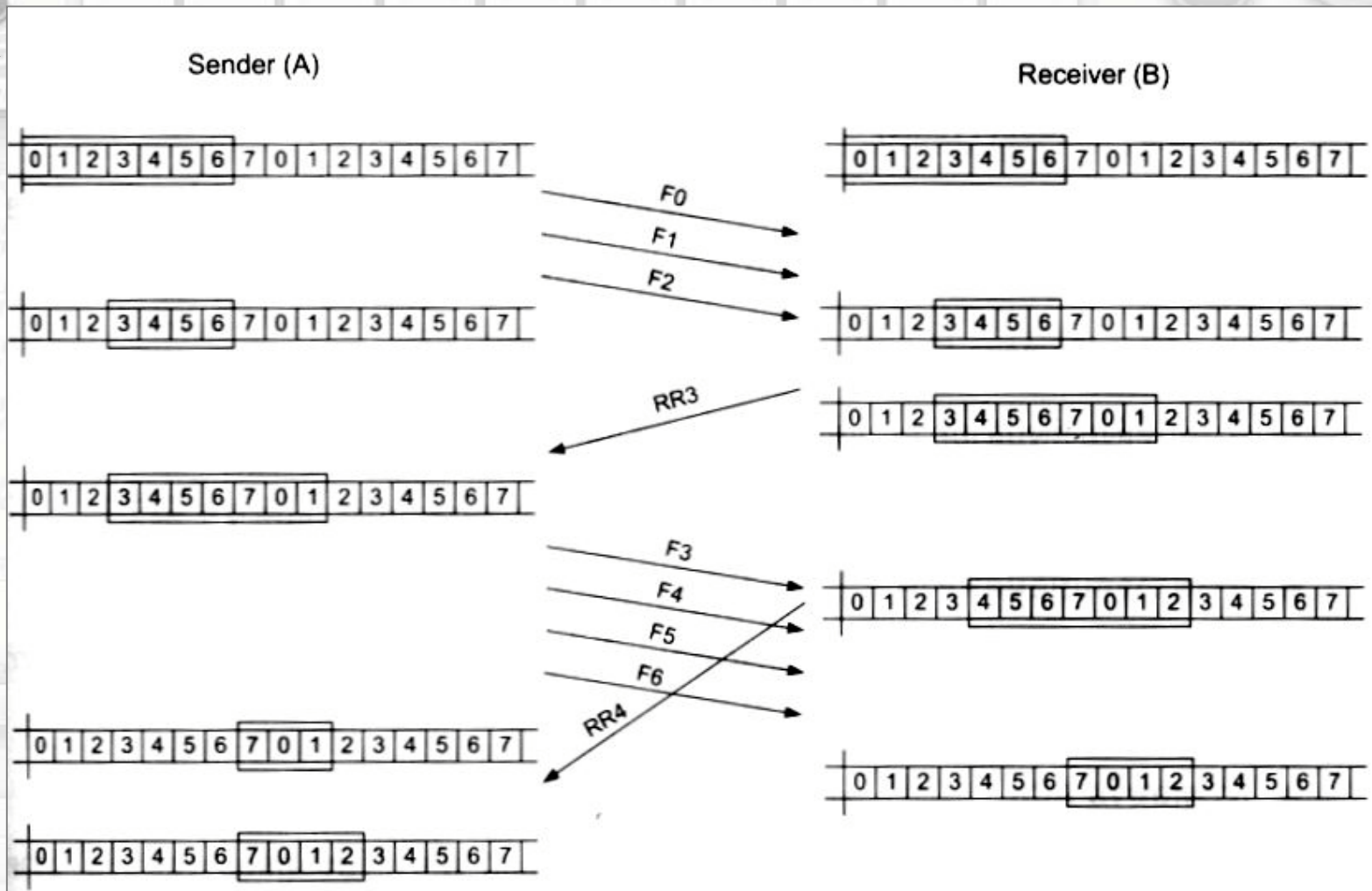
Stop-and-Wait



Type of Flow Control

- Sliding-Window Flow Control โดยการกำหนดช่องหน้าต่าง (Window) โดยฝ่ายส่งจะส่งข้อมูลออกไปทีละหลายๆ เฟรม
- ฝ่ายรับจะตอบกลับการรับมาเพียงบางเฟรมเท่านั้น
- ข้อดี ลดเวลาในการรอคอย ACK จากฝ่ายรับ
- ข้อเสีย อาจต้องส่งเฟรมใหม่ทั้งหมด

Sliding-Window



Error Control

- ทำไมต้องทำ Error Control
 - ต้องตรวจสอบข้อผิดพลาดของเฟรมอย่างไร และต้องทำอะไรบ้าง
 - เพื่อแสดงว่าต้องส่งอย่างไรให้ถึงปลายทางโดยปราศจากข้อผิดพลาด



เมื่อมี Error

- ไม่ต้องทำอะไร(Do Nothing)
- ตอบรับ Message ไปยังฝ่ายส่งเพื่อให้ฝ่ายส่งส่งข้อมูลที่เสียหายมาใหม่
- ฝ่ายรับแก้ไขข้อผิดพลาด(Correct the Error)

ARTIFICIAL INTELLIGENCE

Type of Error

- Lost Frame หรือเฟรมสูญหาย
- Damage Frame หรือเฟรมเสียหายแต่เดินทางมาถึงปลายทาง
- Timeout คือเฟรมใช้เวลาในการเดินทางมากเกินไป
- NAK (Negative Acknowledgment) จะถูกส่งเพื่อบอกว่าเฟรมที่ได้รับมีข้อผิดพลาด
- Go-back-n ใช้บอกว่าต้องส่งใหม่ตั้งแต่เฟรมที่ทำได้

End.

ARTIFICIAL INTELLIGENCE

