



Data Link Control and Protocols

Paramate Horkaew

School of Computer Engineering
Institute of Engineering, Suranaree University of Technology



Lecture Outline

- **Flow and Error Control**
- **Stop-and-Wait ARQ**
- **Go-Back-N ARQ**
 - Sender/Receiver Sliding Window
 - Control Variables
 - Timers
 - Acknowledgement, Resending Frame
 - Operation
 - Sender Window Size and Bi-Dir Tx and Piggybacking
- **Selective Repeat ARQ**
- **HDLC**
 - Configuration and Mechanism
 - Formats and Examples

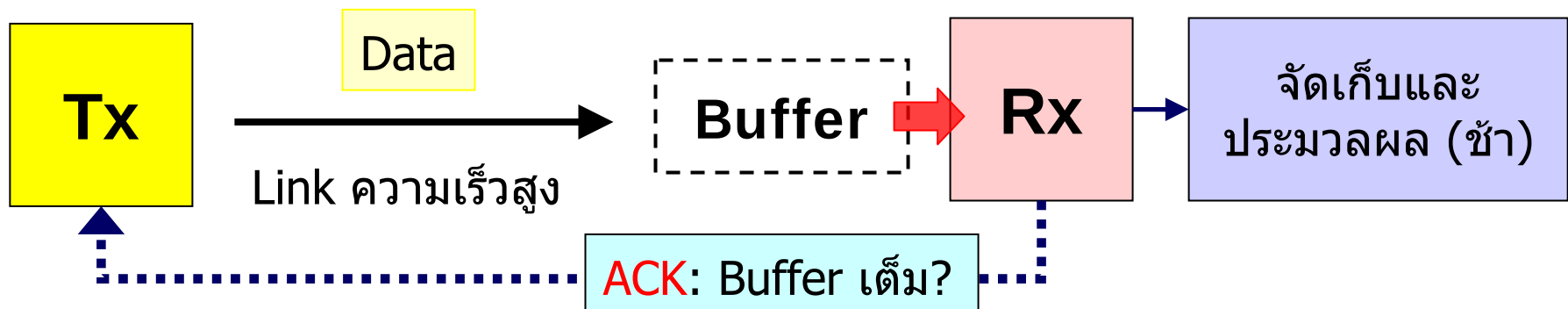
Flow and Error Control (I)

- Flow Control

คือการบริหารจัดการ ปริมาณข้อมูล ที่สามารถส่งได้ก่อนได้รับการตอบรับจากผู้รับ (Acknowledgement)

ความสำคัญ

อุปกรณ์ด้านรับ มีขีดความสามารถในการประมวลผล และจัดเก็บข้อมูลที่ได้รับมาจำกัด ผู้รับจึงต้องแจ้งผู้ส่งให้ทราบ ก่อน ที่ปริมาณข้อมูลจะเข้าสู่พิกัดดังกล่าว ซึ่งเกิดเหตุการณ์ได้ 2 กรณี 1) ร้องขอให้ผู้ส่งส่งข้อมูลขนาดเล็กลง หรือ 2) ให้ผู้ส่งหยุดทำการส่งชั่วคราว ดังแผนผัง



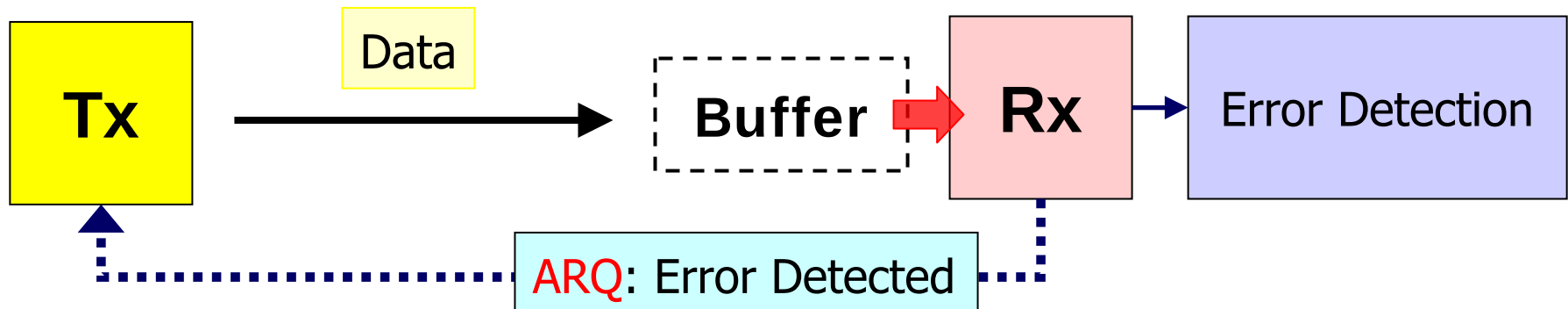
Flow and Error Control (II)

- Error Control

คือกระบวนการ ตรวจจับ และแก้ไขข้อผิดพลาดในข้อมูล (Error Detection and Correction) สำหรับ Data Link Layer มักจะหมายถึง Error Detection ร่วมกับการร้องขอ ให้มีการส่งใหม่ (Automatic Repeat Request: **ARQ**)

ความสำคัญ

เมื่อตรวจพบข้อมูลผิดพลาด อันเนื่องมาจาก ข้อมูลสูญหาย หรือเสียหายจากสัญญาณรบกวน ระหว่างการส่ง สำหรับ Data Link Layer อุปกรณ์ด้านรับมักจะร้องขอให้อุปกรณ์ด้านส่ง ส่งข้อมูลมาใหม่ (ARQ) แทนที่จะแก้ไขข้อผิดพลาดเอง



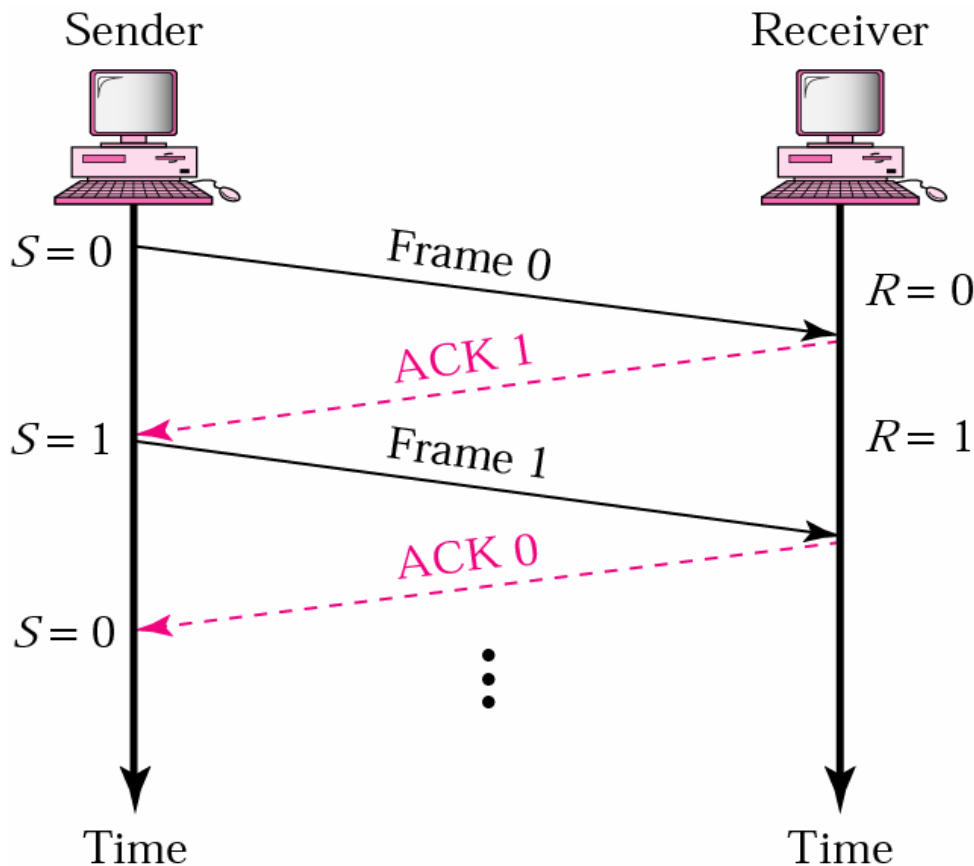


Stop-and-Wait ARQ

- **Sender Makes a Copy** อุปกรณ์ด้านส่งทำสำเนาของข้อมูลไว้ก่อนทำการส่งจนกว่าจะได้รับการตอบรับ
- **Data and ACK Frames are Numbered** เพื่อประโยชน์ในการระบุ ลำดับของข้อมูล จะมีการกำหนดหมายเลขประจำ (ID) ให้กับทั้งส่วนที่เป็นข้อมูล (Data) และส่วนที่เป็นการตอบรับ (ACK)
- **Discard Lost/Damage Frames** ถ้าด้านรับตรวจพบข้อผิดพลาด หรือข้อมูลเรียงลำดับไม่ถูกต้องจะทิ้งข้อมูลนั้น
- **Control Variables** ด้านส่ง และ รับ เก็บตัวแปรควบคุมชื่อ S (หมายเลขข้อมูลที่เพิ่งส่ง) และ R (หมายเลขข้อมูลที่คาดว่าจะได้รับ) ตามลำดับ
- **Timing** ด้านส่งเริ่มจับเวลาเมื่อส่งข้อมูลเสร็จ ถ้าไม่ได้รับ ACK ภายในเวลาที่กำหนด จะส่งซ้ำ (ถือว่าข้อมูลสูญหาย)
- **Positive ACK** ถ้าด้านรับได้รับข้อมูลครบถ้วนสมบูรณ์จะส่ง ACK ไปยังผู้ส่ง พร้อมกับระบุหมายเลข Frame ถัดไปที่ คาดว่าจะได้รับ

Operations

การทำงานของ Stop-and-Wait ARQ แบ่งออกได้เป็น 4 เงื่อนไข



- **Normal Operation**

เมื่อผู้ส่งส่ง Frame 0 จะหยุดเพื่อรอ ACK 1 จากผู้รับ

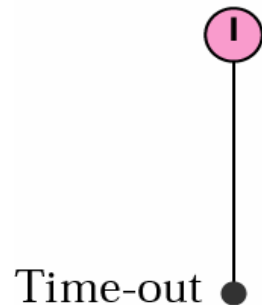
เมื่อผู้ส่งได้รับ ACK 1 จะส่ง Frame 1 ต่อไป แล้วหยุดรอ ACK 0 จากผู้รับ

โดย R เก็บหมายเลข Frame ถัดไปที่ผู้รับคาดหวัง

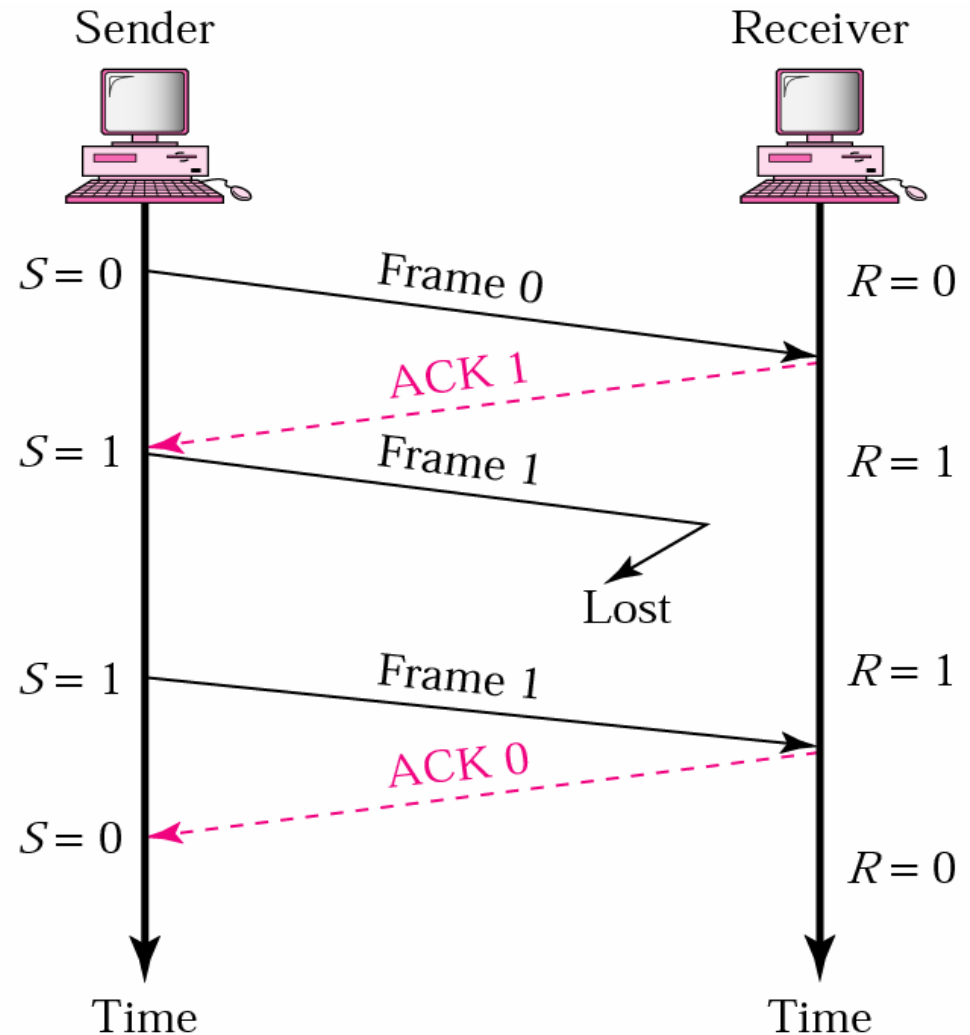


Lost or Damaged Frame

เมื่อเกิดกรณีข้อมูลหาย หรือ
ผิดพลาด ผู้รับจะนิ่งเฉย (ไม่ส่ง
สัญญาณใดๆ กลับมา) พร้อมกับ
เก็บค่า R ที่คาดหวังไว้ (ในรูปคือ
 $R = 1$)

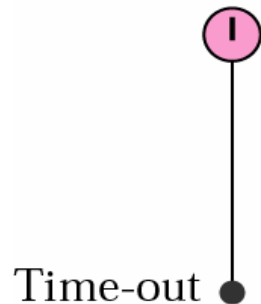


เมื่อเวลา (รอ ACK) ทางด้านส่ง
หมดลง ผู้ส่งจะส่ง Frame ที่สูญ
หายไป หรือผิดพลาดใหม่ทันที

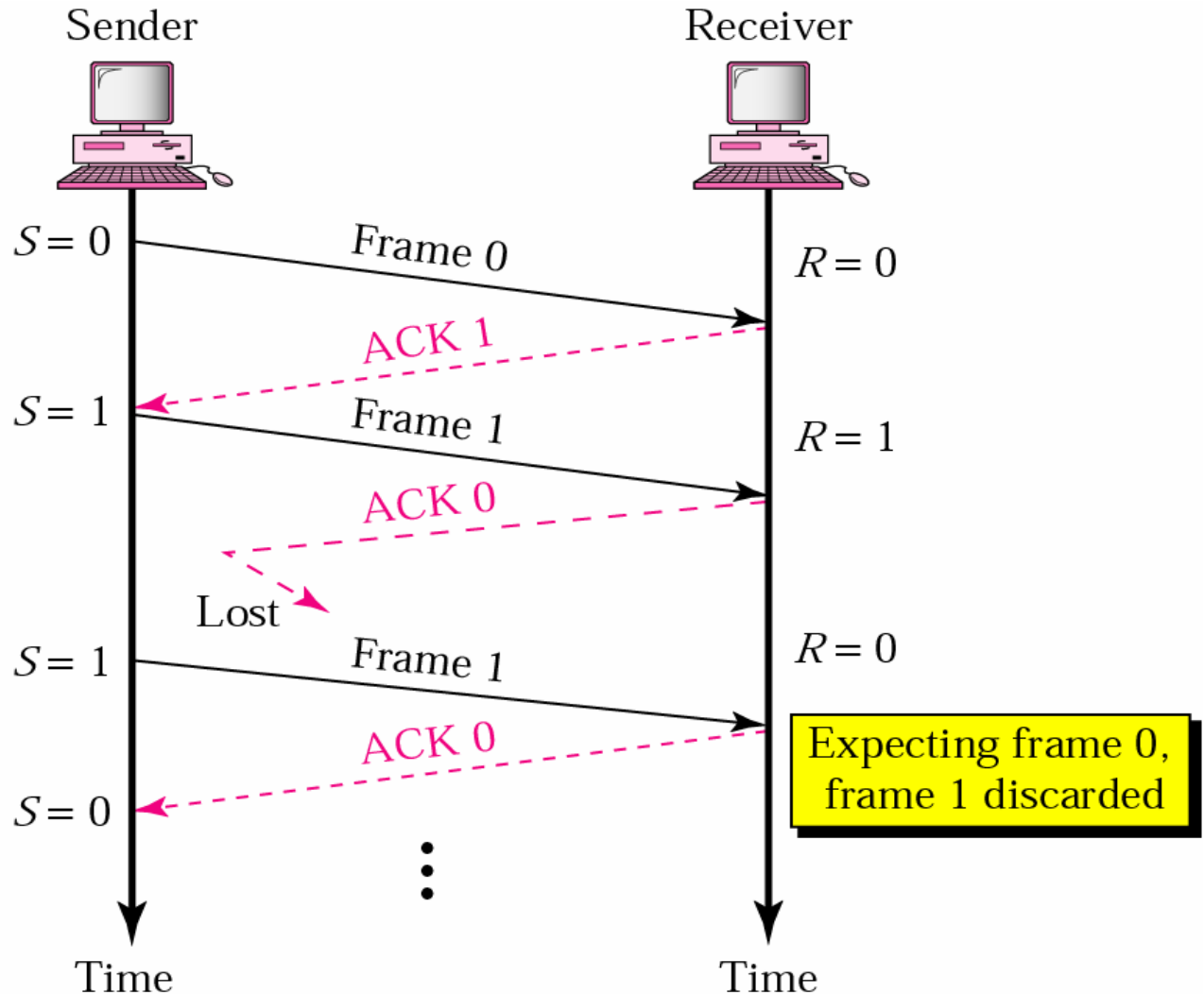


Lost Acknowledgement

เมื่อเกิดกรณี ACK สูญ
หาย เมื่อรอจนหมด
เวลาผู้ส่งจะส่ง Frame
มาใหม่

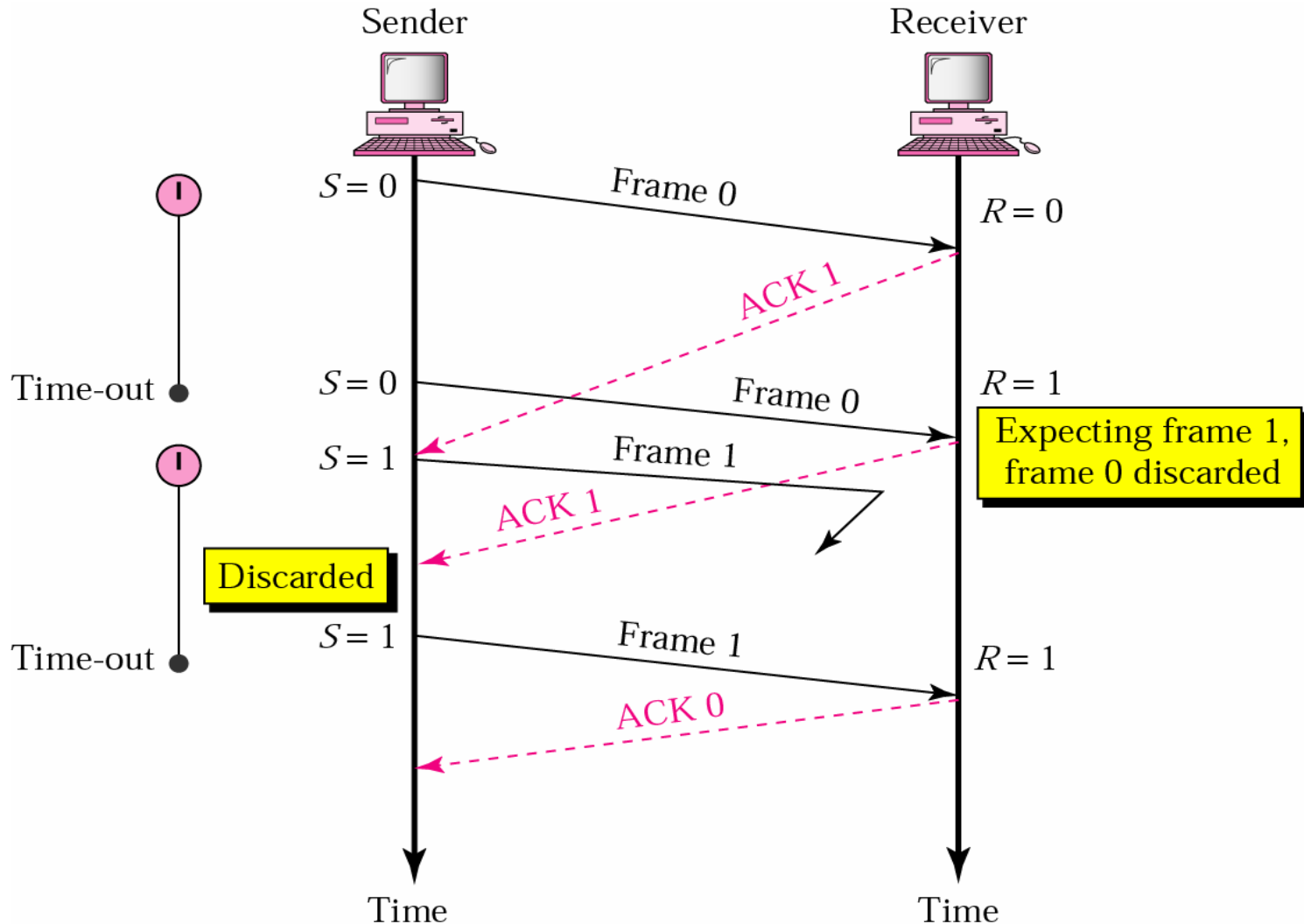


ถ้าผู้รับได้รับ Frame
ซ้ำ จะลบ Frame นั้น
ทิ้ง



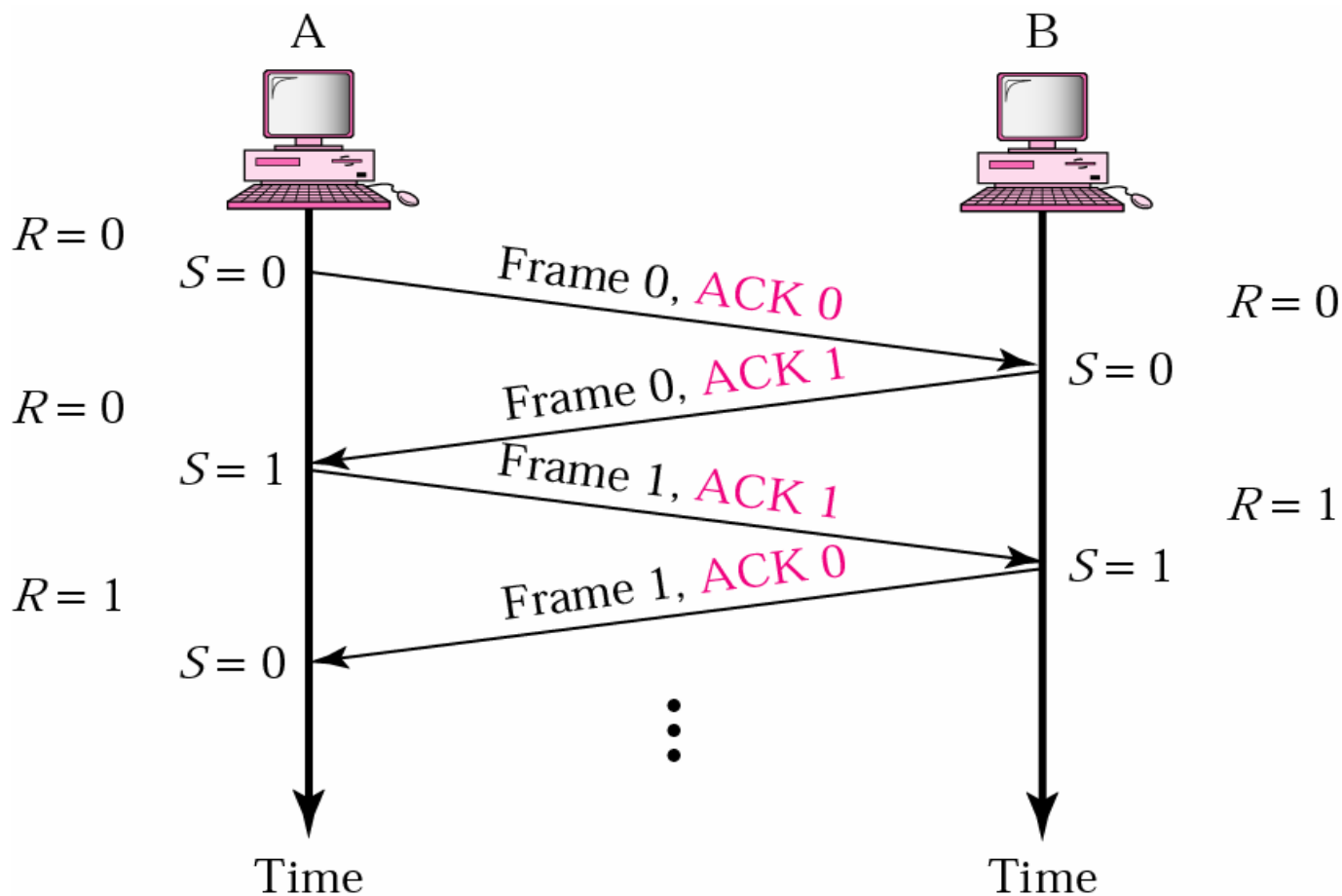


Delayed Acknowledgement



Piggybacking

ในการสื่อสารแบบ Full-Duplex แต่ละฝั่งสามารถส่ง Frame พร้อมกับ ACK ได้





Go-Back-N ARQ

สำหรับวิธี Stop-and-Wait ARQ ในช่วงเวลาหนึ่งๆ จะมีข้อมูลเพียง 1 Frame เท่านั้นที่ใช้ทรัพยากรของตัวกลาง ทั้งนี้เพราะ ผู้ส่งจะต้องรอ ACK ก่อน ถึงจะส่งข้อมูลใหม่ได้ ทำให้มีประสิทธิภาพต่ำ

เพื่อให้ใช้ตัวกลาง (Link) ได้เต็มประสิทธิภาพ จึงต้องส่งข้อมูลมากกว่า 1 Frame ไปในตัวกลางพร้อมๆ กัน ซึ่งทำได้ 2 วิธี ได้แก่

- **Go-Back-N ARQ**

คือกระบวนการทำ ARQ ซึ่งสามารถส่งข้อมูลได้พร้อมกันหลาย Frame ก่อนที่จะประมวลผล ACK ซึ่งหากเกิดข้อผิดพลาดขึ้น จะส่งใหม่ทั้งกลุ่ม Frame

- **Selective Repeat ARQ**

คือกระบวนการทำ ARQ ซึ่งสามารถส่งข้อมูลได้พร้อมกันหลาย Frame ก่อนที่จะประมวลผล ACK ซึ่งหากเกิดข้อผิดพลาดขึ้น จะส่งเฉพาะ Frame ที่ผิดพลาด



Sequence Number

ผู้ส่งต้องกำหนดหมายเลข Data Frame ตามลำดับ ซึ่งต้องมีการกำหนดค่าสูงสุดของลำดับ (เพื่อประหยัดเนื้อที่ใน Header ของ Data Frame)

กำหนดหมายเลข Frame ใช้ $m = 2$ บิต จะมีพิสัยจาก 0 ถึง $2^2 - 1$ 0, 1, 2, 3, 0, 1, 2, 3, 0, 1, ...

กำหนดหมายเลข Frame ใช้ $m = 3$ บิต จะมีพิสัยจาก 0 ถึง $2^3 - 1$ 0, 1, 2, 3, 4, 5, 6, 7, 0, 1, ...

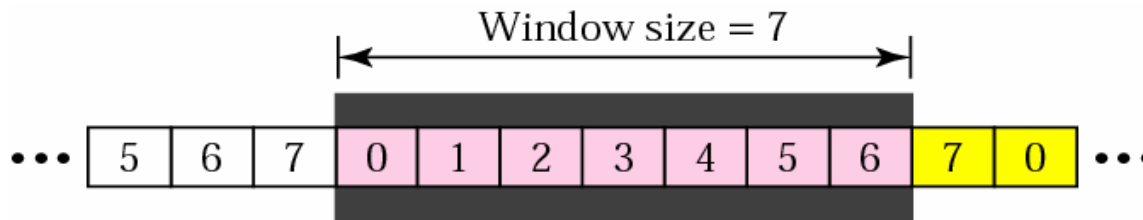
กำหนดหมายเลข Frame ใช้ $m = 4$ บิต จะมีพิสัยจาก 0 ถึง $2^4 - 1$ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 0, 1, ...

เมื่อหมายเลข Data Frame เพิ่มขึ้นครั้งละ 1 มีค่าสูงสุด อนุกรมของหมายเลขจะวนซ้ำกลับมาเริ่มที่หมายเลข 0 ใหม่

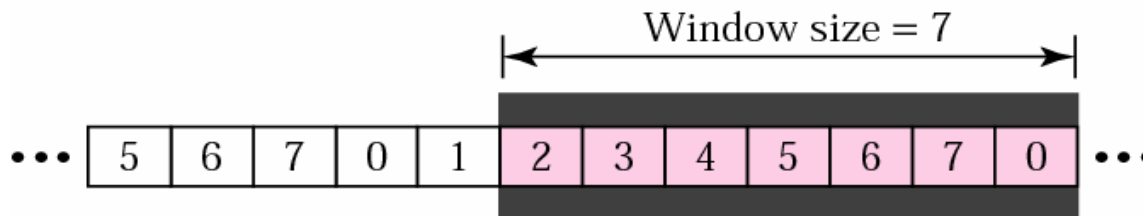


Sender Sliding Window

การกักเก็บ Data Frame ไว้ที่ด้านส่ง ใช้หลักการของการเลื่อนหน้าต่าง โดยคิด
เสมือนว่า Frame จำนวน 1 ชุด ($2^m - 1 = 7$ Frame) ถูกเก็บไว้ในหน่วยความจำ
Buffer (แสดงด้วย Window ในรูป)



a. Before sliding



b. After sliding two frames

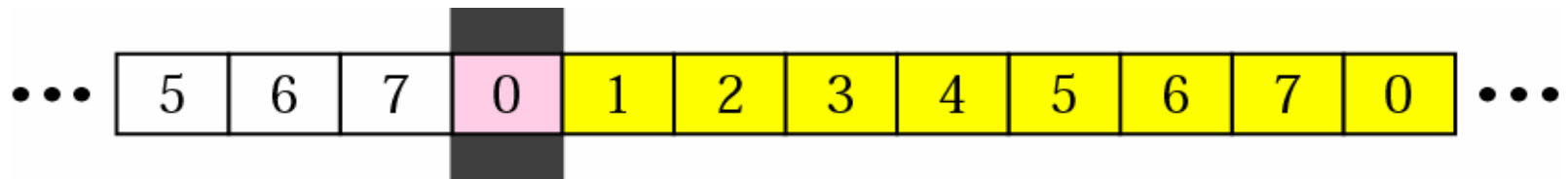
รูป b Window เลื่อนไปทางขวาเมื่อ Frame 0, 1 ได้รับ ACK

Frame ด้านซ้าย ของ Window คือ Frame ที่ได้รับ ACK แล้วซึ่งสามารถลบทิ้งได้

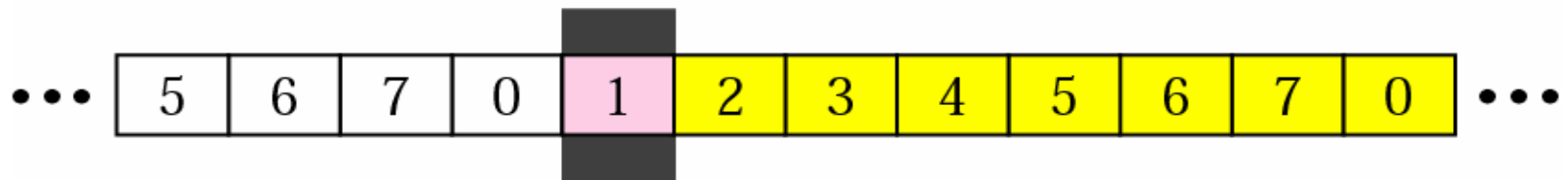
Frame ด้านขวา ของ Window คือ Frame ที่ยังส่งไม่ได้จนกว่า Frame ใน Window จะได้รับ ACK

Receiver Sliding Window

ใน Protocol นี้ ขนาดของหน้าต่างด้านรับ จะกำหนดให้เป็น 1 เสมอ หรืออีกนัยหนึ่ง ด้านรับจะเฝ้ารอ Frame เฉพาะที่จะมาถึง หากได้รับ Frame อื่นที่ไม่ต้องการ Frame นั้นจะถูกลบทิ้ง และจะต้องส่งใหม่



a. Before sliding



b. After sliding

รูป a ด้านรับรอ Frame หมายเลข 0 ซึ่งเมื่อมาถึง (รูป b) Window จะเลื่อนไป Frame 1



Control Variables

★ ด้านส่ง มีตัวแปรควบคุมจำนวน 3 ตัว ได้แก่

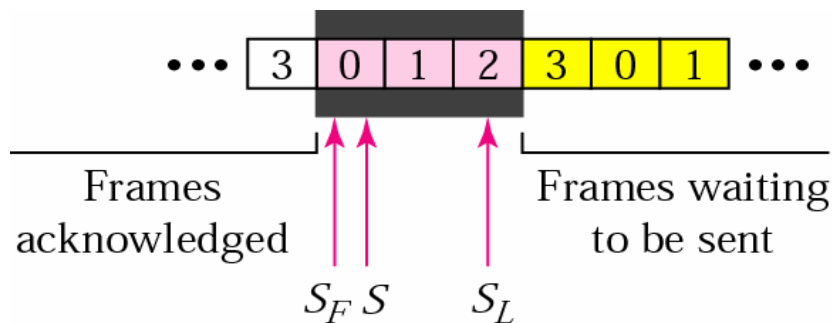
S เก็บลำดับตัวเลขของ Frame ที่เพิ่งส่งออกไป

S_F เก็บลำดับตัวเลขของ Frame แรกใน Window (**First**)

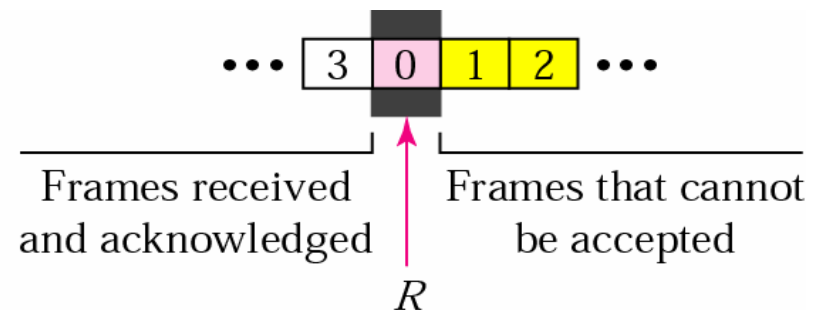
S_L เก็บลำดับตัวเลขของ Frame สุดท้ายใน Window (**Last**)

ดังนั้น ขนาดของ Window หาได้จาก $(S_F - S_L + 1)$ Frame

★ ด้านรับ มีตัวแปรควบคุม 1 ตัว ได้แก่ R ซึ่งเก็บ Frame ที่คาดหวังจะได้รับ



a. Sender window



b. Receiver window



Timers, ACK and Resending

Timer

- ด้านส่ง กำหนดตัวจับเวลาเพื่อรอรับ ACK แต่ ด้านรับ ไม่มีตัวจับเวลา

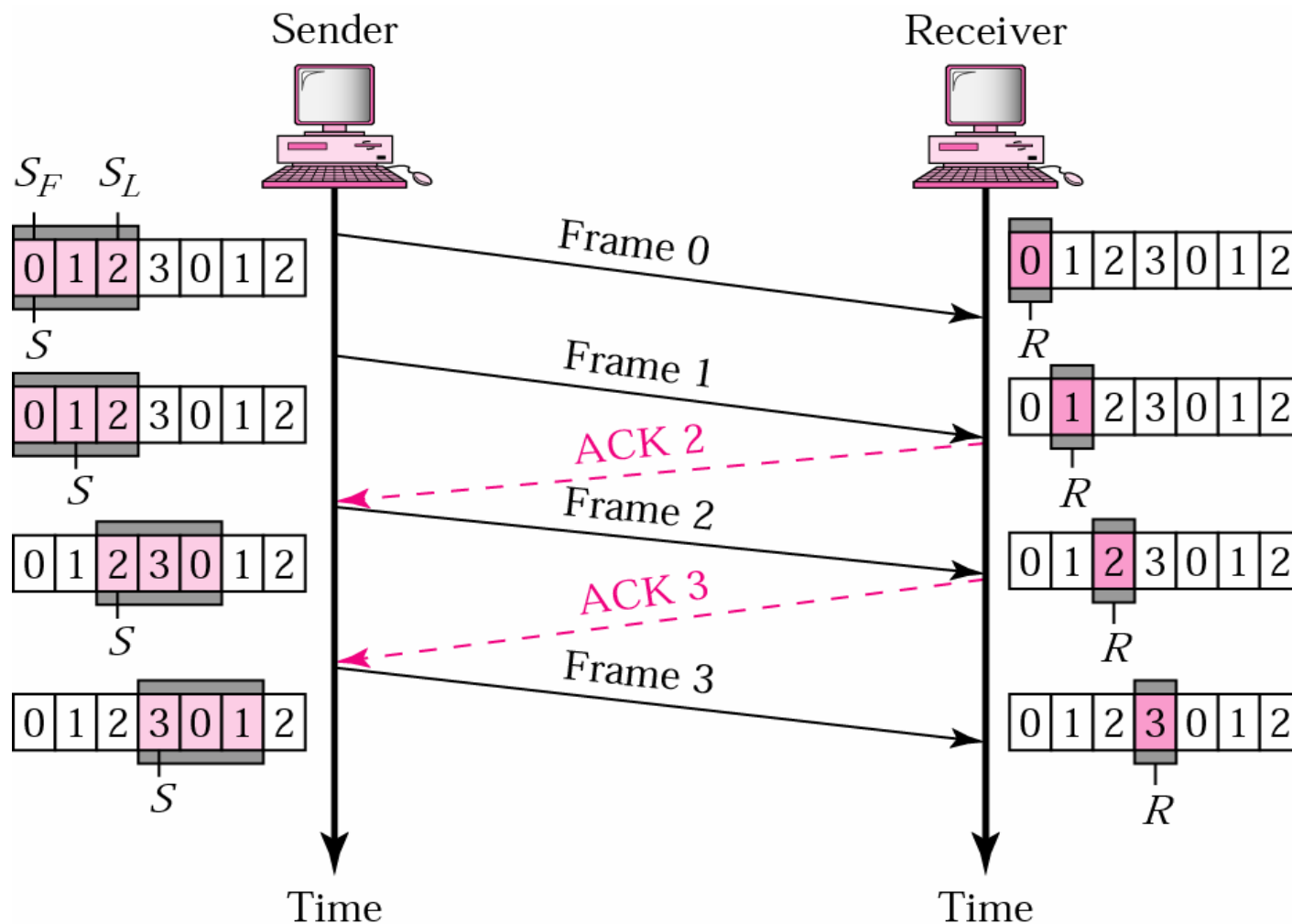
Acknowledgement

- ด้านรับ ส่ง (Positive) ACK เมื่อได้รับ Frame ที่คาดหวังโดยสมบูรณ์ (อาจจะส่ง ACK 1 หรือหลาย Frame ต่อครั้งก็ได้) ถ้า Frame ที่คาดหวังสูญหายหรือผิดพลาด ด้านรับจะนิ่งเฉย และลบ Frame อื่นที่ทะยอยส่งเข้ามาทิ้งจนกว่าจะได้รับ Frame ที่ต้องการ

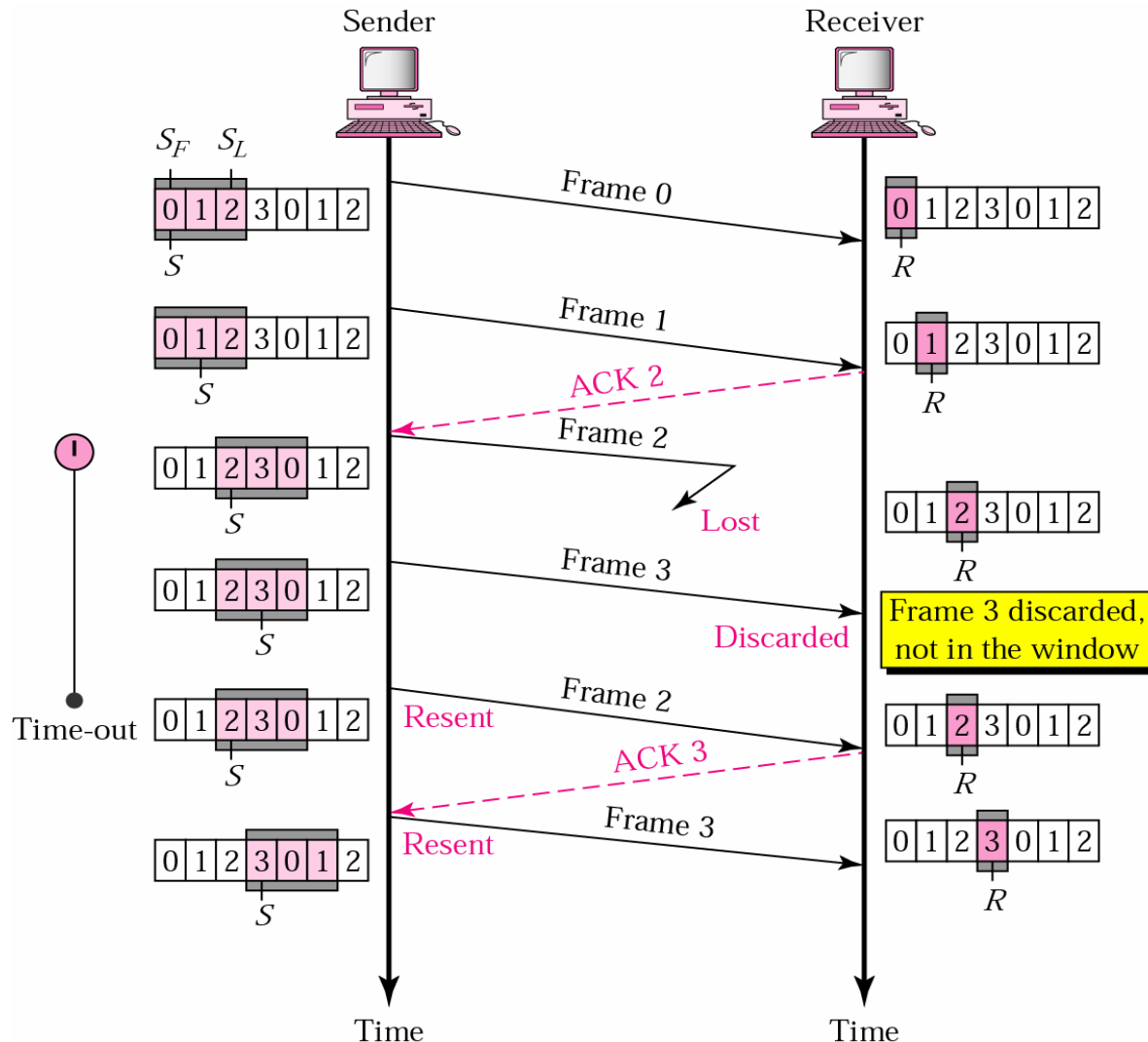
Resending Frame

- การนิ่งเฉยของด้านรับทำให้ Timer ด้านส่งหมดเวลา ซึ่งทำให้ด้านส่ง ทำการส่ง Frame ตั้งแต่ Frame แรกที่ Timer หมดเวลาจนกระทั่งถึง Frame สุดท้ายใน Window ขณะนั้น เช่นถ้า Timer ของ Frame 3 หมดลงแต่ได้ส่งไปถึง Frame ที่ 6 จะทำการส่ง Frame 3, 4, 5, และ 6 ใหม่

Normal Operation



Damaged or Lost Frame



ในกรณี Data Frame
เกิดสูญหาย

เช่น Frame 2 ไปไม่ถึง
ถึงด้านรับ ระหว่างนั้น
ด้านส่งได้ทำการส่ง
Frame 3 ไปด้านรับ
แต่ถูกลบทิ้ง เนื่องจาก
เป็น Frame ที่ด้านรับ
ไม่คาดหวัง

เมื่อเวลารอ ACK ของ
Frame 2 หมดด้านส่ง
จึงทำการส่ง Frame 2
ใหม่ รวมถึง Frame 3
ด้วย



Problems with ACK

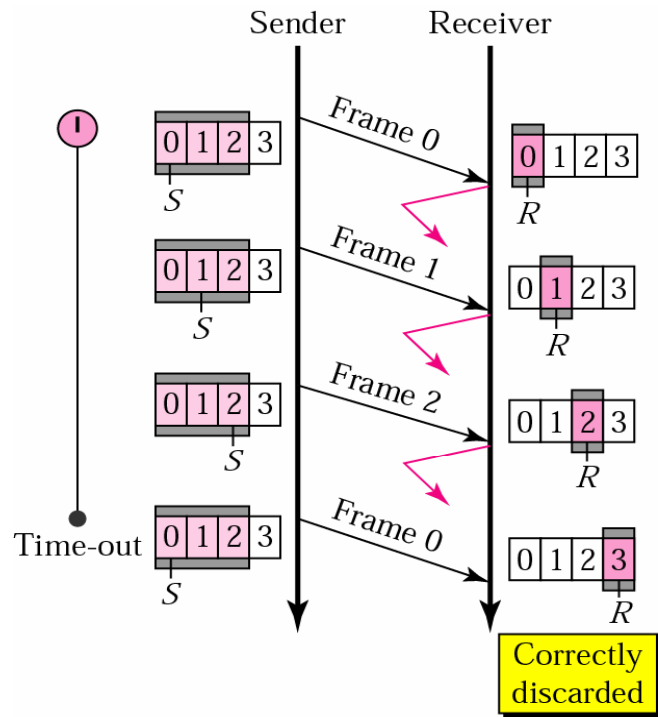
Damaged or Lost ACK

- เมื่อสัญญาณ ACK สูญหาย (ไปไม่ถึงด้านส่ง) เกิดเหตุการณ์ได้ 2 กรณี
 - ถ้า ACK ของ Frame ถัดไปมาถึงก่อนที่ Timer ของ ACK ปัจจุบันจะหมดเวลา ไม่จำเป็นต้องส่งใหม่ ทั้งนี้เนื่องจากการนับ ACK เป็นแบบสะสม นั่นคือ ถ้า ACK 1 ถึง ACK 3 สูญหายไป แต่ ด้านส่งได้รับ ACK 4 ก่อน Timer ของ ACK 1 หมดเวลา ด้านรับได้รับข้อมูล 0, 1, 2, และ 3 เรียบร้อยแล้ว
 - ถ้า ACK ของ Frame ถัดไปมาไม่ทันเวลาของ ACK ปัจจุบันหมด Frame ที่ยังไม่ได้รับ ACK จะถูกส่งใหม่ทั้งหมด

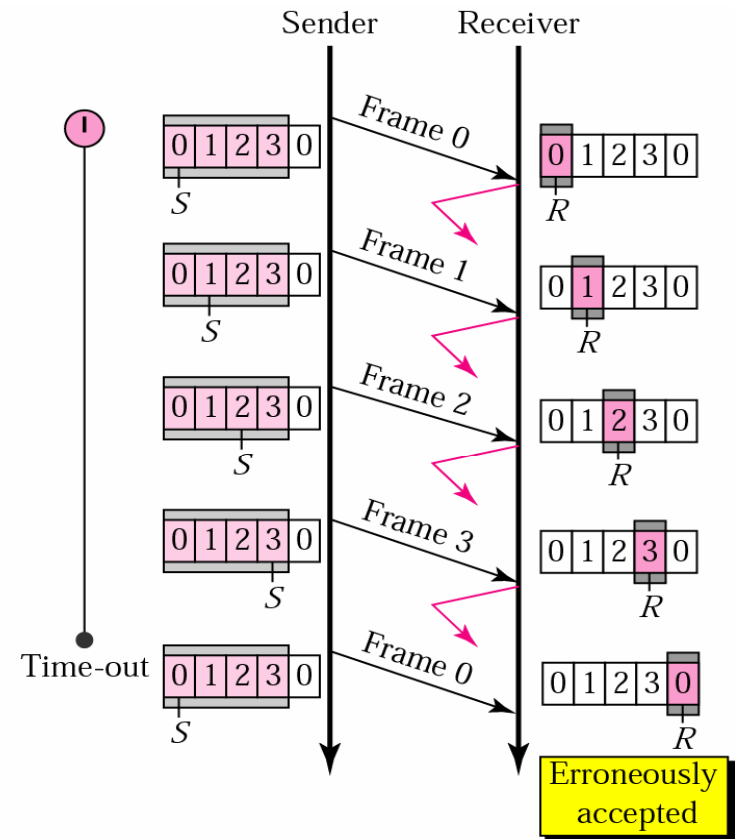
Resending Frame

- การนิ่งเฉยของด้านรับทำให้ Timer ด้านส่งหมดเวลา ซึ่งทำให้ด้านส่ง ทำการส่ง Frame ตั้งแต่ Frame แรกที่ Timer หมดเวลาจนกระทั่งถึง Frame สุดท้ายใน Window ขณะนั้น เช่นถ้า Timer ของ Frame 3 หมดลงแต่ได้ส่งไปถึง Frame ที่ 6 จะทำการส่ง Frame 3, 4, 5, และ 6 ใหม่

Sender Window Size



a. Window size $< 2^m$



b. Window size $= 2^m$

ขนาดของ Window ต้องน้อยกว่า 2^m เมื่อ m คือจำนวนของ bit ที่ใช้แสดง Header ระบุหมายเลขของ Frame เพื่อป้องกันการการสับสน ดังรูป



Selective Repeat ARQ

- **Go-Back- N ARQ**

คือกระบวนการทำ ARQ ซึ่งสามารถส่งข้อมูลได้พร้อมกันหลาย Frame ก่อนที่จะประมวลผล ACK ซึ่งหากเกิดข้อผิดพลาดขึ้น จะส่งใหม่ทั้งกลุ่ม Frame

- **Selective Repeat ARQ**

คือกระบวนการทำ ARQ ซึ่งสามารถส่งข้อมูลได้พร้อมกันหลาย Frame ก่อนที่จะประมวลผล ACK ซึ่งหากเกิดข้อผิดพลาดขึ้น จะส่งเฉพาะ Frame ที่ผิดใหม่

ดังนั้น วิธี Selective Repeat ARQ จึงเหมาะกับ ตัวกลางที่มีสัญญาณรบกวนสูง ซึ่งมีแนวโน้มจะทำให้เกิดข้อผิดพลาดบ่อย จึงไม่ควรส่ง Frame ที่ถูกต้องอยู่แล้วซ้ำ

อย่างไรก็ดี วิธี Selective Repeat ARQ จะเพิ่มความซับซ้อนให้กับกระบวนการของทางด้านรับ ซึ่งจะได้กล่าวต่อไป

Sender and Receiver Windows

★ ด้านส่ง มีตัวแปรควบคุมจำนวน 3 ตัว เหมือนวิธี Go-Back-N ARQ

S เก็บลำดับตัวเลขของ Frame ที่เพิ่งส่งออกไป

S_F เก็บลำดับตัวเลขของ Frame แรกใน Window (**First**)

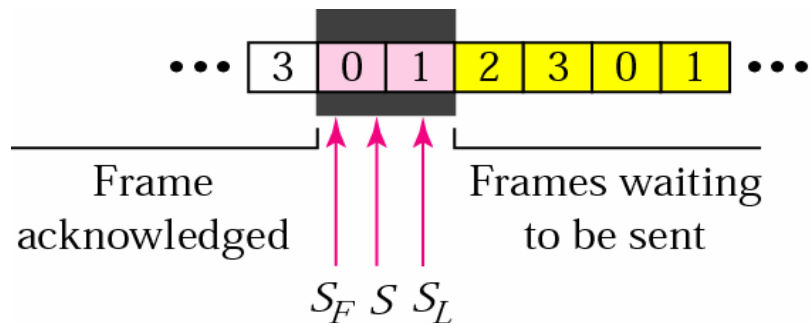
S_L เก็บลำดับตัวเลขของ Frame สุดท้ายใน Window (**Last**)

แต่ ขนาดของ Window หาได้จาก $(S_F - S_L + 1)$ Frame มีค่าครึ่งหนึ่งของ 2^m

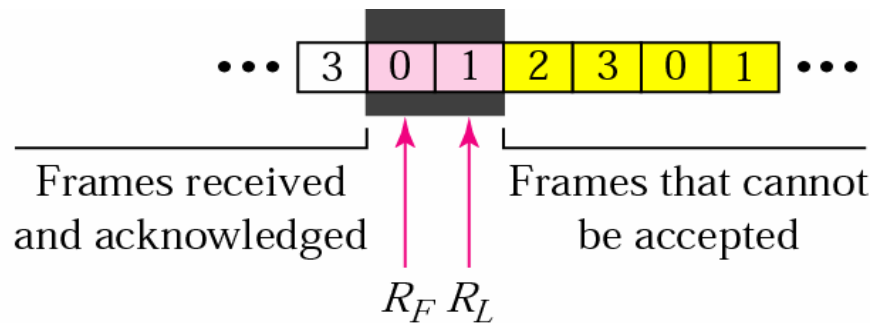
★ ด้านรับ มีตัวแปรควบคุม 3 ตัว ซึ่งเก็บ ช่วงของ Frame ที่คาดหวังจะได้รับ

R_F เก็บลำดับตัวเลขของ Frame แรกใน Window (**First**)

R_L เก็บลำดับตัวเลขของ Frame สุดท้ายใน Window (**Last**)

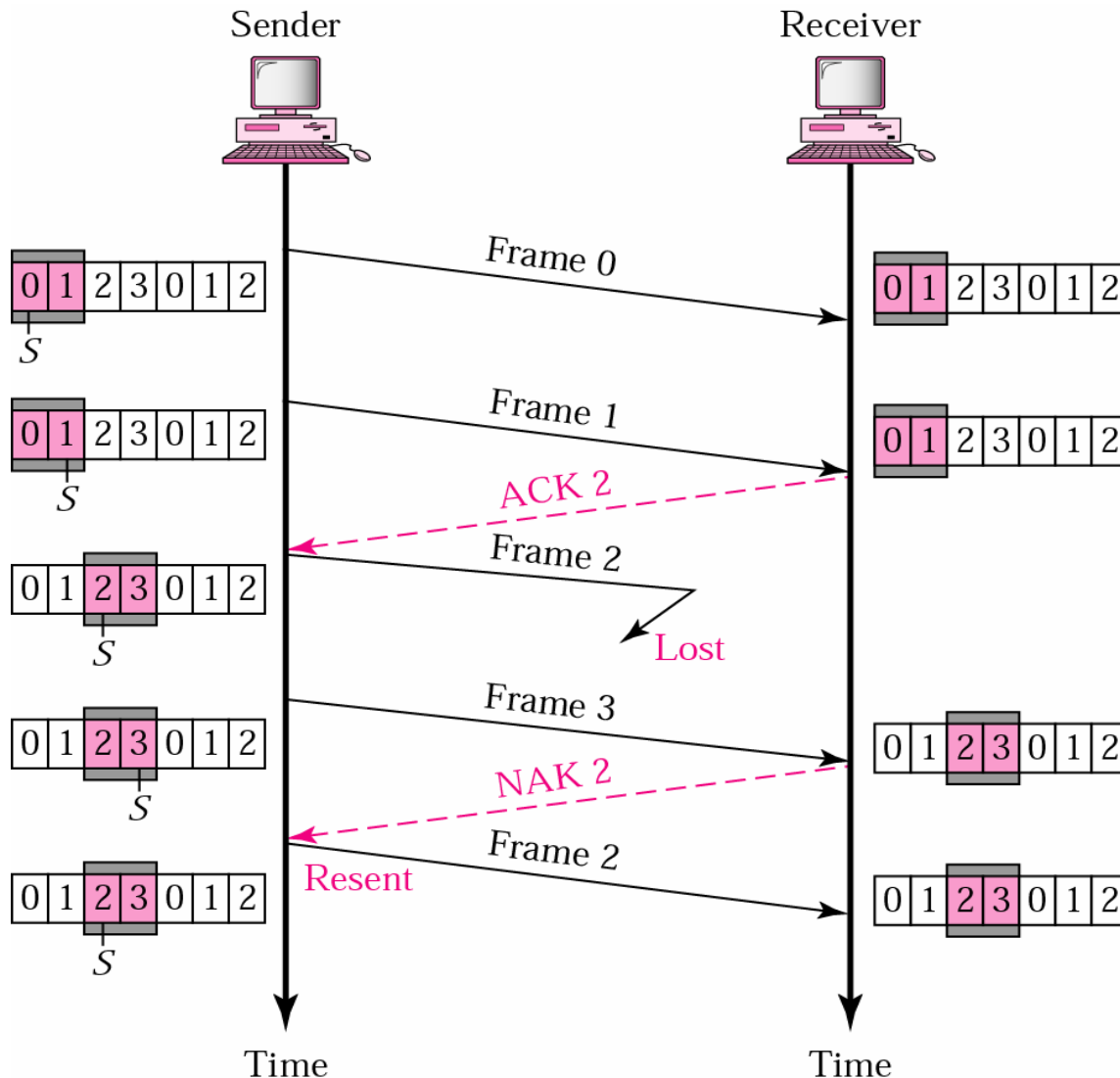


a. Sender window



b. Receiver window

Operations



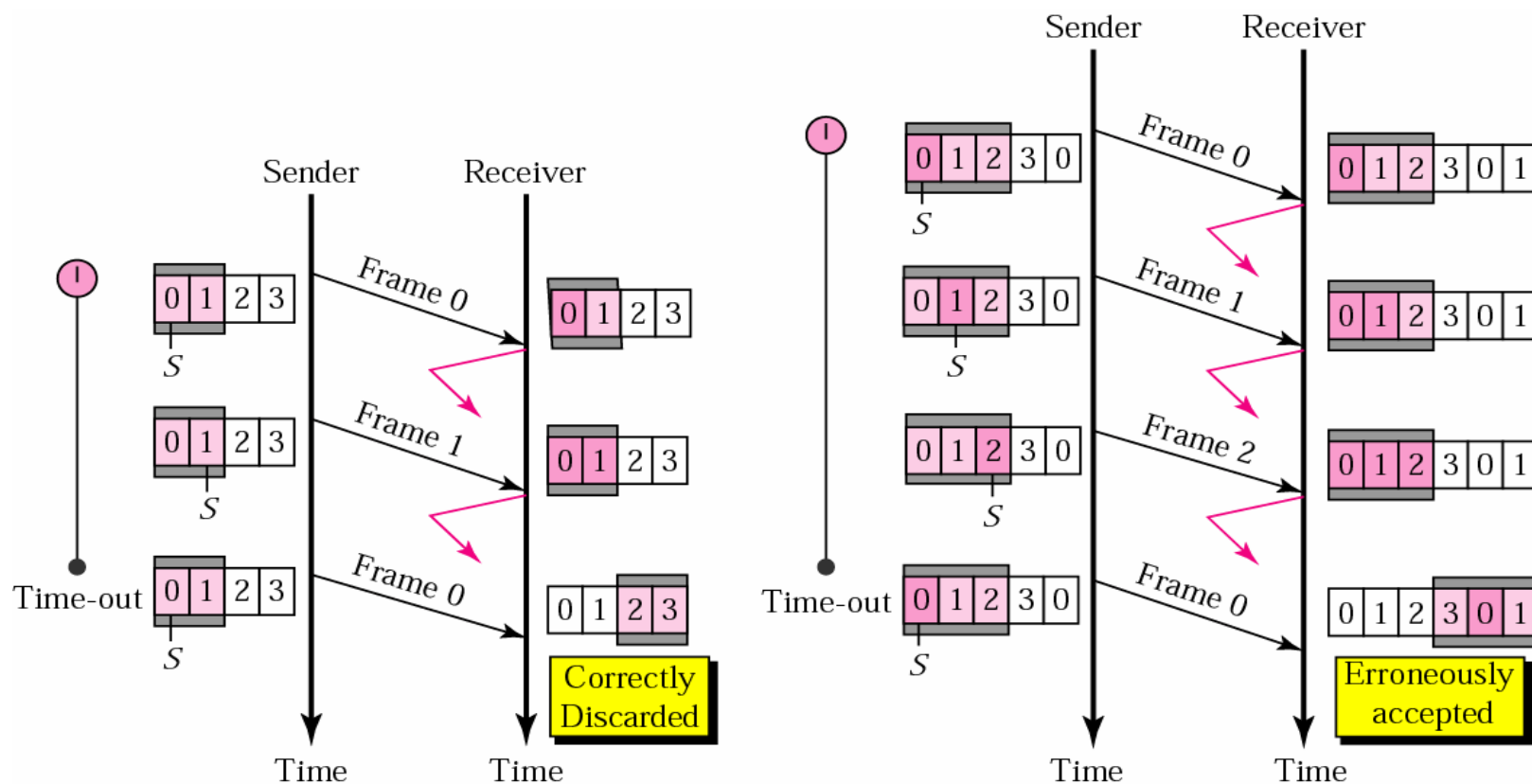
ในกรณี Data Frame
เกิดสูญหาย

เช่น Frame 2 ไปไม่ถึง
ถึงด้านรับ ระหว่างนั้น
ด้านส่งได้ทำการส่ง
Frame 3 ไปด้านรับ
ซึ่งอยู่ภายในช่วงของ
Frame คาดหวัง จึง
ทำการรับ พร้อมกับ
ส่ง NAK 2 (Negative
ACK) ไปเตือนด้านส่ง
ด้านส่งจึงทำการส่ง
Frame 2 ใหม่



Sender Window Size

เงื่อนไขขนาดของ Window ซึ่งต้องเป็นครึ่งหนึ่งของ 2^m นั้น เนื่องจากเหตุผล เพื่อป้องกันการรับ Frame ซ้ำ กรณีที่ ACK สูญหายระหว่างทางดังรูปด้านขวา



a. Window size = 2^{m-1}

b. Window size $> 2^{m-1}$



Lecture Outline

- **Flow and Error Control**
- **Stop-and-Wait ARQ**
- **Go-Back-N ARQ**
 - Sender/Receiver Sliding Window
 - Control Variables
 - Timers
 - Acknowledgement, Resending Frame
 - Operation
 - Sender Window Size and Bi-Dir Tx and Piggybacking
- **Selective Repeat ARQ**
- **HDLC**
 - Configuration and Mechanism
 - Formats and Examples