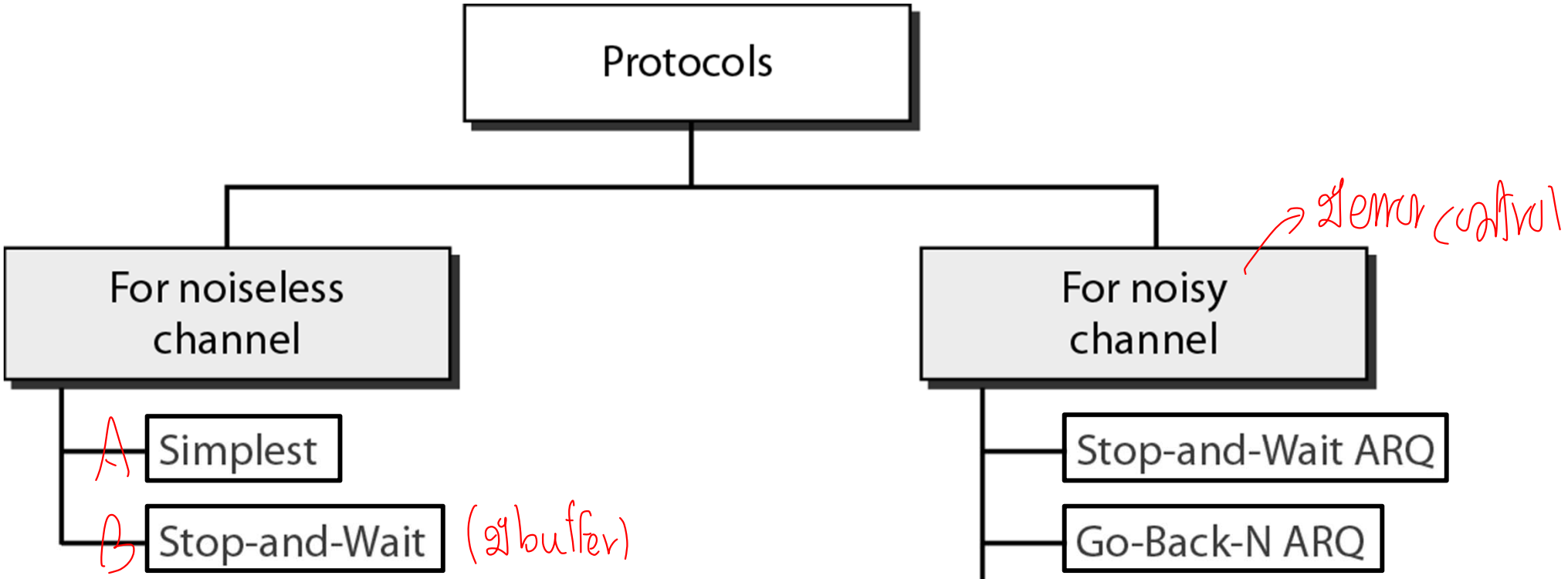
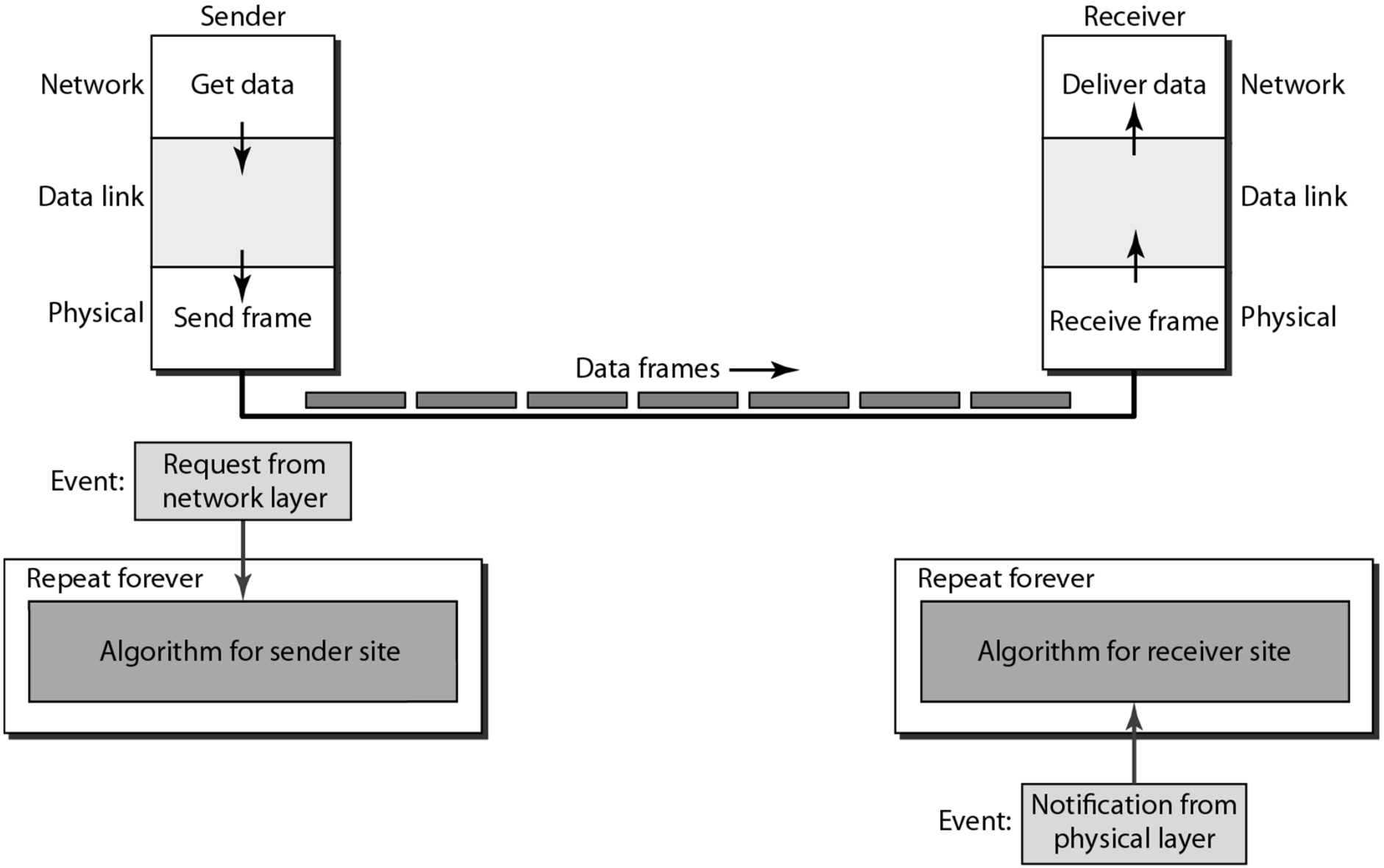
Protocols

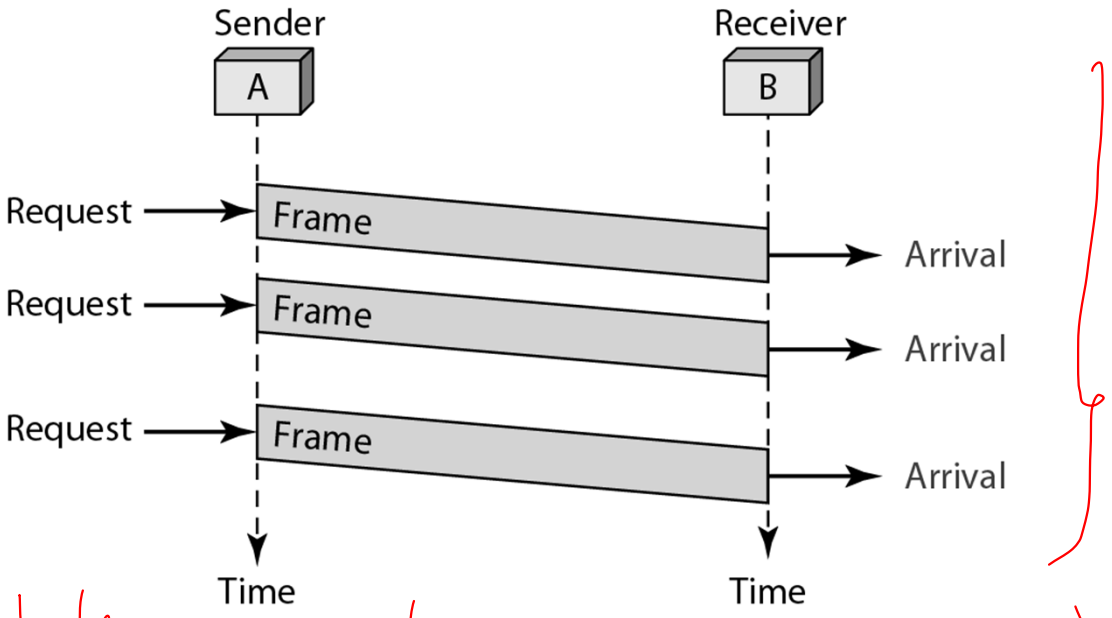

Protocols
For noiseless channel
For noisy channel
Simplest
Stop-and-Wait
Stop-and-Wait ARQ
Go-Back-N ARQ
Selective Repeat ARQ

Simplest Protocol



Simplest Protocol

- Flow diagram



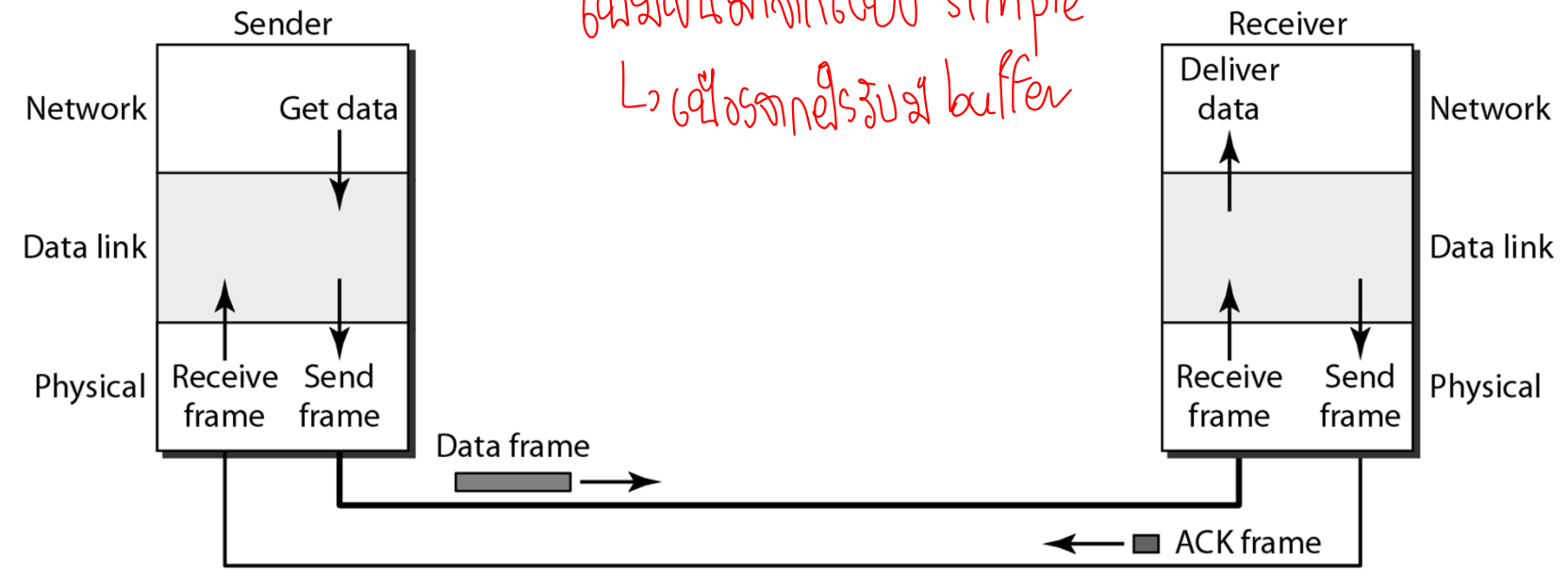
- Algorithm

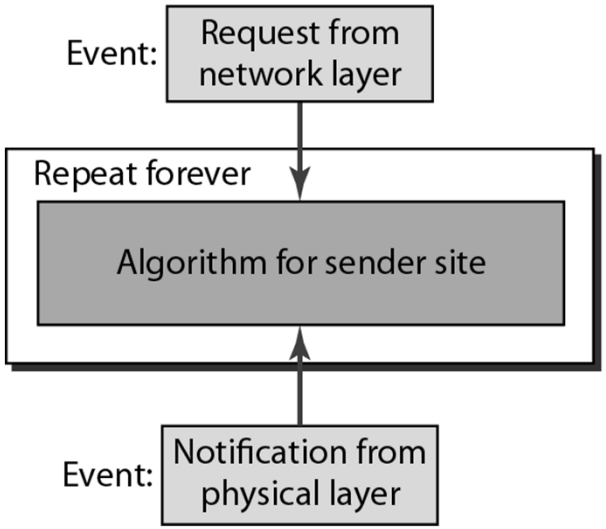
```
1  while(true)                    // Repeat forever
2  {
3    WaitForEvent();              // Sleep until an event occurs
4    if(Event(RequestToSend))     //There is a packet to send
5    {
6       GetData();
7       MakeFrame();
8       SendFrame();              //Send the frame
9    }
10 }
```

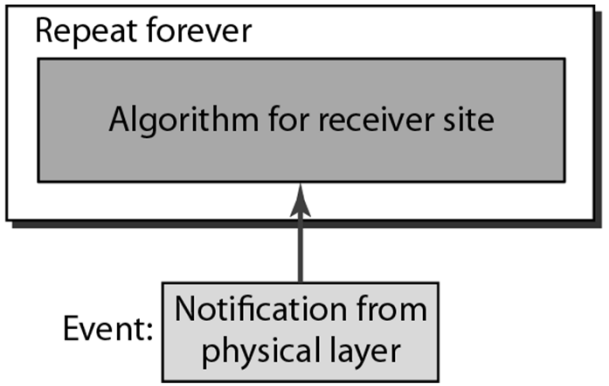
```
1  while(true)                    // Repeat forever
2  {
3    WaitForEvent();              // Sleep until an event occurs
4    if(Event(ArrivalNotification)) //Data frame arrived
5    {
6       ReceiveFrame();
7       ExtractData();
8       DeliverData();            //Deliver data to network layer
9    }
10 }
```

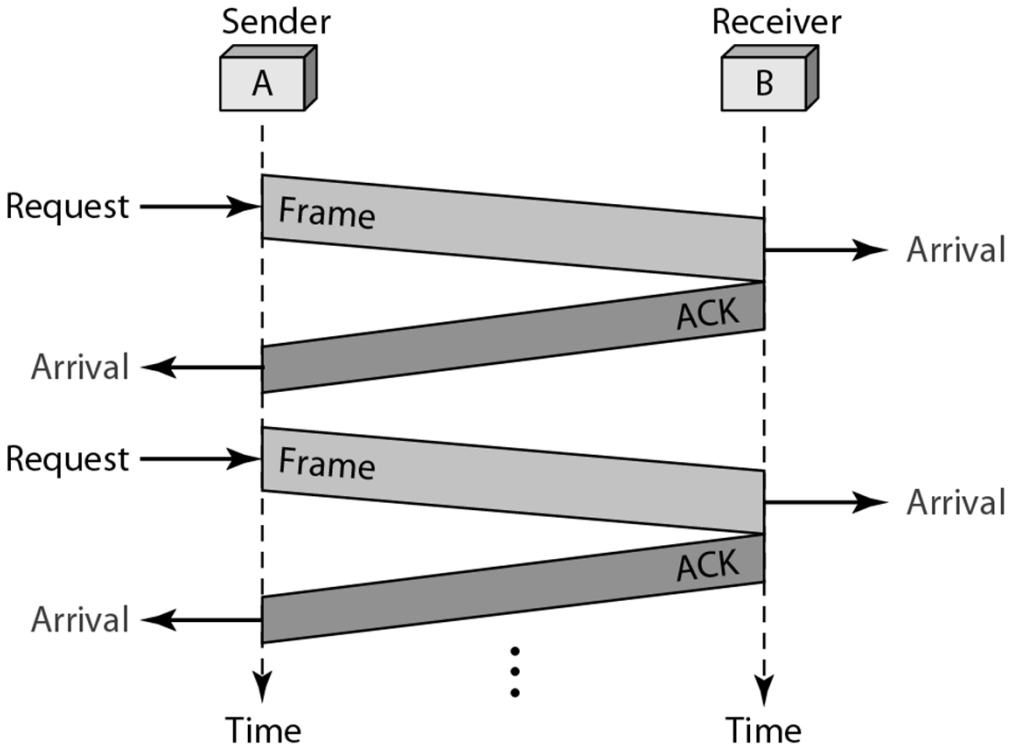
Stop-and-Wait Protocol

เพิ่มขึ้นมาจากแบบ simple
คือมีการใช้ buffer









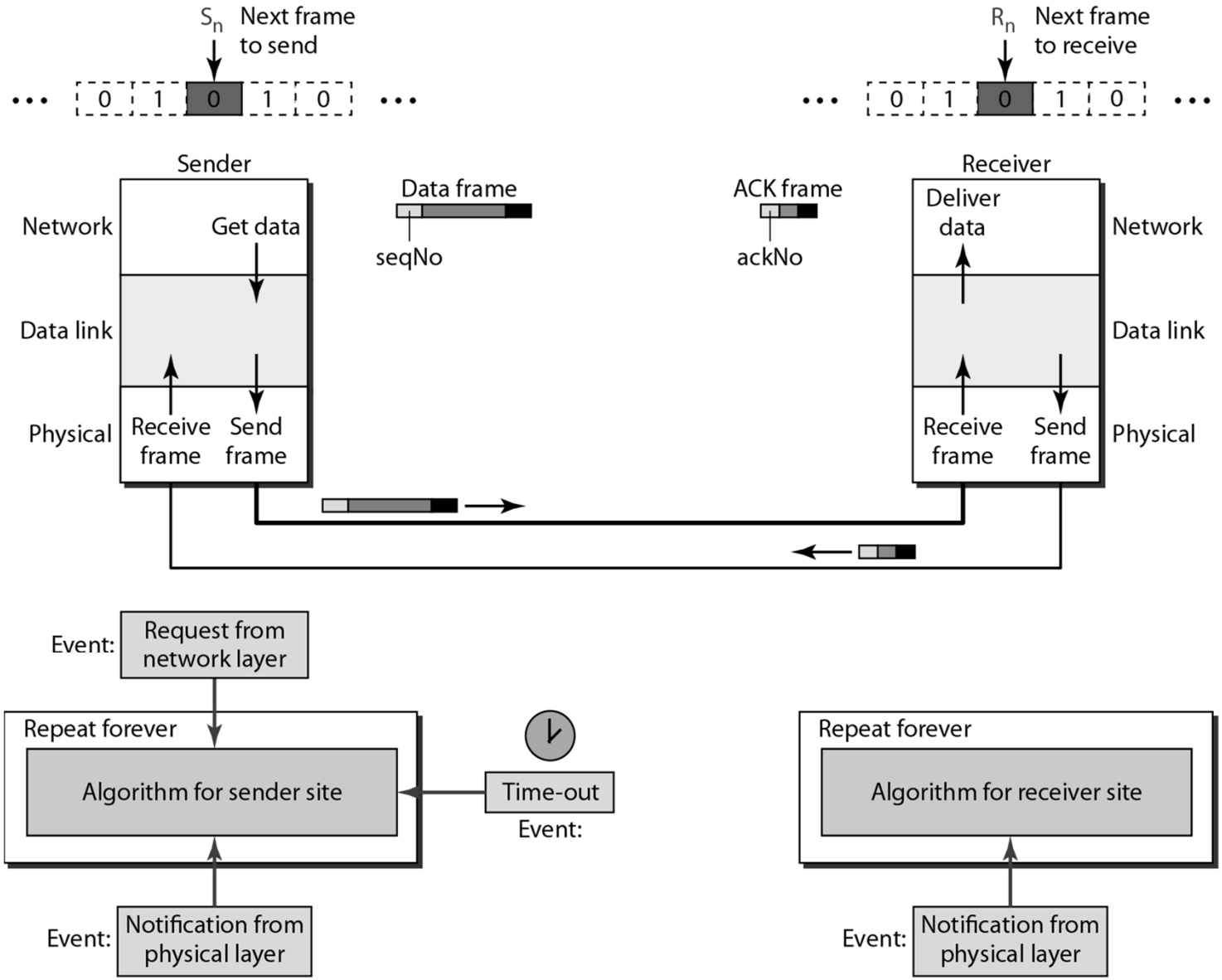

```
1  while(true)                          //Repeat forever
2  canSend = true                       //Allow the first frame to go
3  {
4    WaitForEvent();                    // Sleep until an event occurs
5    if(Event(RequestToSend) AND canSend)
6    {
7       GetData();
8       MakeFrame();
9       SendFrame();                    //Send the data frame
10      canSend = false;                //Cannot send until ACK arrives
11   }
12   WaitForEvent();                    // Sleep until an event occurs
13   if(Event(ArrivalNotification)      // An ACK has arrived
14    {
15      ReceiveFrame();                 //Receive the ACK frame
16      canSend = true;
17    }
18 }
```

```
1  while(true)                          //Repeat forever
2  {
3    WaitForEvent();                    // Sleep until an event occurs
4    if(Event(ArrivalNotification)) //Data frame arrives
5    {
6       ReceiveFrame();
7       ExtractData();
8       Deliver(data);                  //Deliver data to network layer
9       SendFrame();                    //Send an ACK frame
10   }
11 }
```

Stop-and-Wait Automatic Repeat Request

- copy & retransmitting frame
  - when the timer expires => Error correction
- frame sequence numbers
  - based on modulo-2 arithmetic
- the acknowledgment number
  - modulo-2 arithmetic the sequence number of the next frame expected

Stop-and-Wait Automatic Repeat Request



Stop-and-Wait Automatic Repeat Request

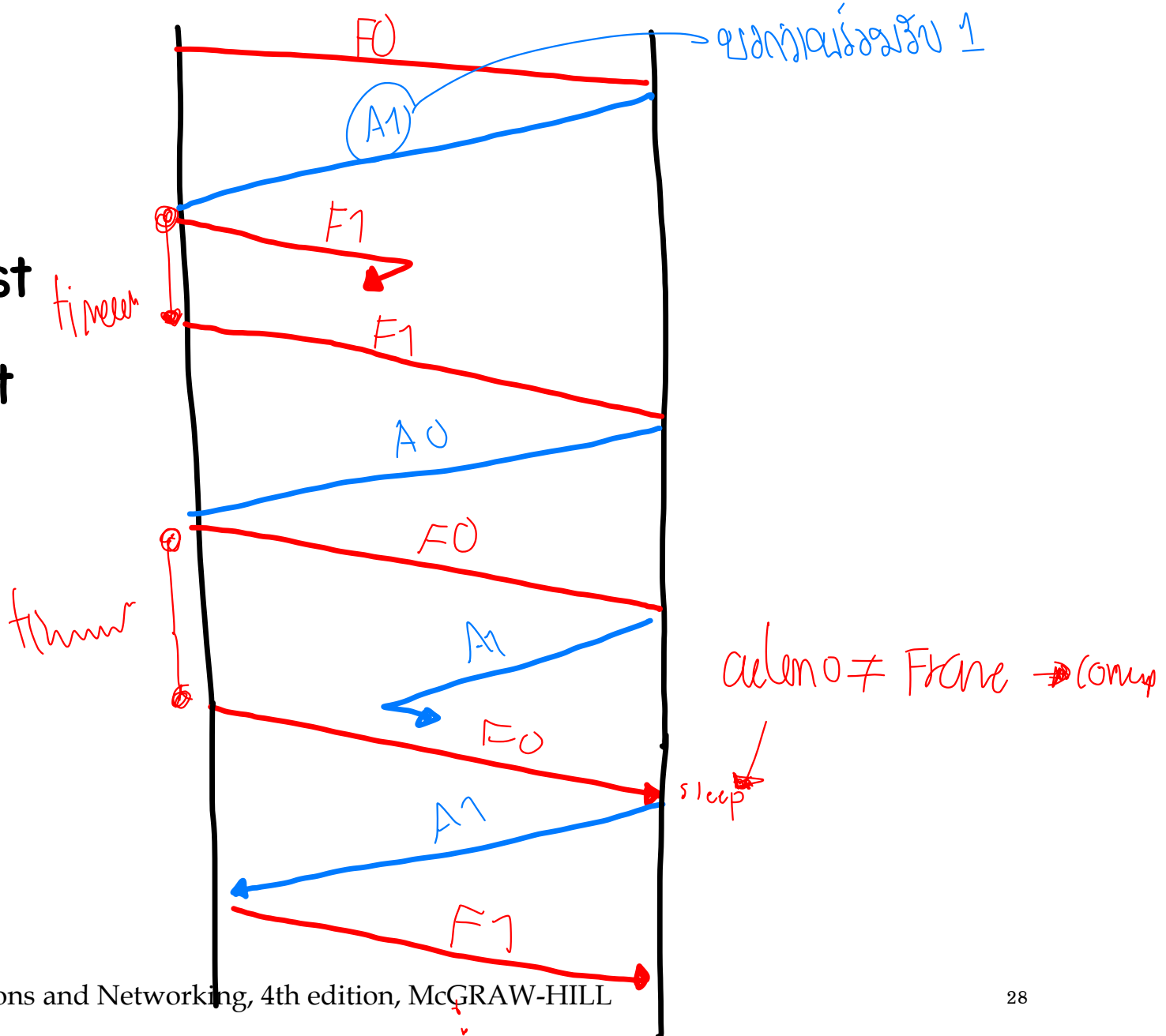
- **Algorithm**

```
1  Sn = 0;                          // Frame 0 should be sent first
2  canSend = true;                  // Allow the first request to go
3  while(true)                      // Repeat forever
4  {
5    WaitForEvent();                // Sleep until an event occurs
6    if(Event(RequestToSend) AND canSend)
7    {
8       GetData();
9       MakeFrame(Sn);              //The seqNo is Sn
10      StoreFrame(Sn);             //Keep copy
11      SendFrame(Sn);
12      StartTimer();
13      Sn = Sn + 1;
14      canSend = false;
15   }
16   WaitForEvent();                // Sleep
17     if(Event(ArrivalNotification)) // An ACK has arrived
18     {
19       ReceiveFrame(ackNo);       //Receive the ACK frame
20       if(not corrupted AND ackNo == Sn) //Valid ACK
21          {
22            Stoptimer();
23            PurgeFrame(Sn-1);     //Copy is not needed
24            canSend = true;
25          }
26      }
27
28     if(Event(TimeOut))           // The timer expired
29     {
30      StartTimer();
31      ResendFrame(Sn-1);          //Resend a copy check
32     }
33 }
```

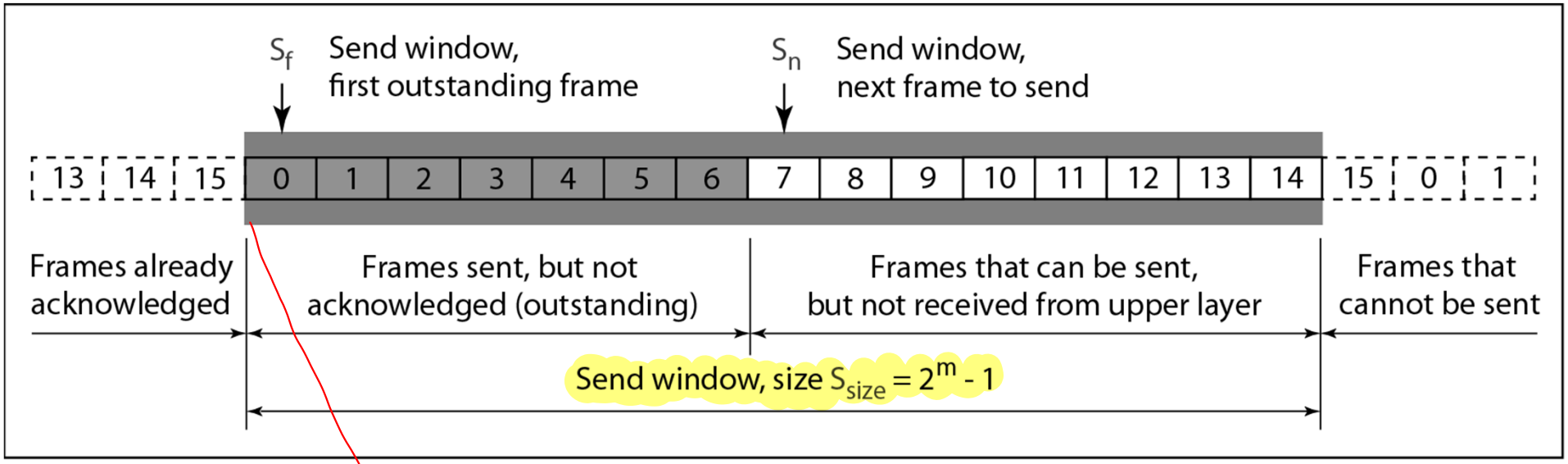
```
1  Rn = 0;                          // Frame 0 expected to arrive first
2  while(true)
3  {
4    WaitForEvent();                // Sleep until an event occurs
5    if(Event(ArrivalNotification))  //Data frame arrives
6    {
7       ReceiveFrame();
8       if(corrupted(frame));
9          SendFrame(Rn);
10      if(seqNo == Rn)             //Valid data frame
11      {
12       ExtractData();
13        DeliverData();            //Deliver data
14        Rn = Rn + 1;
15      }
16       SendFrame(Rn);             //Send an ACK
17   }
18 }
```

Stop-and-Wait Automatic Repeat Request

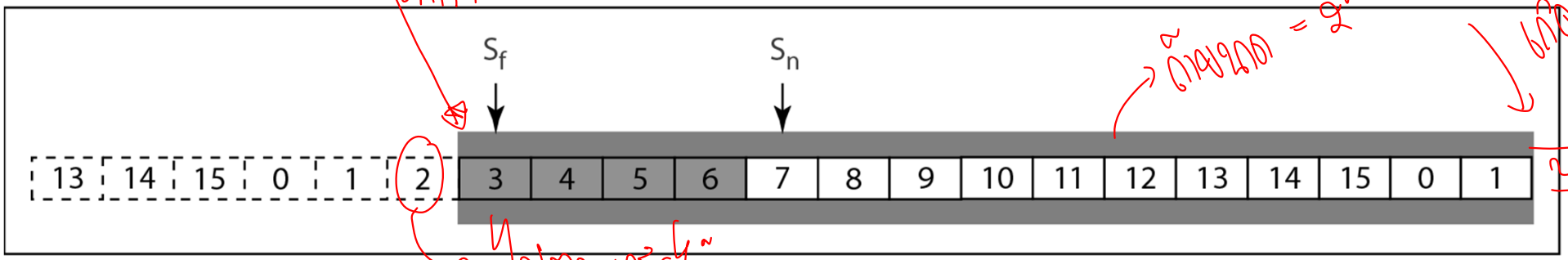
- Flow diagram
  - Send 5
  - Data 2nd lost
  - ACK 3th lost



Go-Back-N Automatic Repeat Request



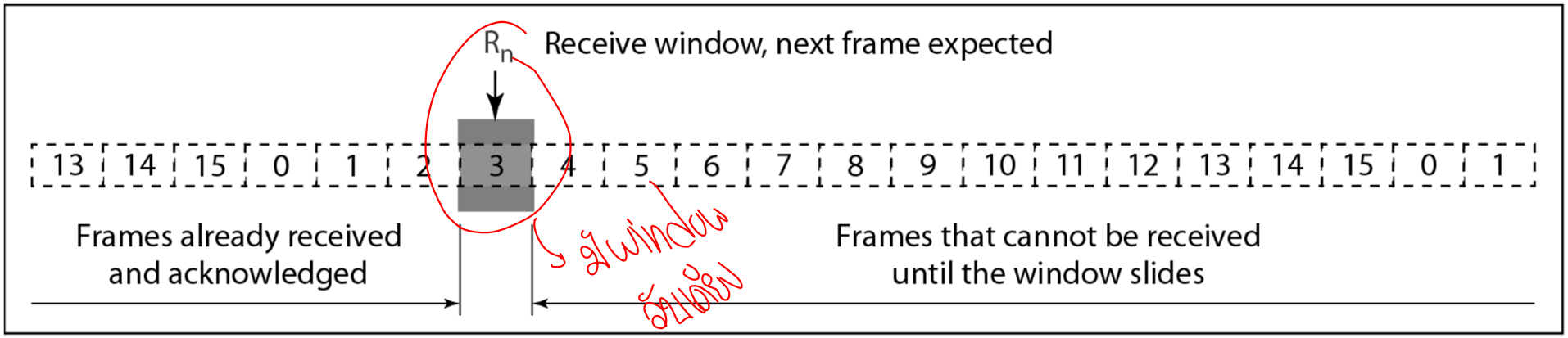
a. Send window before sliding



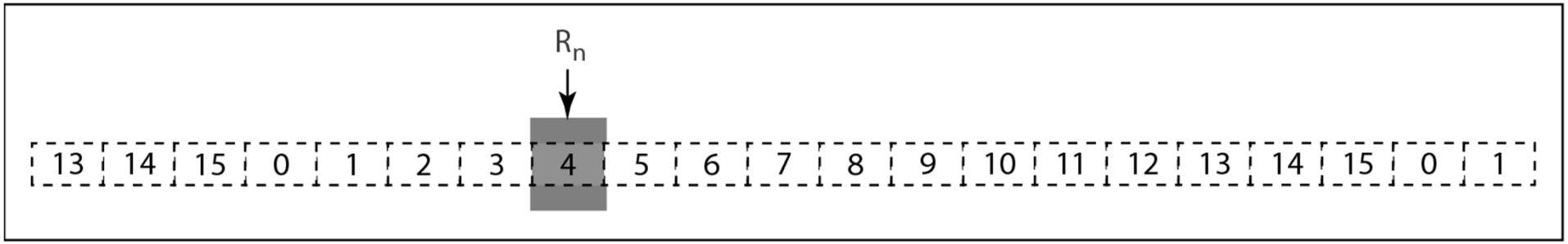
b. Send window after sliding

- the sequence numbers are modulo 2^m, where m is the size of the sequence number field in bits.

Go-Back-N Automatic Repeat Request

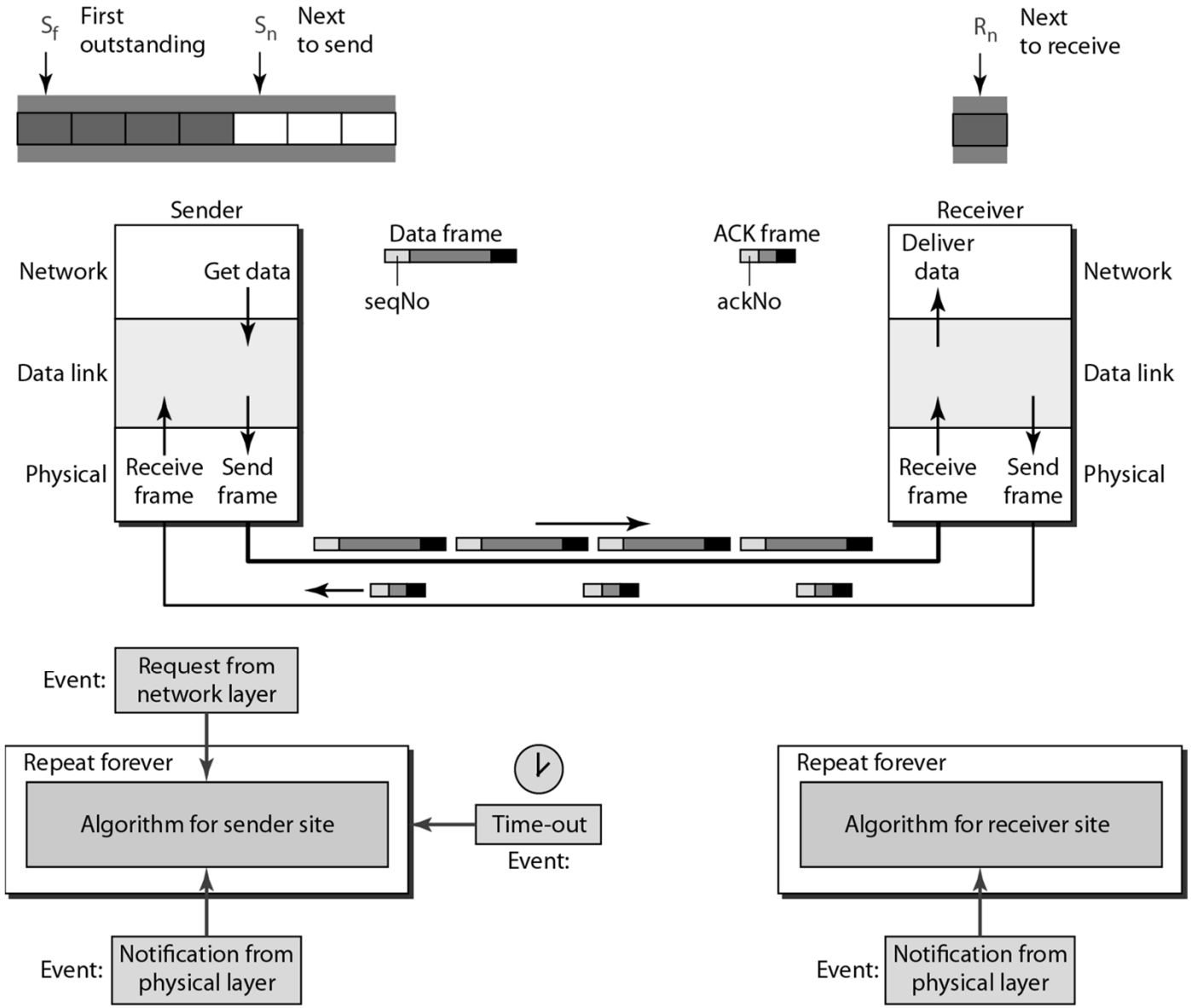


a. Receive window

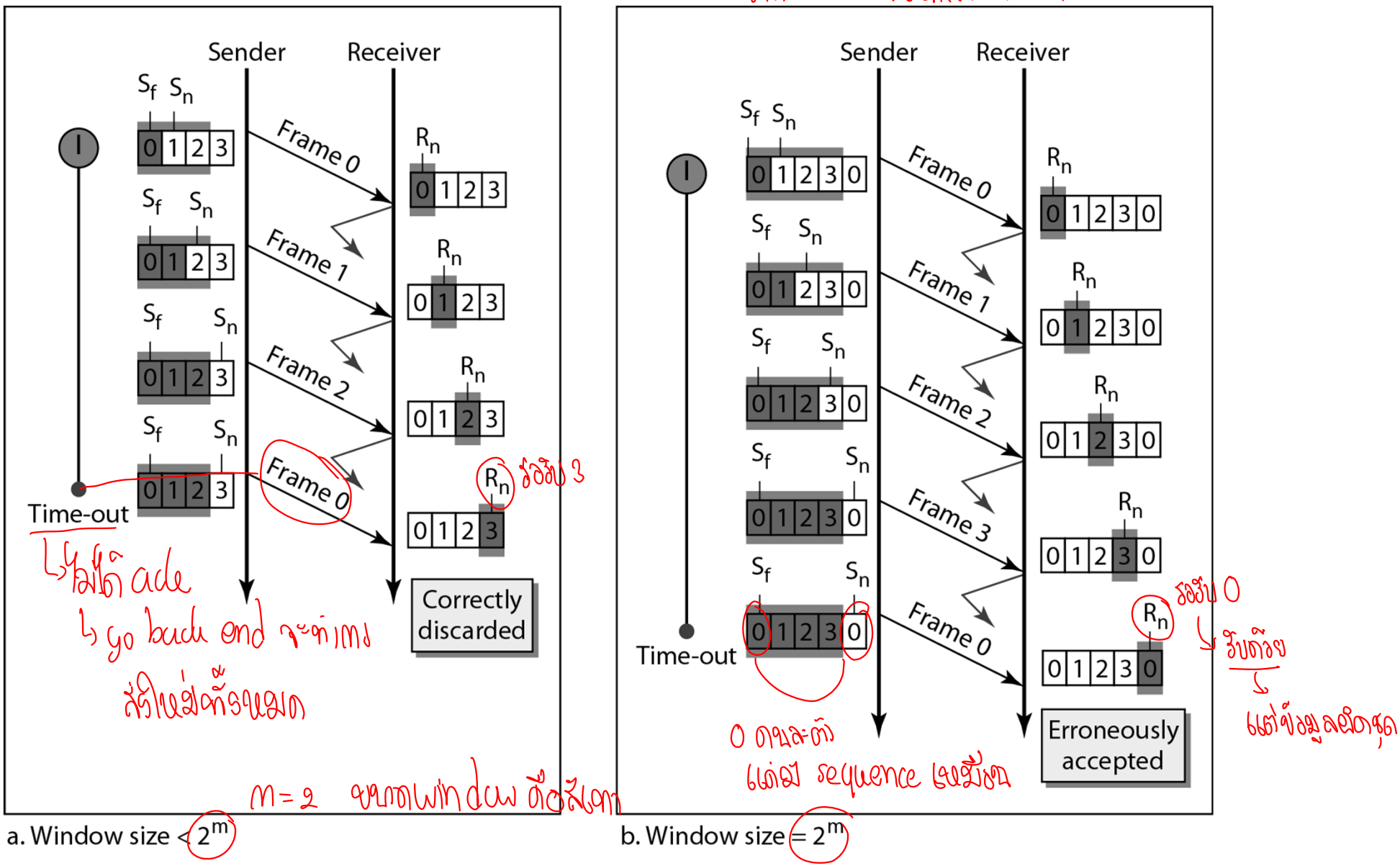


b. Window after sliding

Go-Back-N Automatic Repeat Request



Go-Back-N Automatic Repeat Request

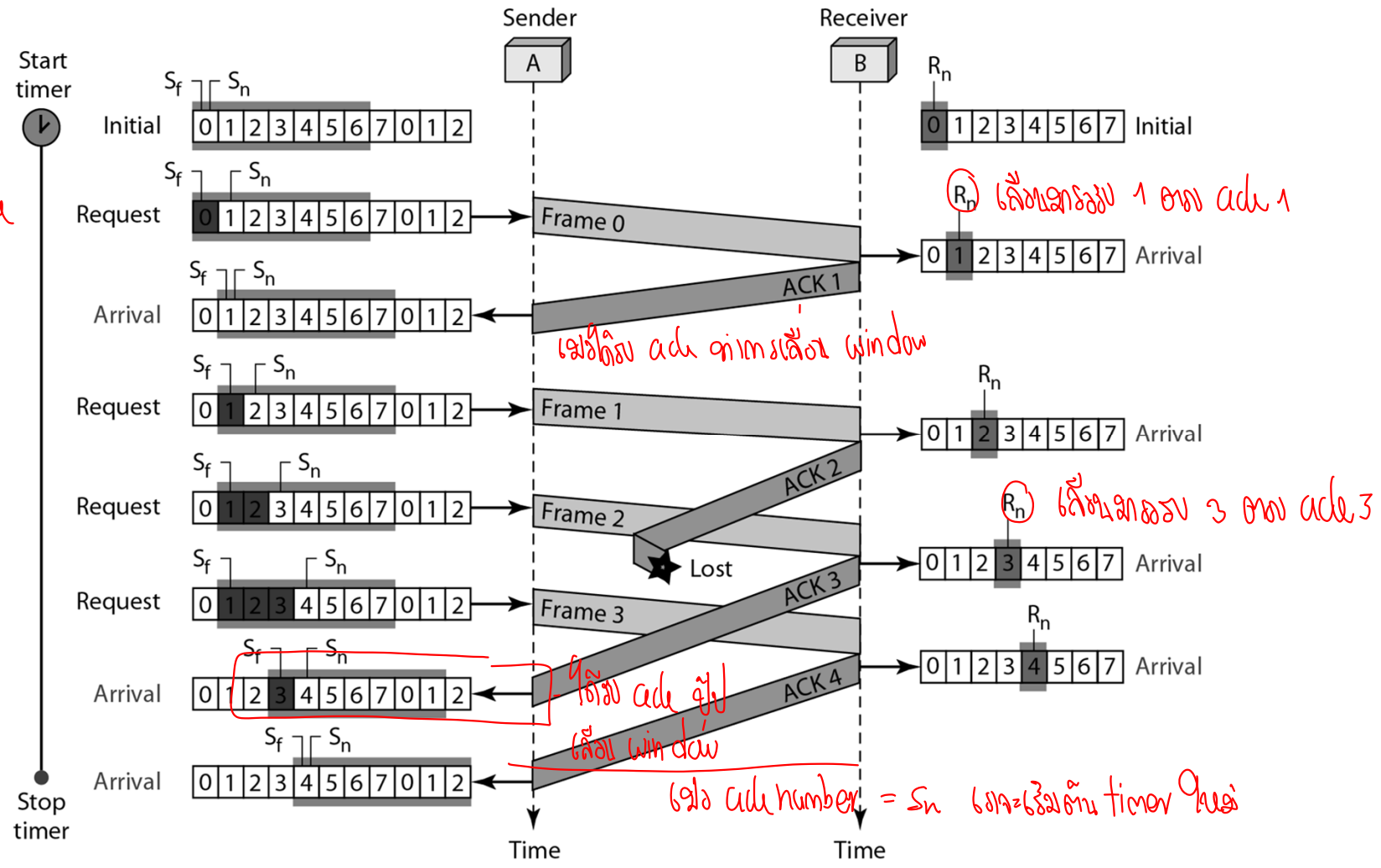


```
Sw = 2^m - 1;
Sf = 0;
Sn = 0;

while (true)                          //Repeat forever
{
 WaitForEvent();
  if(Event(RequestToSend))            //A packet to send
  {
     if(Sn-Sf >= Sw)                  //If window is full
           Sleep();
     GetData();
     MakeFrame(Sn);
     StoreFrame(Sn);
     SendFrame(Sn);
     Sn = Sn + 1;
     if(timer not running)
          StartTimer();
  }

  if(Event(ArrivalNotification))  //ACK arrives
  {
     Receive(ACK);
     if(corrupted(ACK))
          Sleep();
     if((ackNo>Sf)&&(ackNo<=Sn))  //If a valid ACK
     While(Sf <= ackNo)
      {
       PurgeFrame(Sf);
       Sf = Sf + 1;
      }
  }

  if(Event(TimeOut))              //The timer expires
  {
   StartTimer();
   Temp = Sf;
   while(Temp < Sn);
    {
     SendFrame(Sf);
    }
  }
}
```

```
Rn = 0;

while (true)                          //Repeat forever
{
  WaitForEvent();

  if(Event(ArrivalNotification)) /Data frame arrives
  {
     Receive(Frame);
     if(corrupted(Frame))
          Sleep();
     if(seqNo == Rn)                  //If expected frame
     {
       DeliverData();                 //Deliver data
       Rn = Rn + 1;                   //Slide window
       SendACK(Rn);
     }
  }
}
```

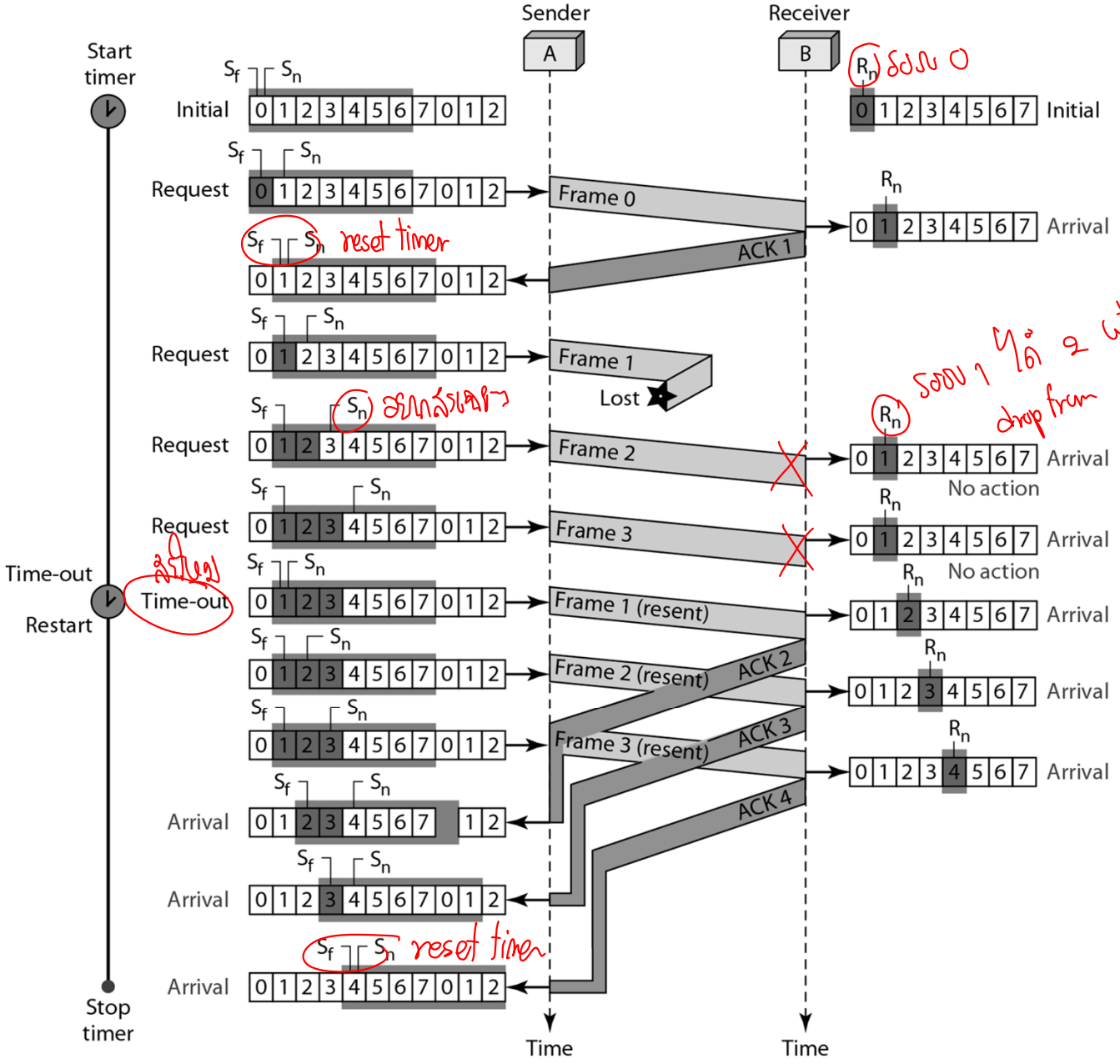
Go-Back-N Automatic Repeat Request

- Flow diagram



Go-Back-N Automatic Repeat Request

- Flow diagram



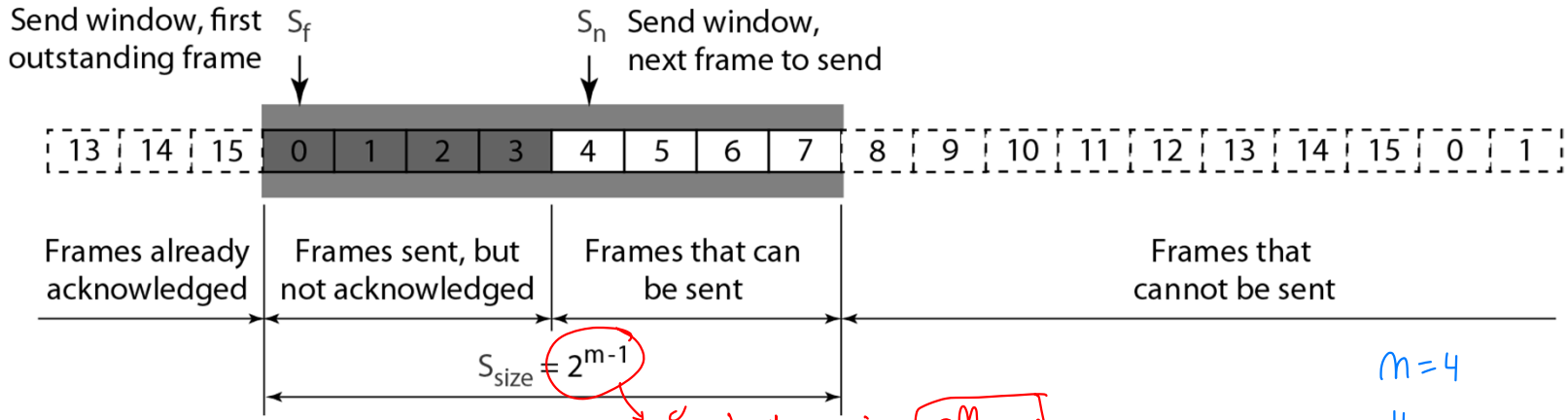
Selective Repeat Automatic Repeat Request

ข้อมากกว่า go back n

↳ ขนาด window ยังเท่าเดิม

↳ แต่ตัวเลข window ยังเท่าเดิม

- **Send window**



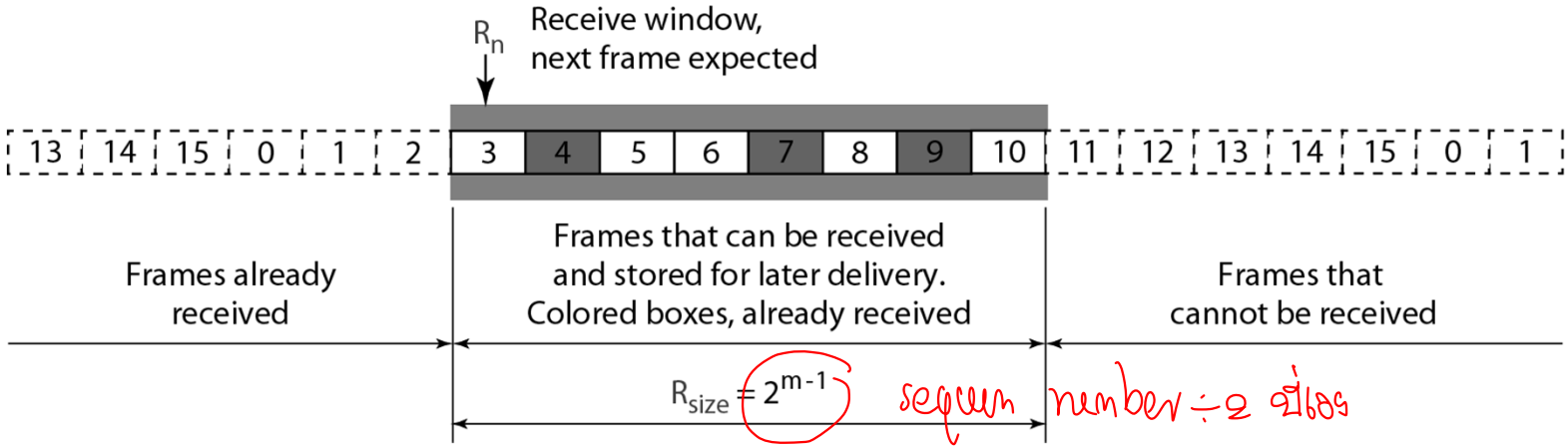
อันนี้เท่าเดิมเสมอ $2^m - 1$

$m = 4$

$2^4 = 16$

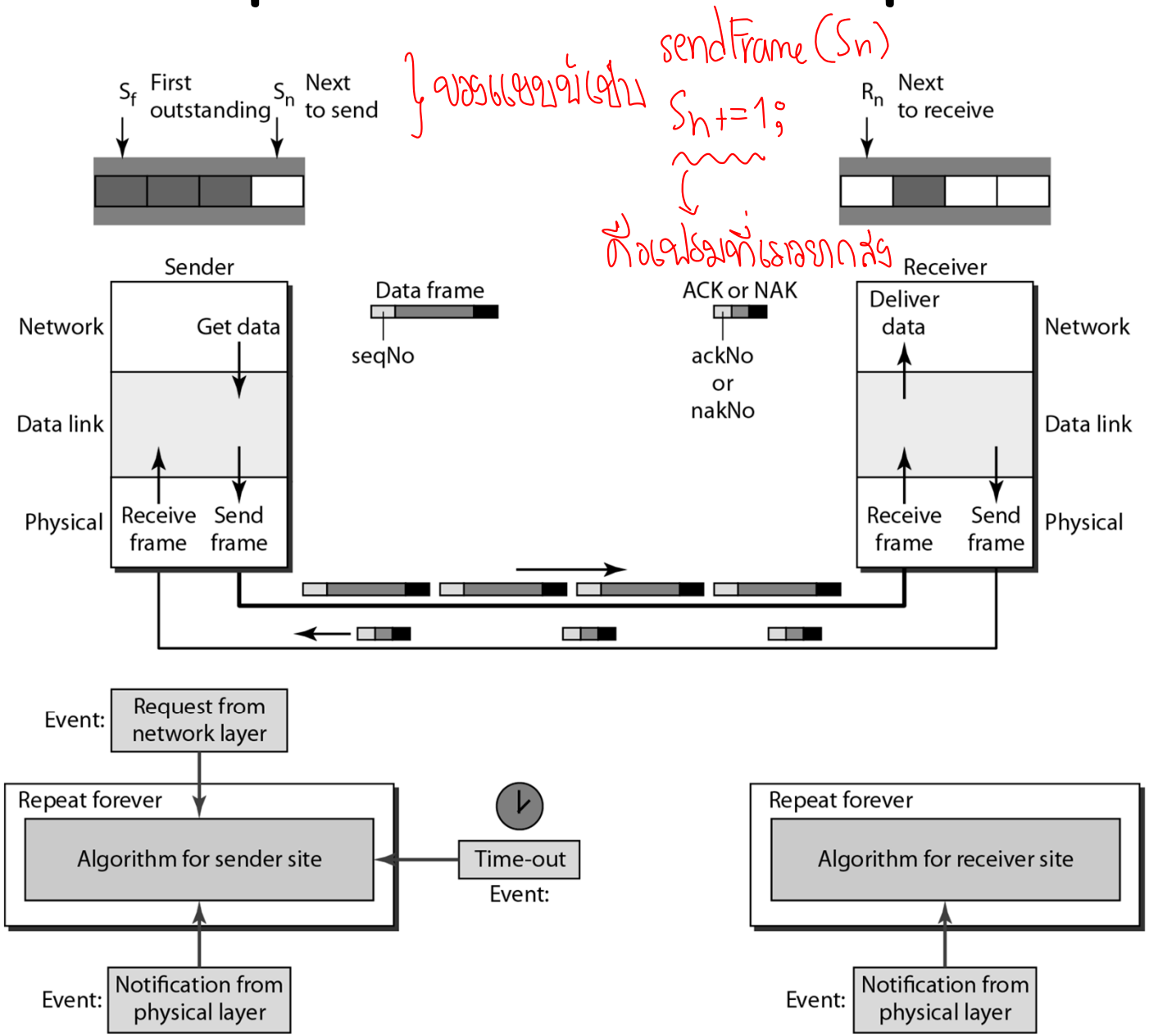
$2^3 = 8$

- **Receive window**

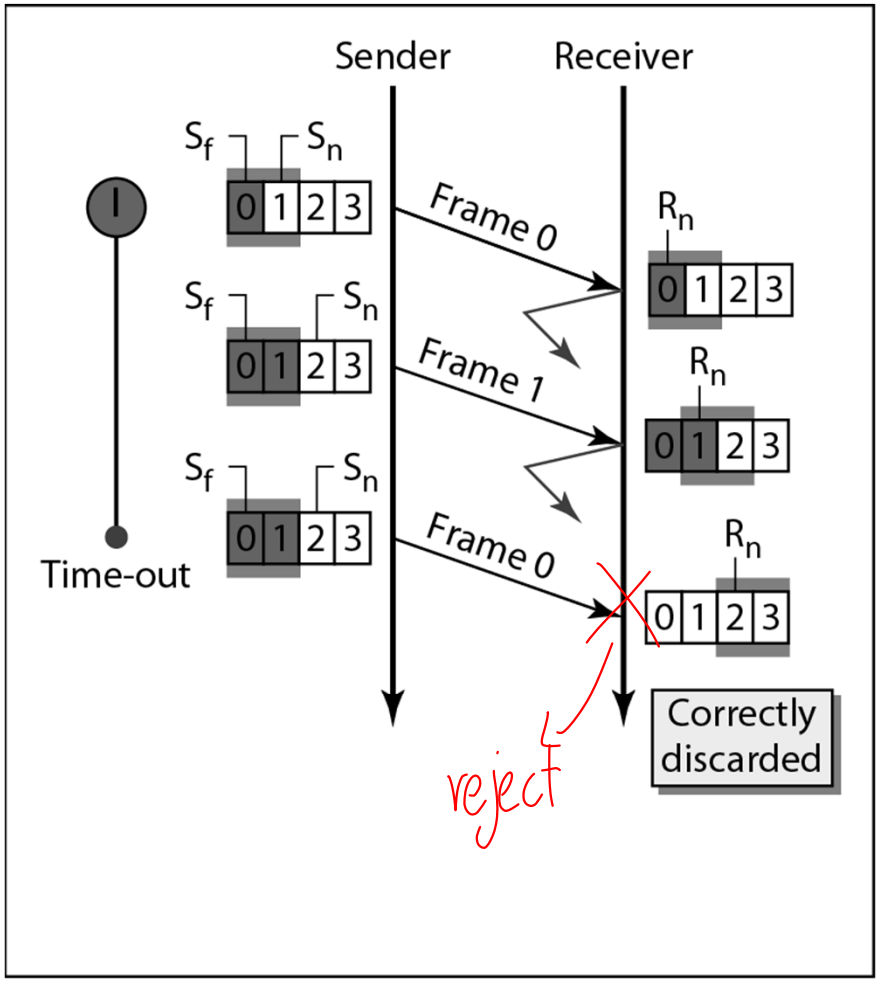


sequen number ÷2 ตัวเอง

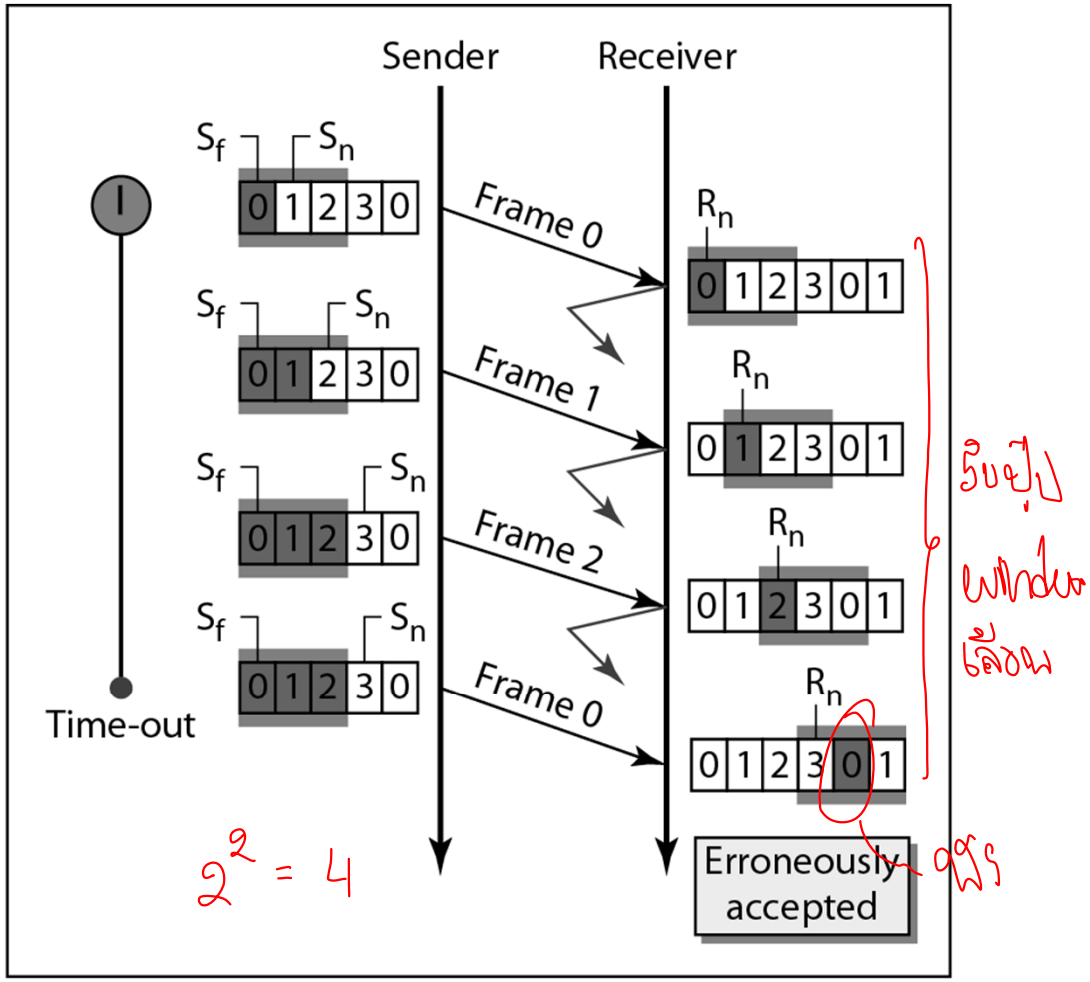
Selective Repeat Automatic Repeat Request



Selective Repeat Automatic Repeat Request



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



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

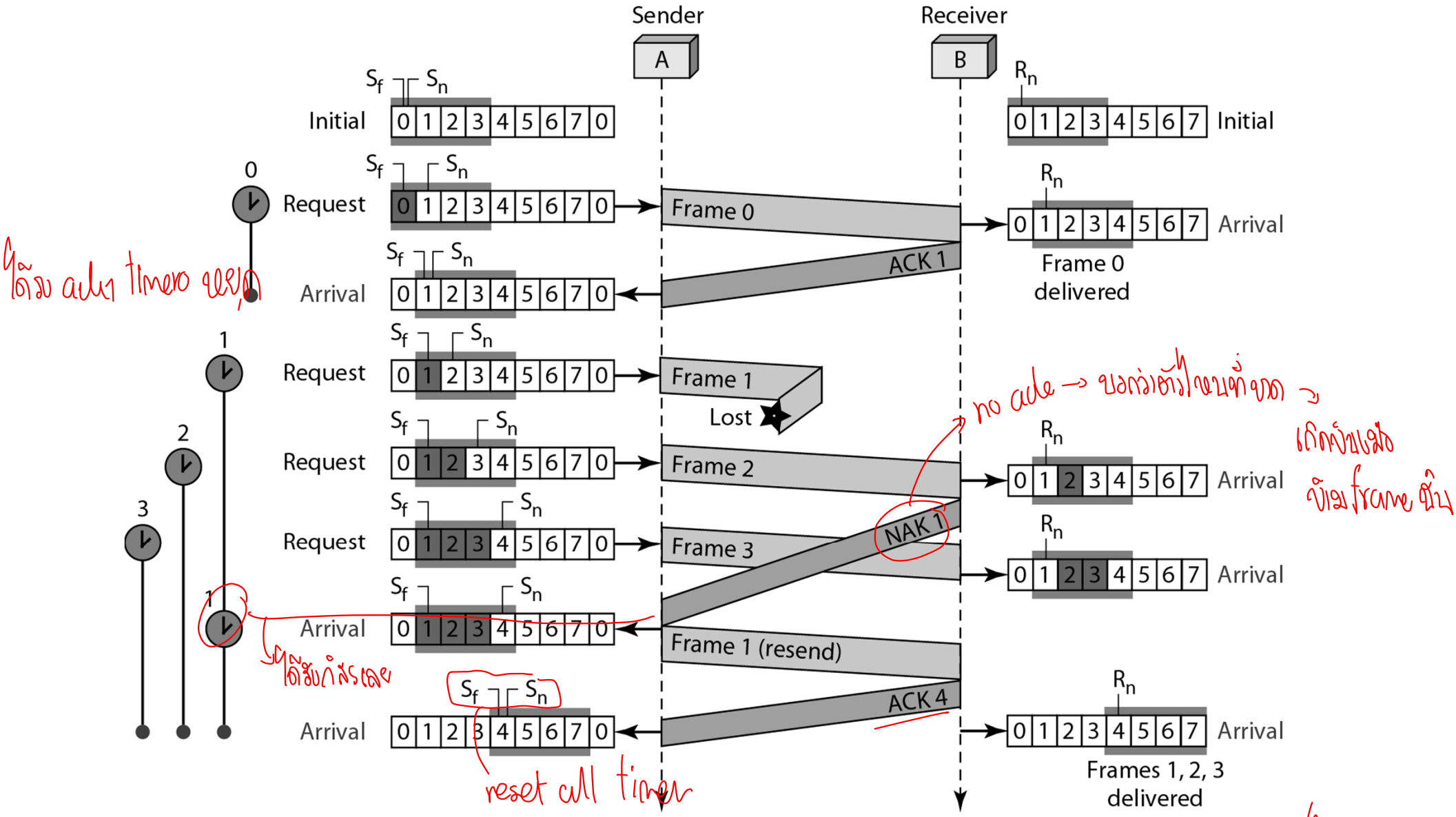
```
Sw = 2^(m-1) ;
Sf = 0;
Sn = 0;

while (true)                          //Repeat forever
{
  WaitForEvent();
  if(Event(RequestToSend))            //There is a packet to send
  {
     if(Sn-Sf >= Sw)                  //If window is full
            Sleep();
     GetData();
     MakeFrame(Sn);
     StoreFrame(Sn);
     SendFrame(Sn);
     Sn = Sn + 1;
     StartTimer(Sn);
  }

  if(Event(ArrivalNotification)) //ACK arrives
  {
     Receive(frame);                  //Receive ACK or NAK
     if(corrupted(frame))
          Sleep();
     if (FrameType == NAK)
        if (nakNo between Sf and Sn)
        {
         resend(nakNo);
         StartTimer(nakNo);
        }
     if (FrameType == ACK)
        if (ackNo between Sf and Sn)
        {
          while(sf < ackNo)
          {
           Purge(sf);
           StopTimer(sf);
           Sf = Sf + 1;
          }
        }
  }

  if(Event(TimeOut(t)))               //The timer expires
  {
   StartTimer(t);
   SendFrame(t);
  }
}
```

Handwritten notes (left listing): โค้ดเพิ่มขึ้นเรื่อยๆ; line `Sleep();` (window full): // ถ้าถึงขอบเกิน range ของ window.; `StartTimer(Sn);`: // สั่งจับ timer frame ต่อ frame; `if(corrupted(frame))`: // ack ที่ส่งกลับไม่ตรง; `if (FrameType == NAK)`: // ถ้าเป็น nak; `if (nakNo between Sf and Sn)`: // nak อยู่ใน range window; `resend(nakNo);`: // ส่ง frame นั้นใหม่; `StartTimer(nakNo);`: // พร้อมตั้ง timer; `if (FrameType == ACK)`: // ถ้าเป็น ack; `Purge(sf);`: // เอาตัว copy ออก; `Sf = Sf + 1;`: // เลื่อน window จนถึง ack ที่ได้รับ; `StartTimer(t);`: // นับเวลาเริ่มต้นเวลาใหม่; `SendFrame(t);`: // ทำการส่ง frame นั้น

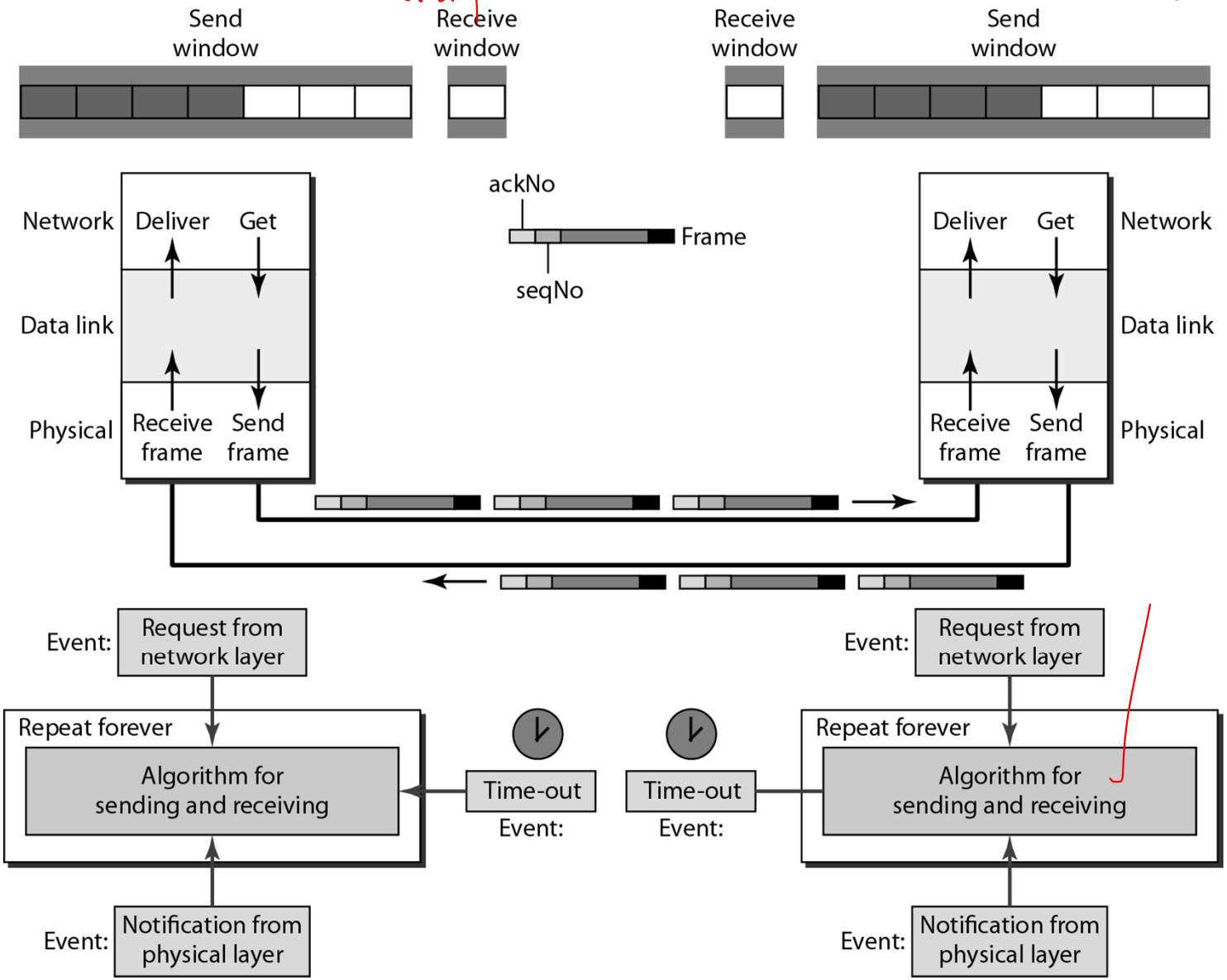
```
Rn = 0;
NakSent = false;
AckNeeded = false;
Repeat(for all slots)
    Marked(slot) = false;

while (true)                               //Repeat forever
{
  WaitForEvent();

   if(Event(ArrivalNotification))          /Data frame arrives
   {
      Receive(Frame);
      if(corrupted(Frame))&& (NOT NakSent)
      {
       SendNAK(Rn);
       NakSent = true;
       Sleep();
      }
      if(seqNo <> Rn)&& (NOT NakSent)
      {
       SendNAK(Rn);
       NakSent = true;
       if ((seqNo in window)&&(!Marked(seqNo))
       {
        StoreFrame(seqNo)
        Marked(seqNo)= true;
        while(Marked(Rn))
        {
         DeliverData(Rn);
         Purge(Rn);
         Rn = Rn + 1;
         AckNeeded = true;
        }
         if(AckNeeded);
         {
         SendAck(Rn);
         AckNeeded = false;
         NakSent = false;
         }
       }
      }
   }
}
```

Handwritten notes (right listing): // ช่องลำดับ frame ในการรับ; } what the fuck man.; อ่านตรงนี้

Selective Repeat Automatic Repeat Request



Piggybacking in Go-Back-N ARQ



Question

- สาเหตุที่ต้องมีการใช้งาน Protocol ในการจัดการในการส่งข้อมูล (Data link control) สำหรับ node-to-node comm. ?

  ↳ จัดระบบการส่งข้อมูล

- ความสำคัญของ Flow control คือ?

  ↳ ควบคุมการส่ง เพื่อป้องกัน buffer เต็ม

- ความสำคัญของ Error control คือ?

  arq → การส่งใหม่ กรณี data lost

Protocols comparison

Protocol	Flow control	Error control	Sender	Receiver
Simplest	✗	✗ noisless	✗	✗
Stop-and-Wait	✓	✗ channel	✓ wait ack	✓ send ack
Stop-and-Wait ARQ	✓	✓	✓ sequence number (0,1) timeout wait ack	✓ send ack (0,1)
Go-Back-N ARQ	✓	✓	✓ sequence number window < 2^m timeout เลื่อน win เมื่อรับ ack reset timeout เมื่อ ack=เลขที่ส่ง	✓ window = 1 send ack
Selective Repeat ARQ	✓	✓	✓ sequence number window = 2^{m-1} timeout ต่อ frame เลื่อน win เมื่อรับ ack เมื่อ nack, timeout ส่งใหม่ เมื่อ ackNo = Sn reset timer	✓ ขนาด 2^{m-1} เลื่อน window เมื่อช่องแรกมีคนเข้าก่อน

ความแตกต่างระหว่าง stop and wait [redacted] , Go backN [redacted] , Selective repeat [redacted]

- พวกนี้คือ flow control protocol
- stop-and-wait
  - send packet and wait for ack. Once the ack reaches the sender, it transmit the next packet. If ack is not received, it retransmits the previous packet.
- Go Back N
  - the sender send N packets which equal to the window size. Then, the sender wait for a cumulative ack to sent more packets. The receiver receive only
    - ที่ R_n ของฝั่งรับเพิ่มขึ้นเรื่อยๆ

    in-order packets. As in case of packet loss, the entire window would be re-transmitted.
- Selective Repeat
  - sender sends packet of window size and the receiver acknowledge all packet wheter they were received in order or not. In this case the receiver maintain a buffer to contain out of order packets and sort them. The sender selective retransmits the lost packet and move window forward