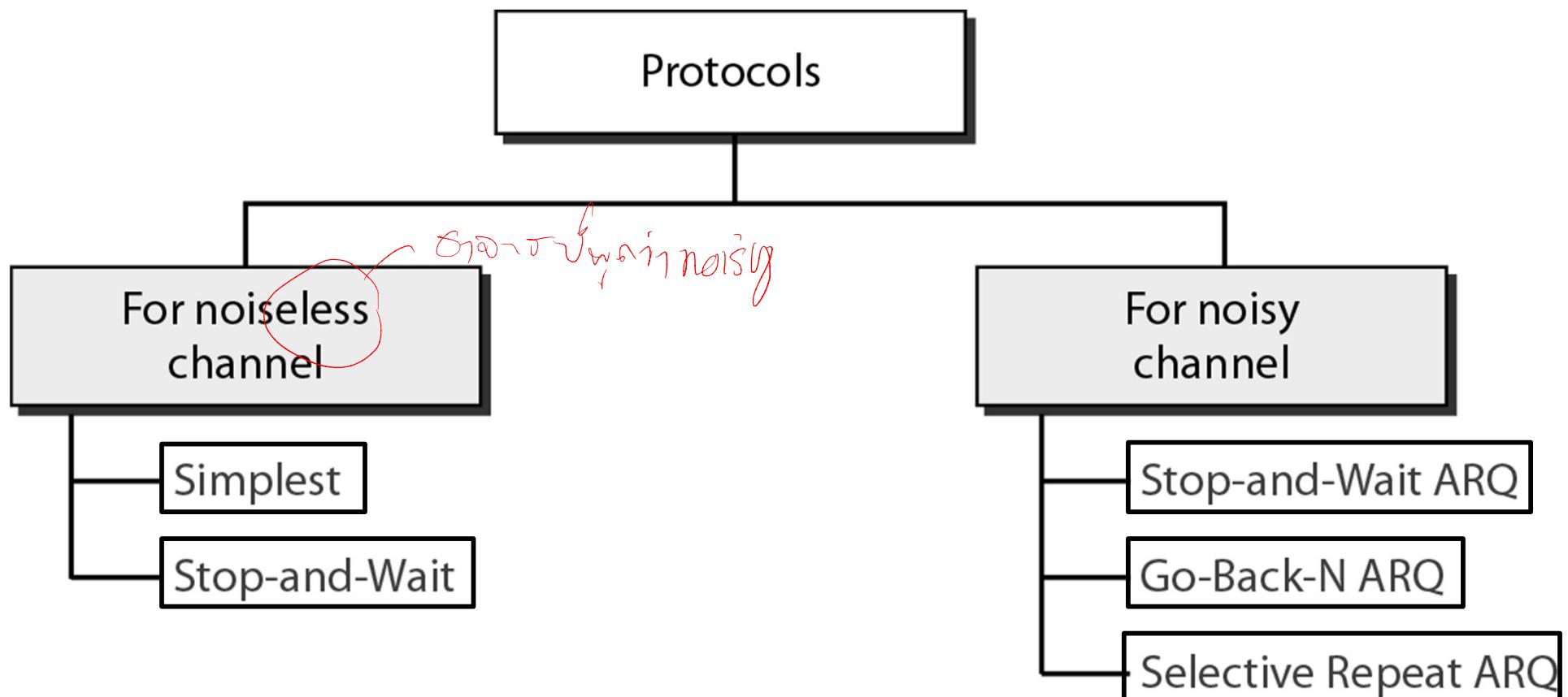
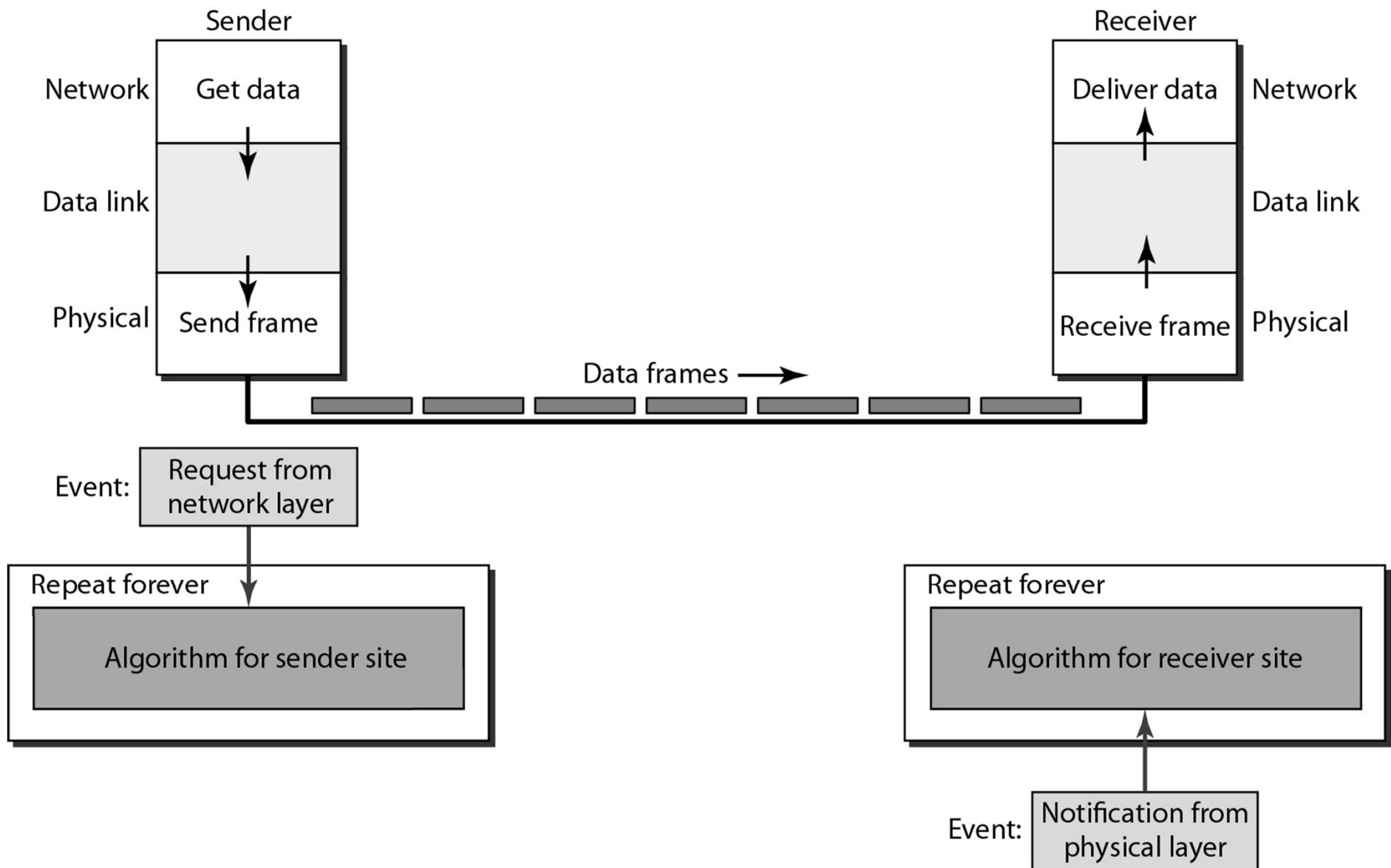


Protocols

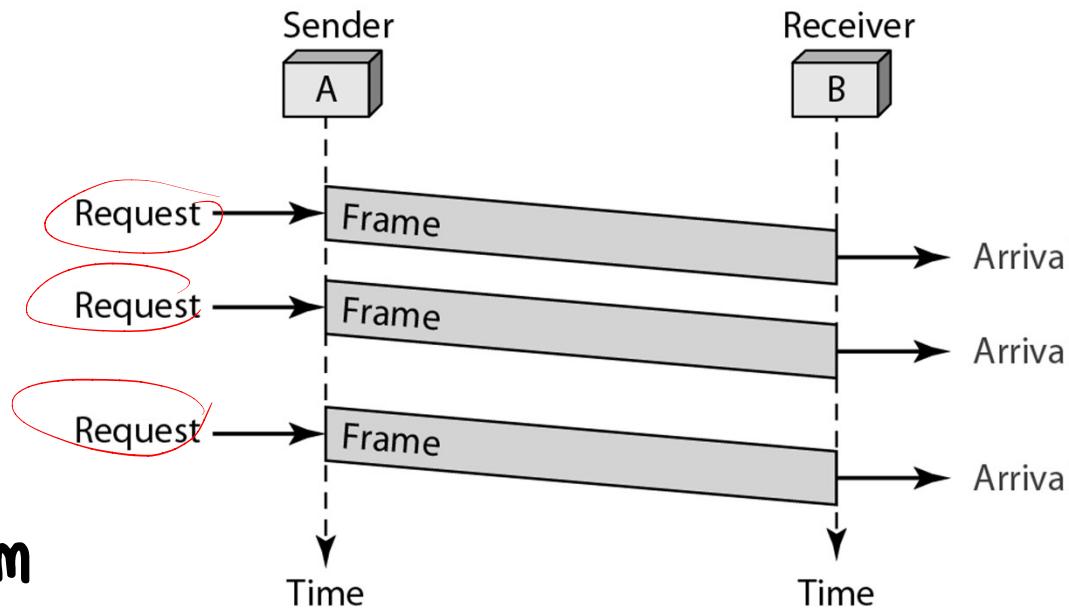


Simplest Protocol



Simplest Protocol

- Flow diagram



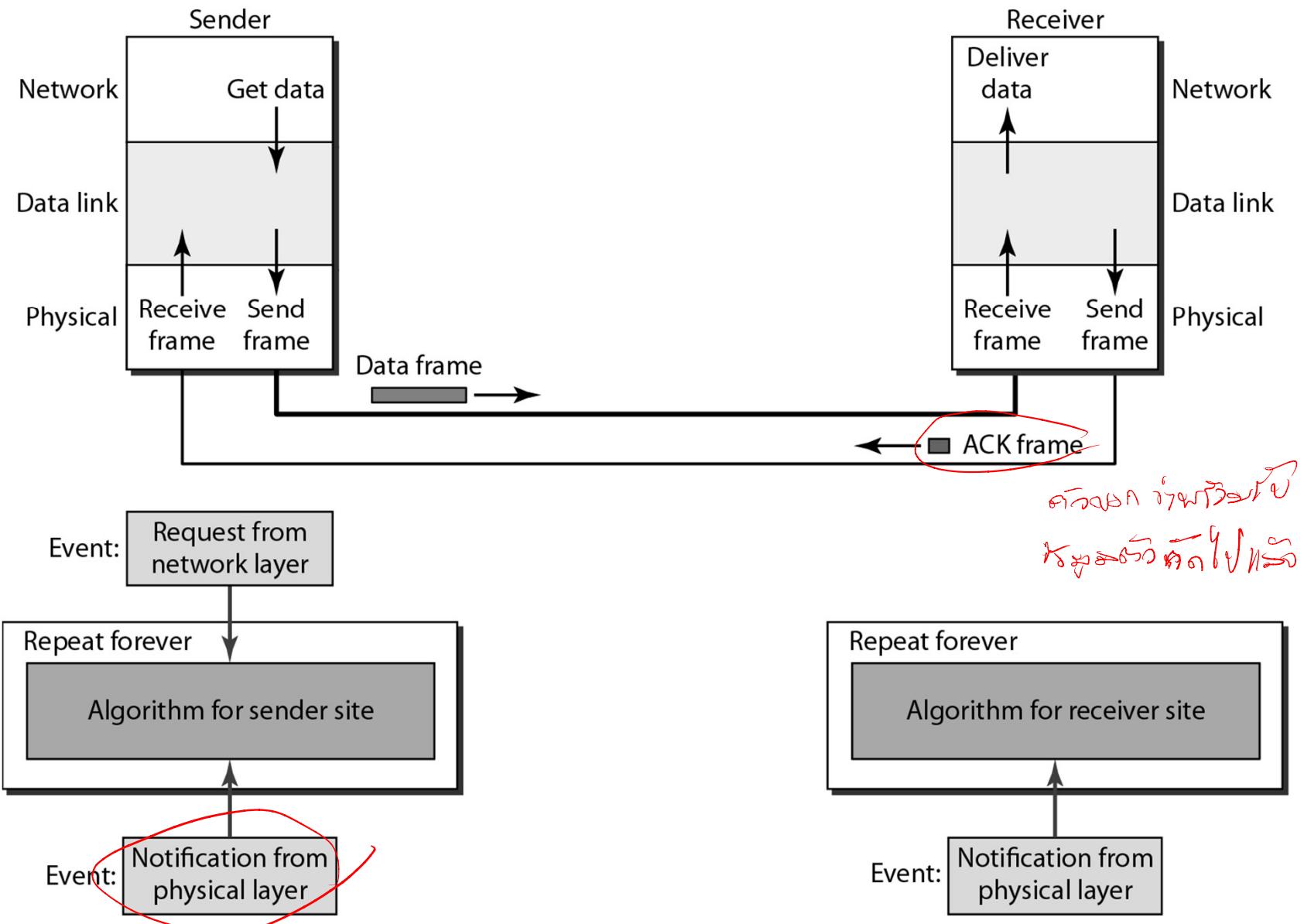
- Algorithm

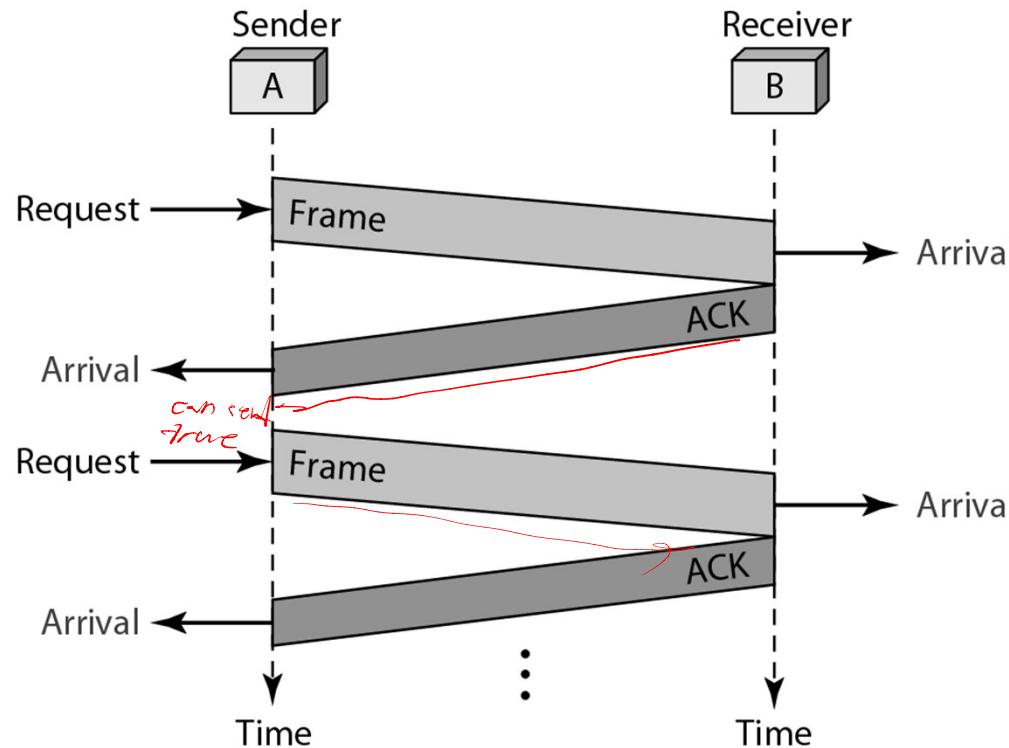
```
1 while(true)           // Repeat forever
2 {
3   WaitForEvent();     // Sleep until an event occurs
4   if(Event(RequestToSend)) //There is a packet to send
5   {
6     GetData();
7     MakeFrame();
8     SendFrame();      //Send the frame
9   }
10 }
```

```
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

స్టాప-ఎండ్-వైట్ పోర్కోల్





```

2 canSend = true           //Allow the first frame to go
1 while(true)              //Repeat forever
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 }
```

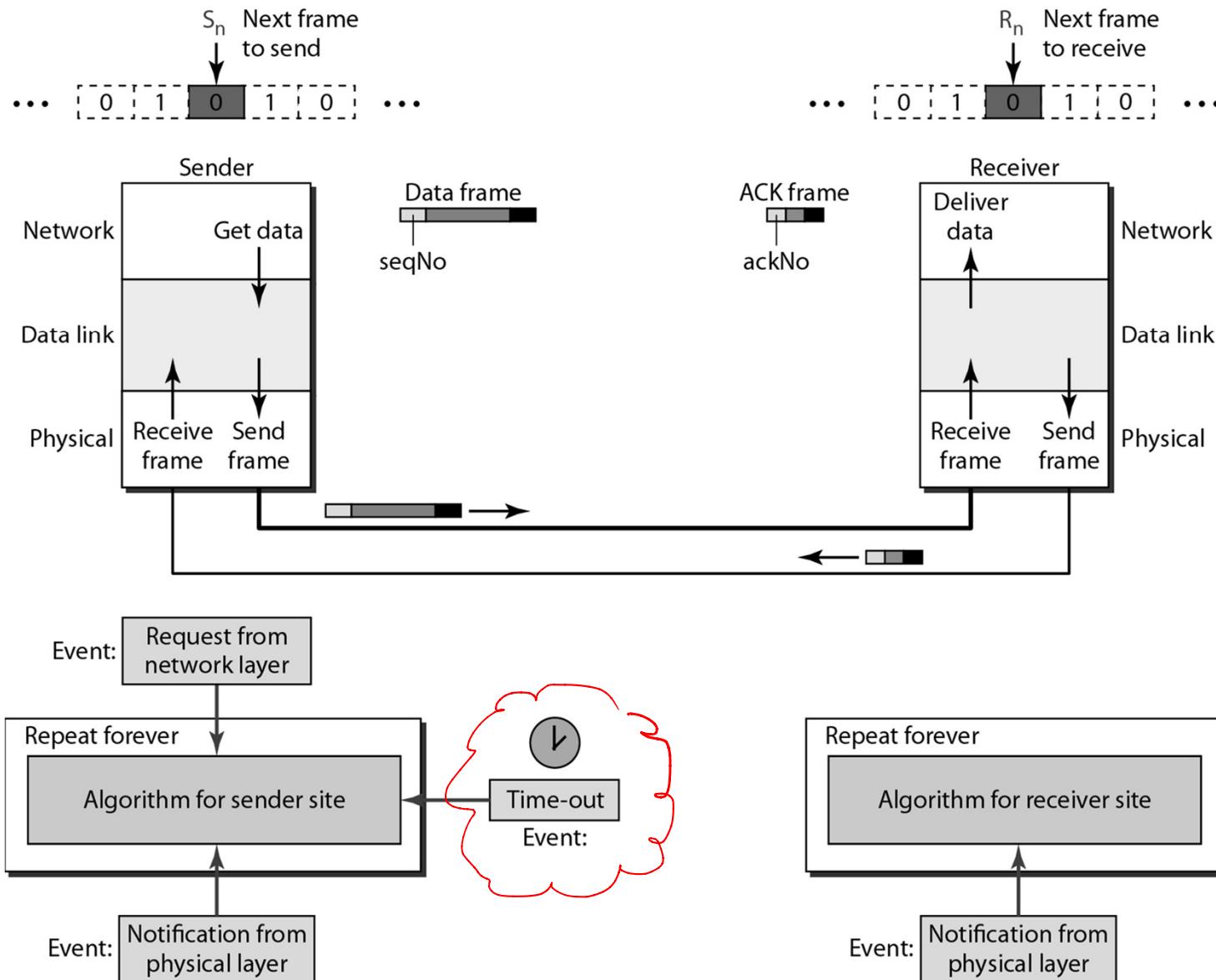
```

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

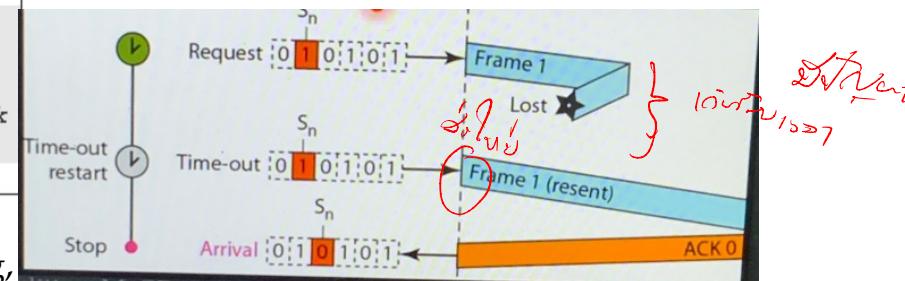
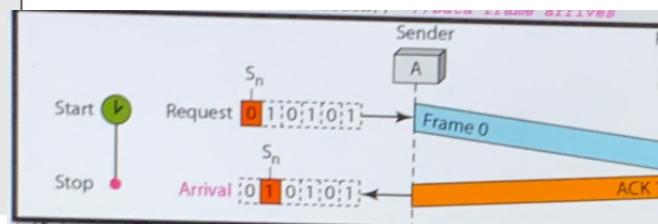
```

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();                  //Resend a copy check
31    ResendFrame(Sn-1);
32  }
33 }
```



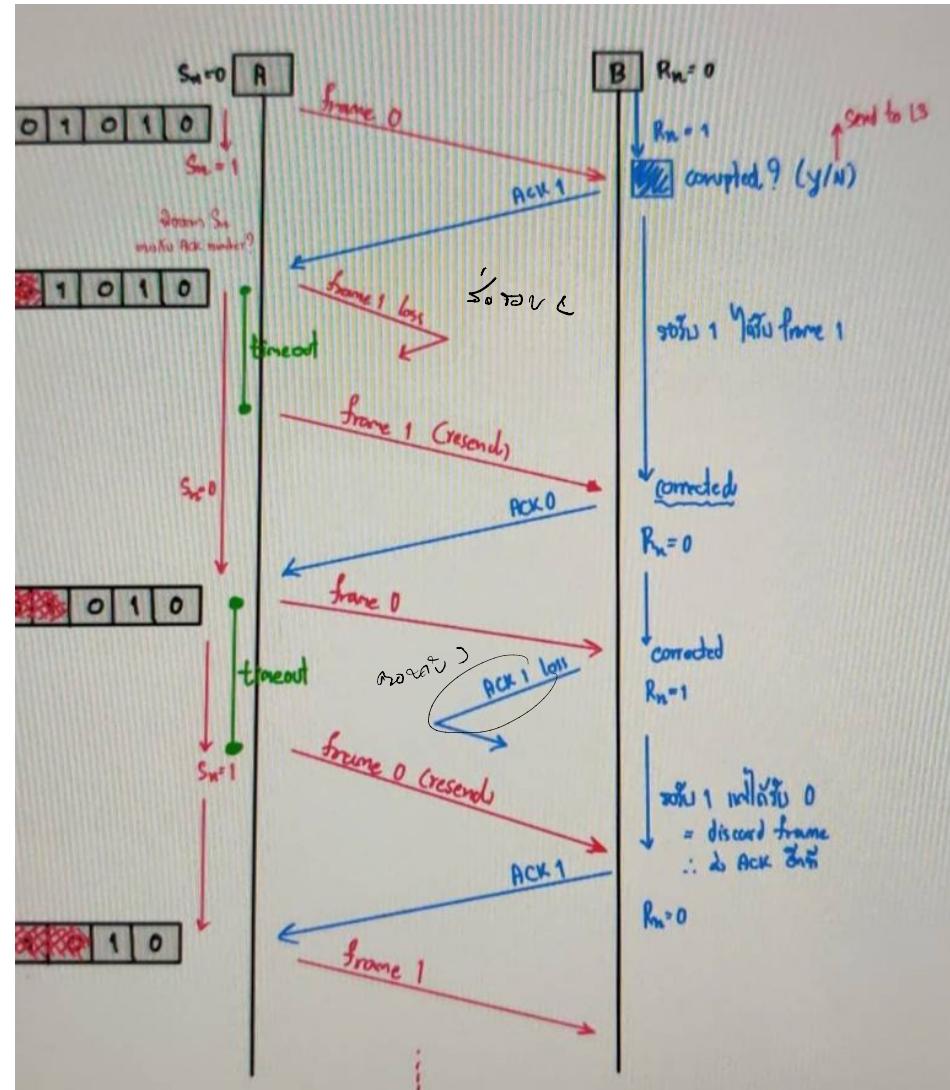
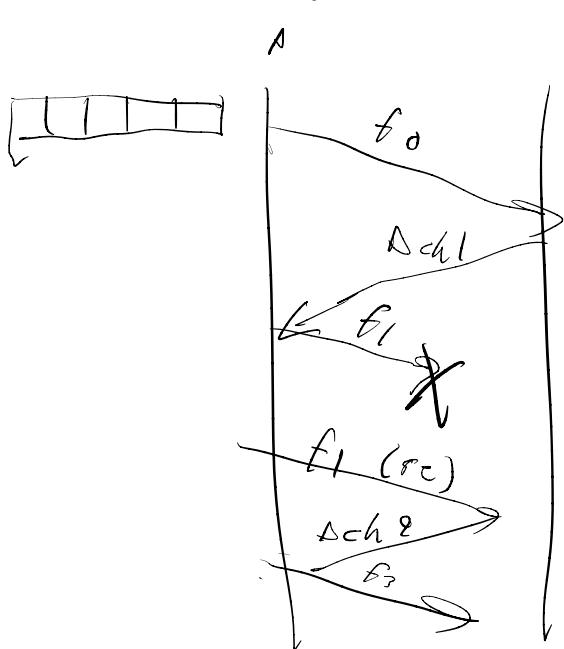
```

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       sleep();
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 }
```

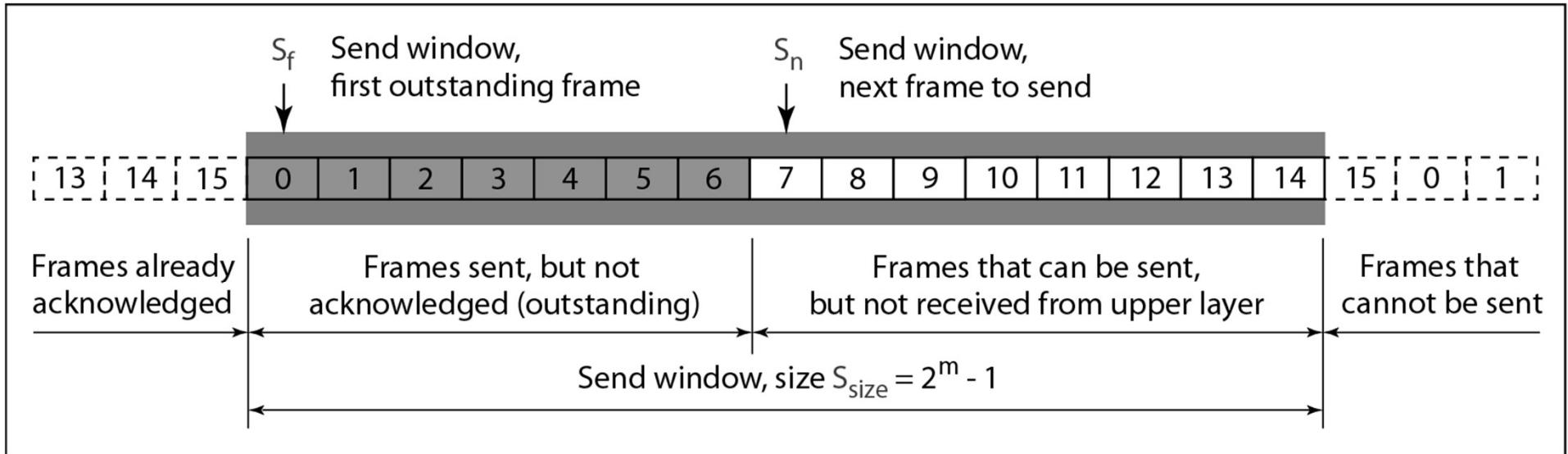


Stop-and-Wait Automatic Repeat Request

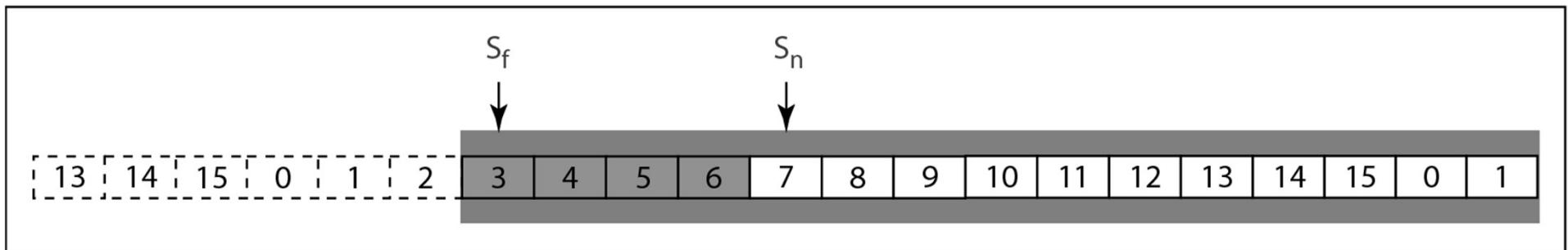
- Flow diagram
 - Send 5
 - Data 2nd lost
 - ACK 3th lost



Go-Back-N Automatic Repeat Request



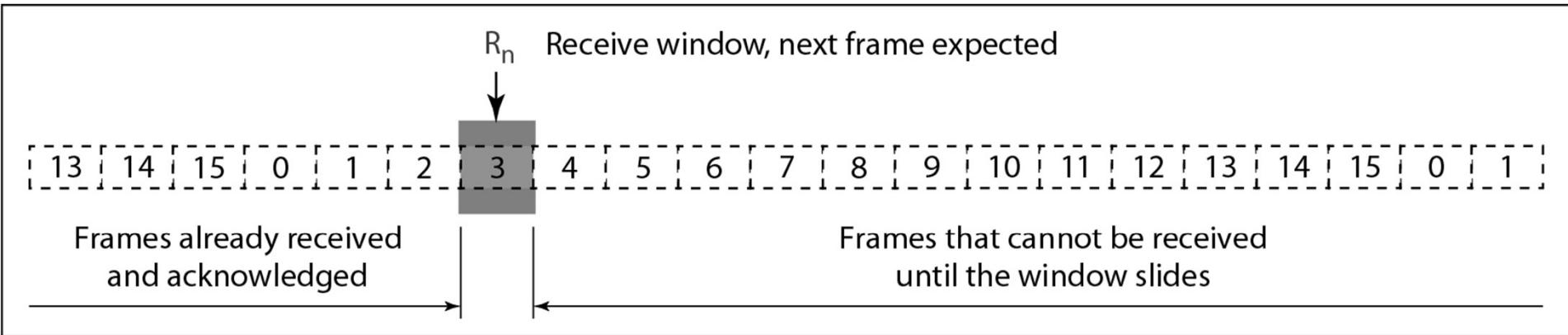
a. Send window before sliding



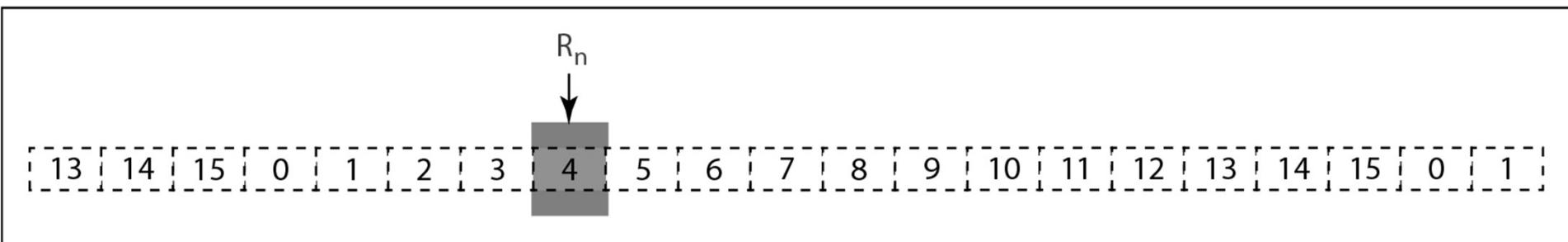
b. Send window after sliding

- 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

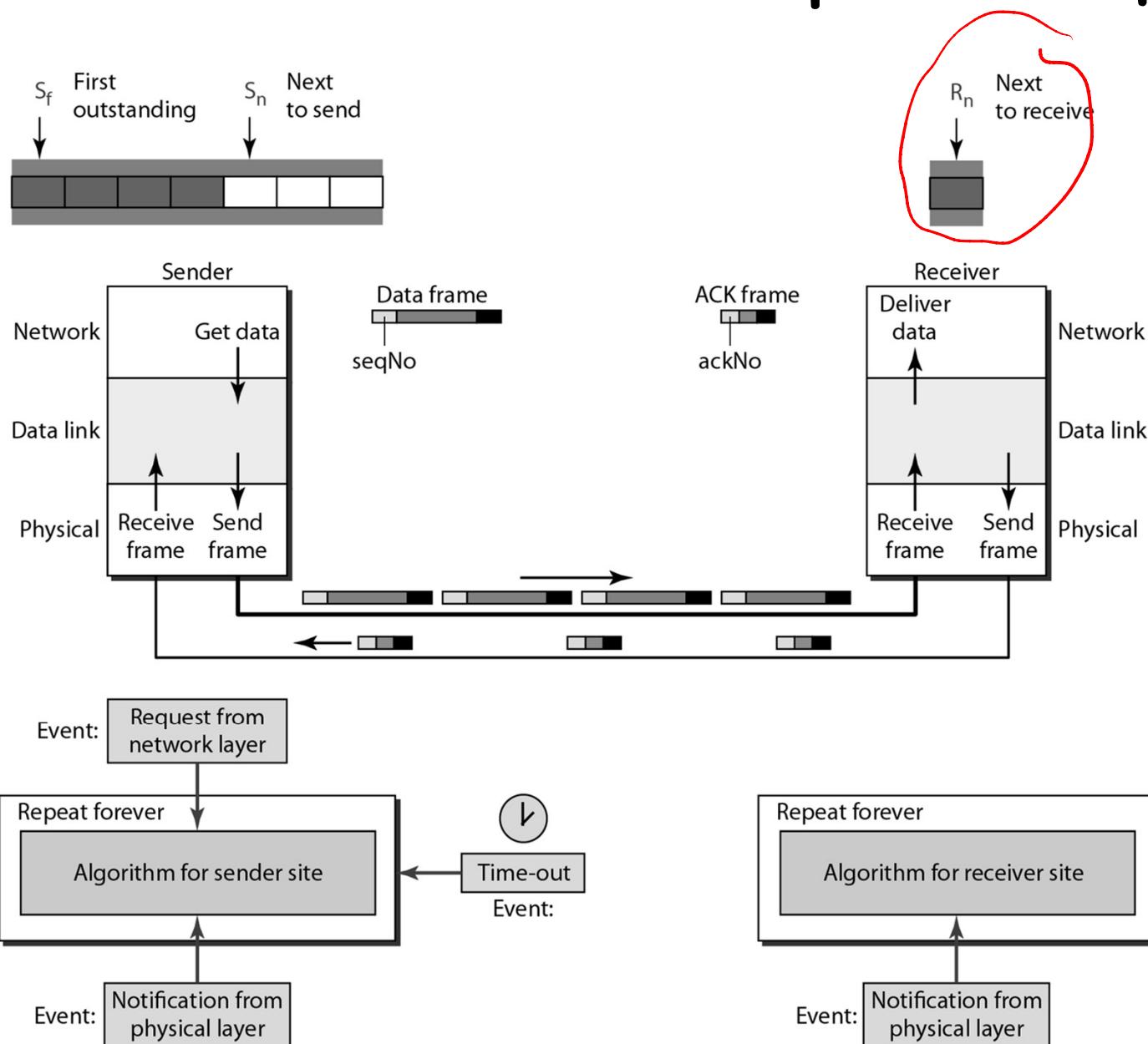


a. Receive window

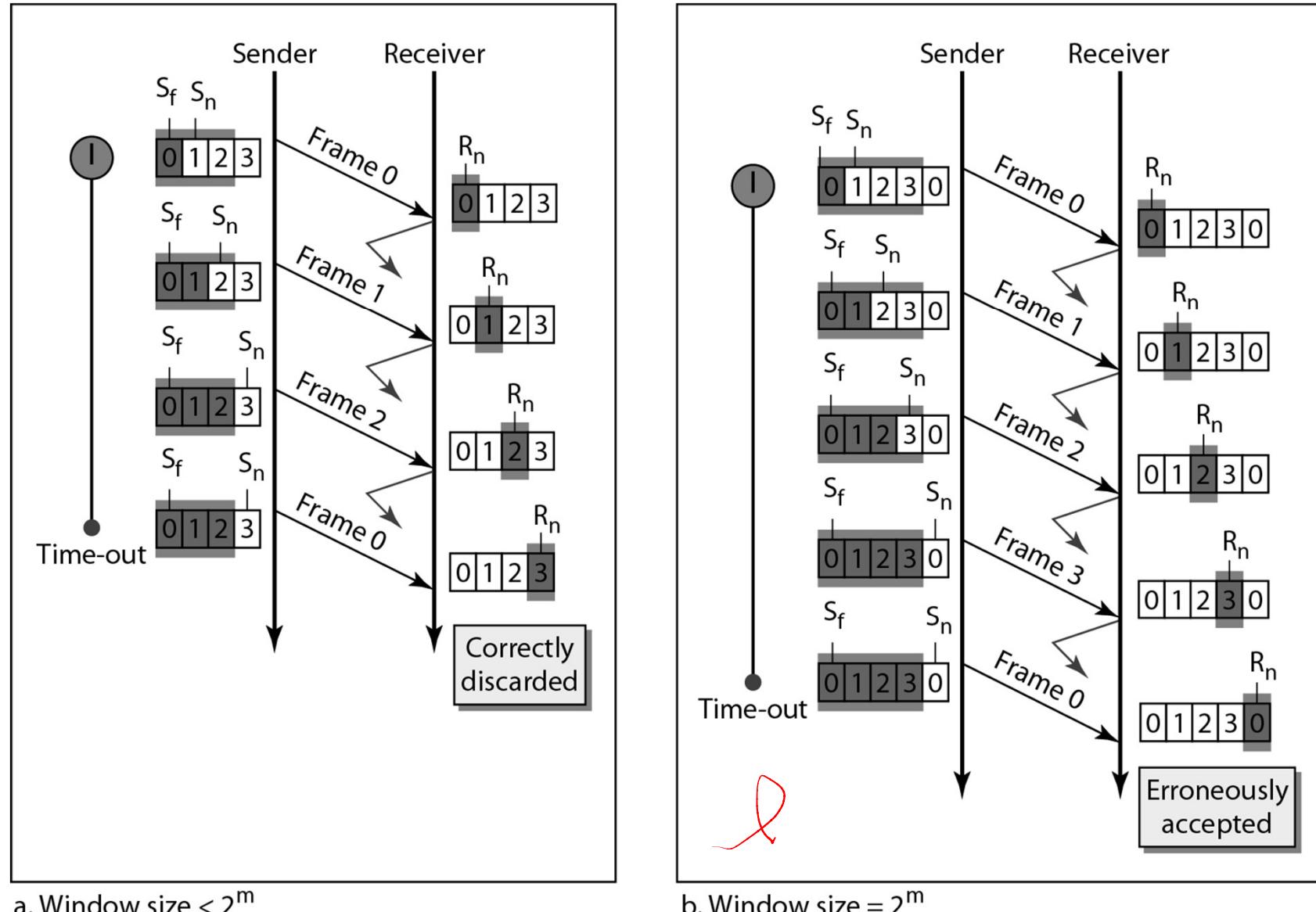


b. Window after sliding

Go-Back-N Automatic Repeat Request



Go-Back-N Automatic Repeat Request



```

1 Sw = 2m - 1;
2 Sf = 0;
3 Sn = 0;
4
5 while (true) //Repeat forever
6 {
7   WaitForEvent(); //Event
8   if(Event(RequestToSend)) //A packet to send
9   {
10     if(Sn-Sf >= Sw) //If window is full
11       Sleep();
12     GetData();
13     MakeFrame(Sn);
14     StoreFrame(Sn);
15     SendFrame(Sn);
16     Sn = Sn + 1; //if(timer not running)
17     StartTimer();
18   }
19 }

20 ACK //ACK arrives
21 if(Event(ArrivalNotification)) //ACK arrives
22 {
23   Receive(ACK);
24   if(corrupted(ACK))
25     Sleep();
26   if((ackNo>Sf)&&(ackNo<=Sn)) //If a valid ACK
27   While(Sf <= ackNo)
28   {
29     PurgeFrame(Sf);
30     Sf = Sf + 1;
31   }
32   StopTimer();
33 }

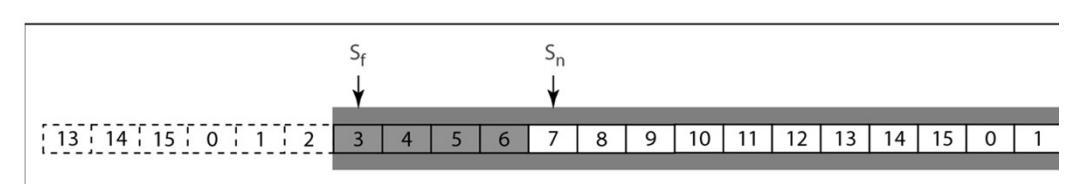
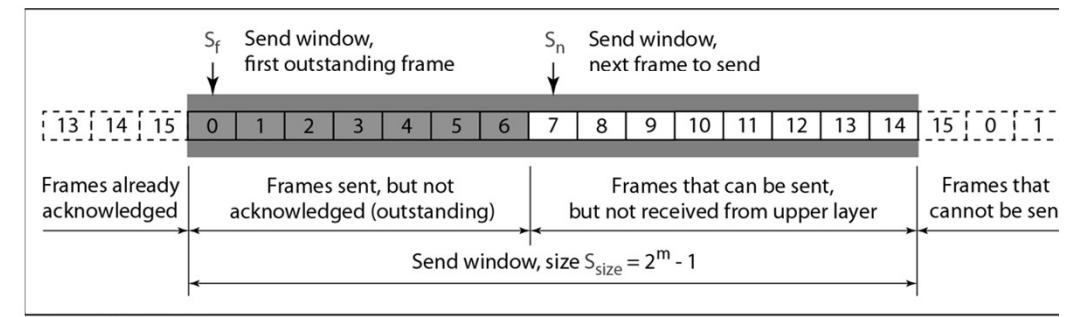
34 //Timeout
35 if(Event(TimeOut)) //The timer expires
36 {
37   StartTimer(); //Event
38   Temp = Sf;
39   while(Temp < Sn);
40   {
41     SendFrame(Temp);
42     Temp = Temp + 1;
43   }
44 }
45 }

```

```

1 Rn = 0;
2
3 while (true) //Repeat forever
4 {
5   WaitForEvent();
6
7   if(Event(ArrivalNotification)) //Data frame arrives
8   {
9     Receive(Frame); //Event
10    if(corrupted(Frame))
11      Sleep();
12    if(seqNo == Rn) //If expected frame
13    {
14      DeliverData(); //Deliver data
15      Rn = Rn + 1; //Slide window
16      SendACK(Rn);
17    }
18 }
19 }

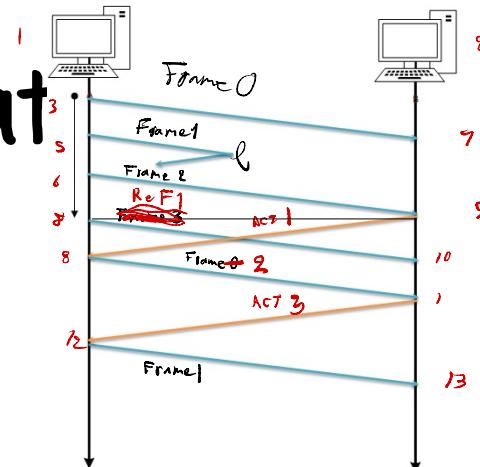
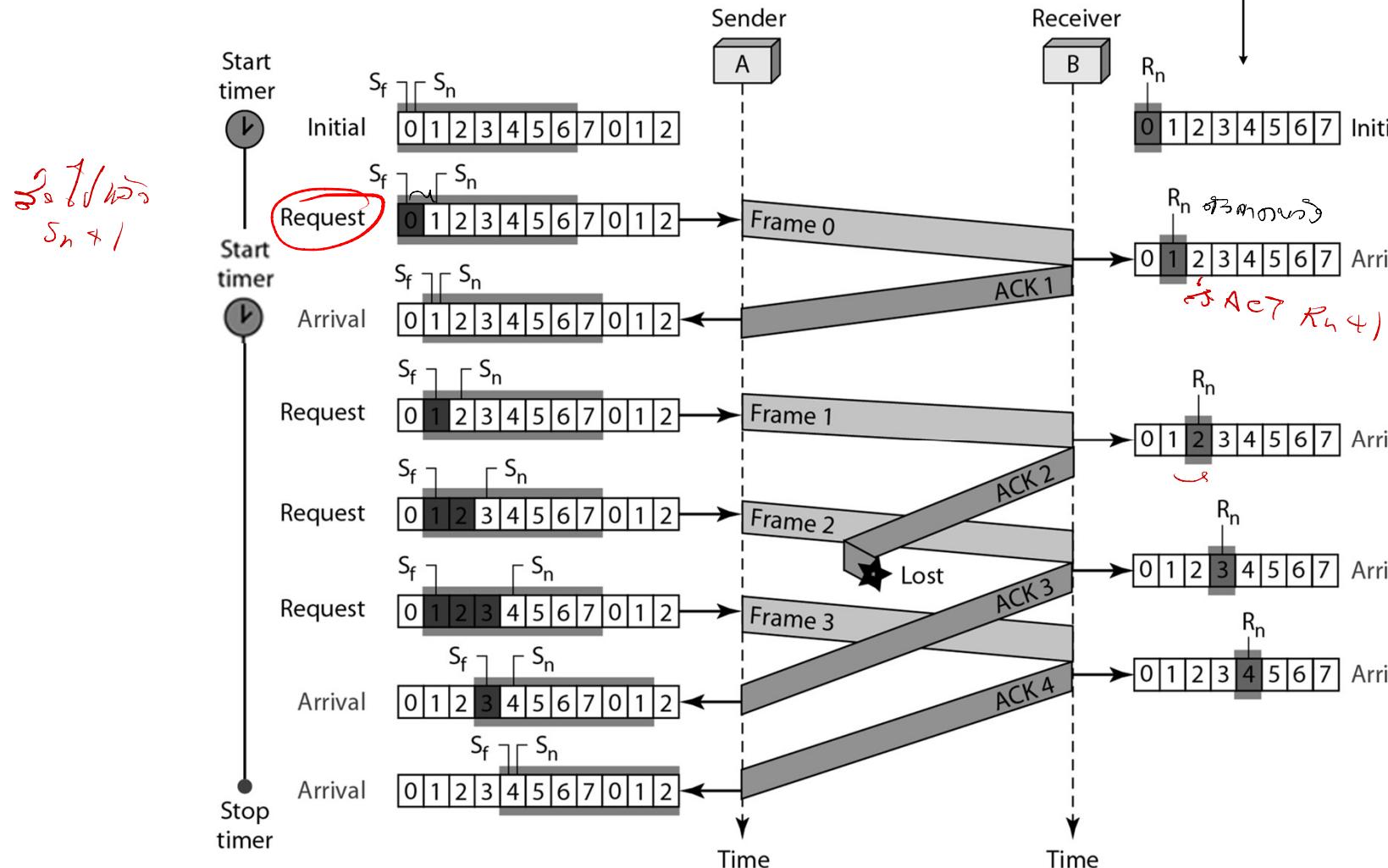
```



Go-Back-N Automatic Repeat

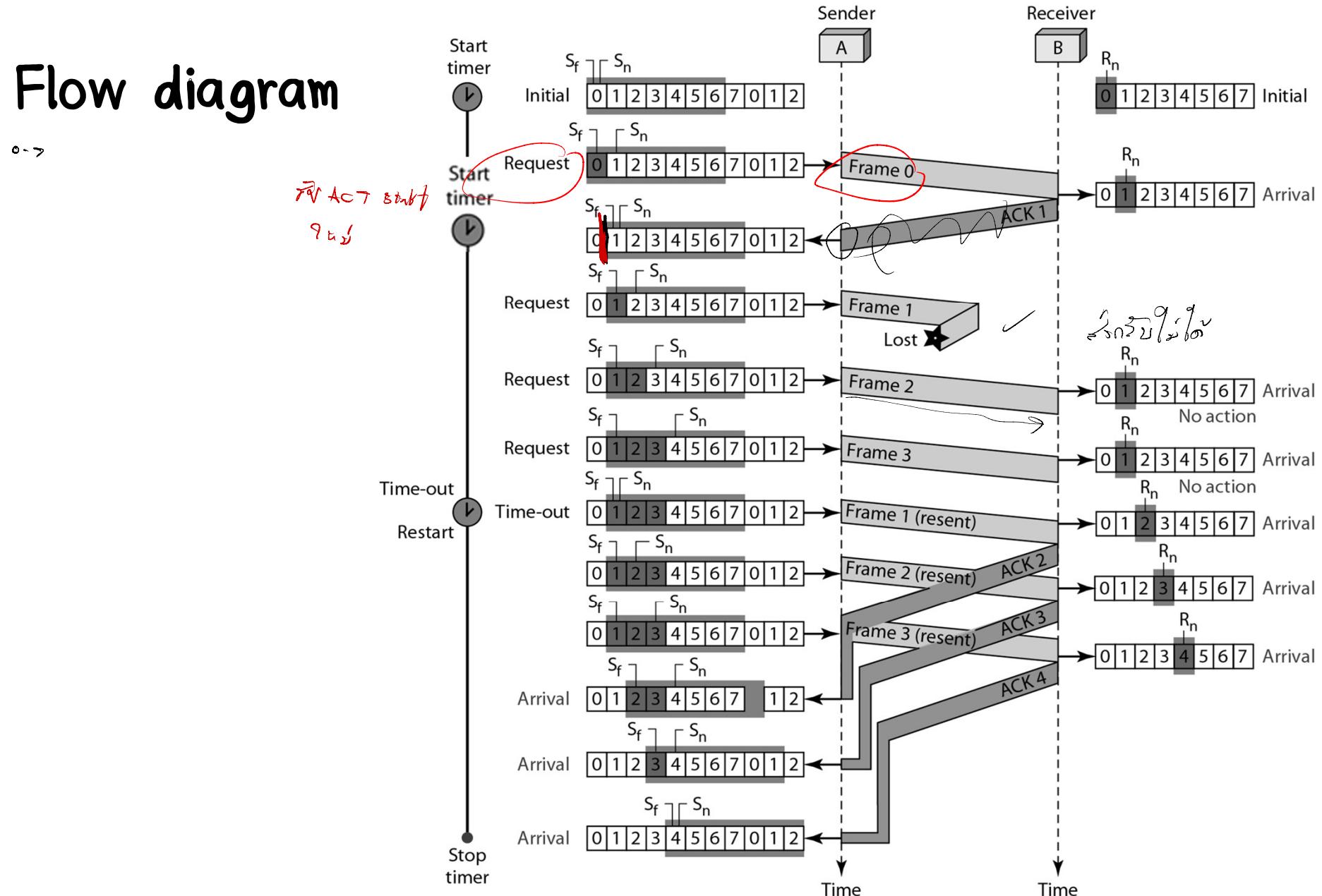
$m=3$

- Flow diagram



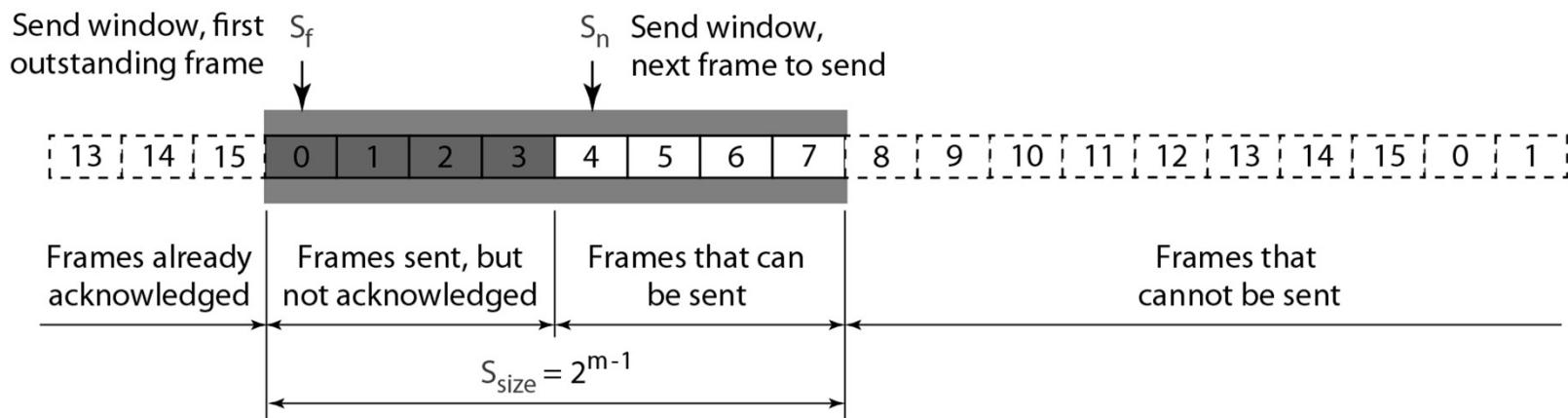
Go-Back-N Automatic Repeat Request

- Flow diagram

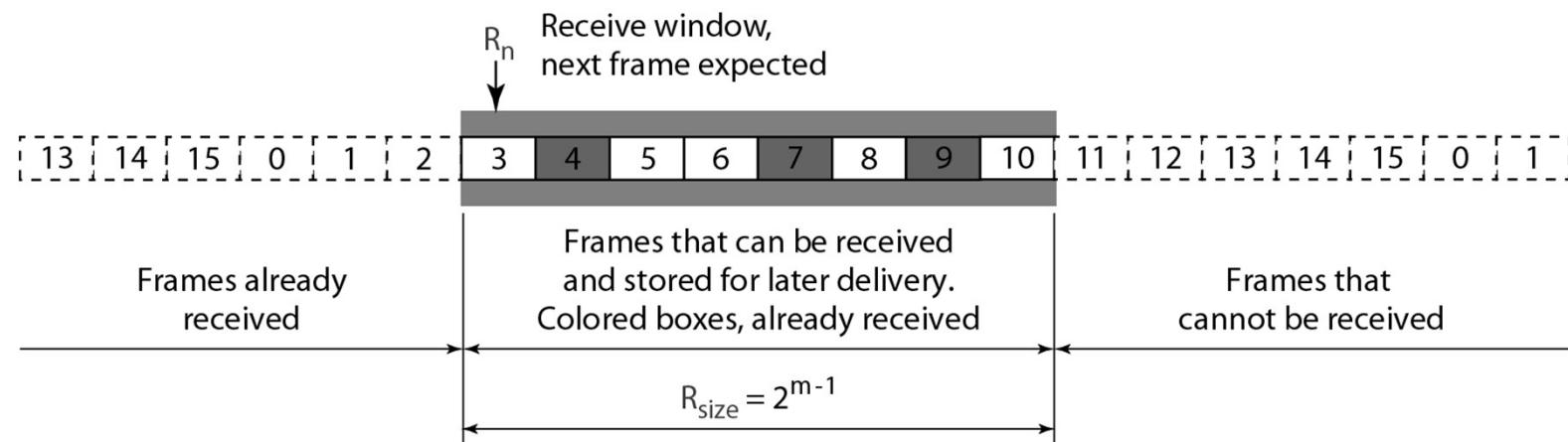


Selective Repeat Automatic Repeat Request

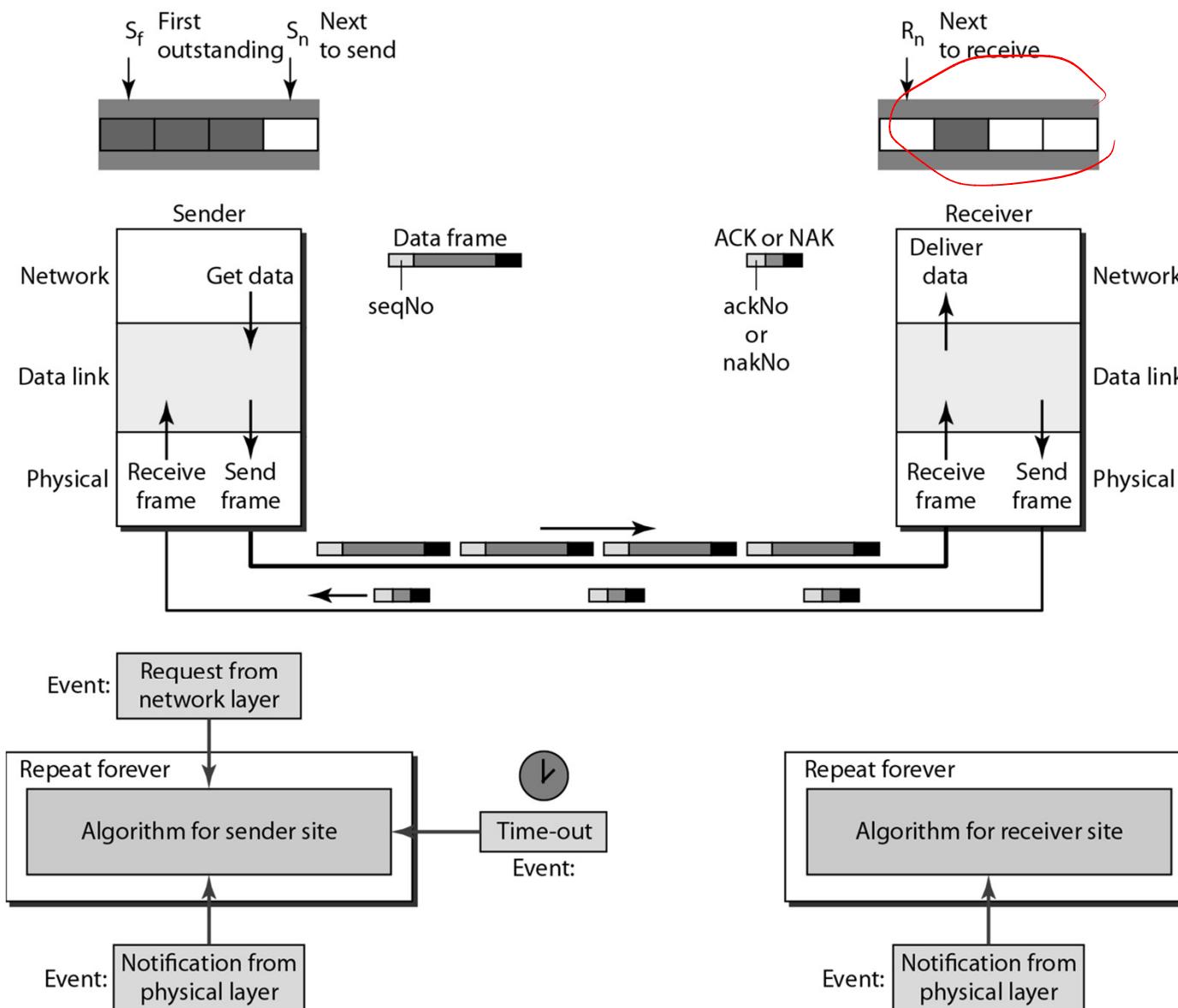
- Send window



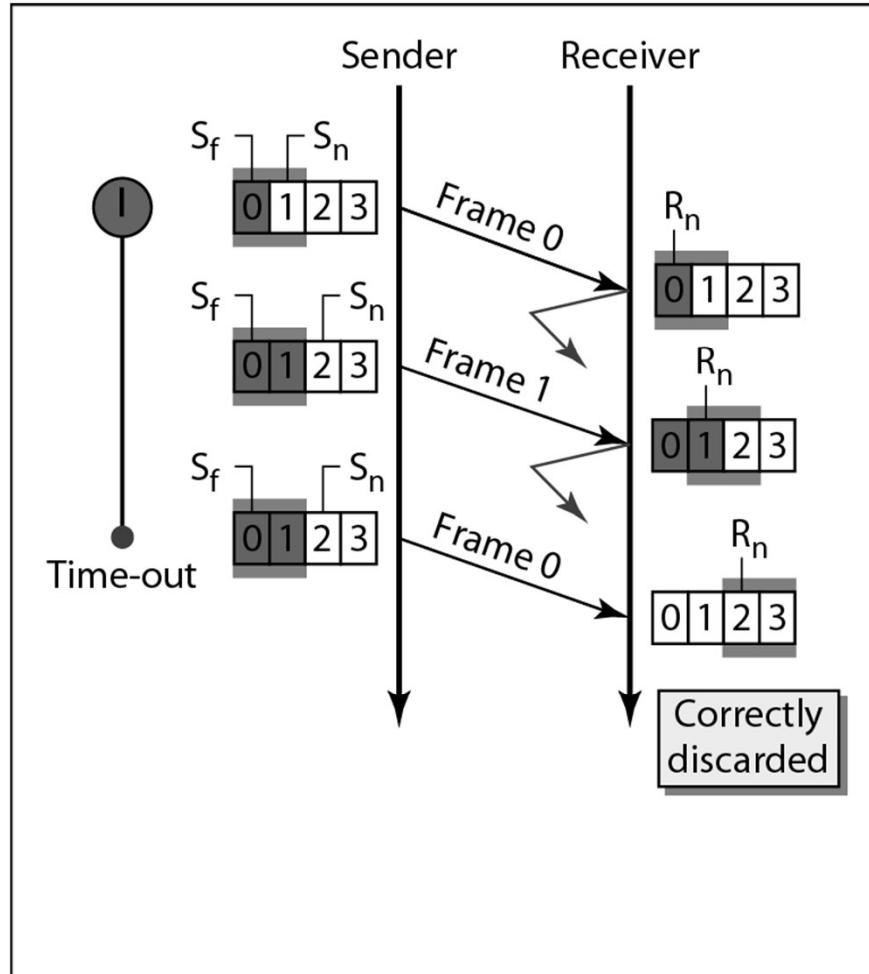
- Receive window



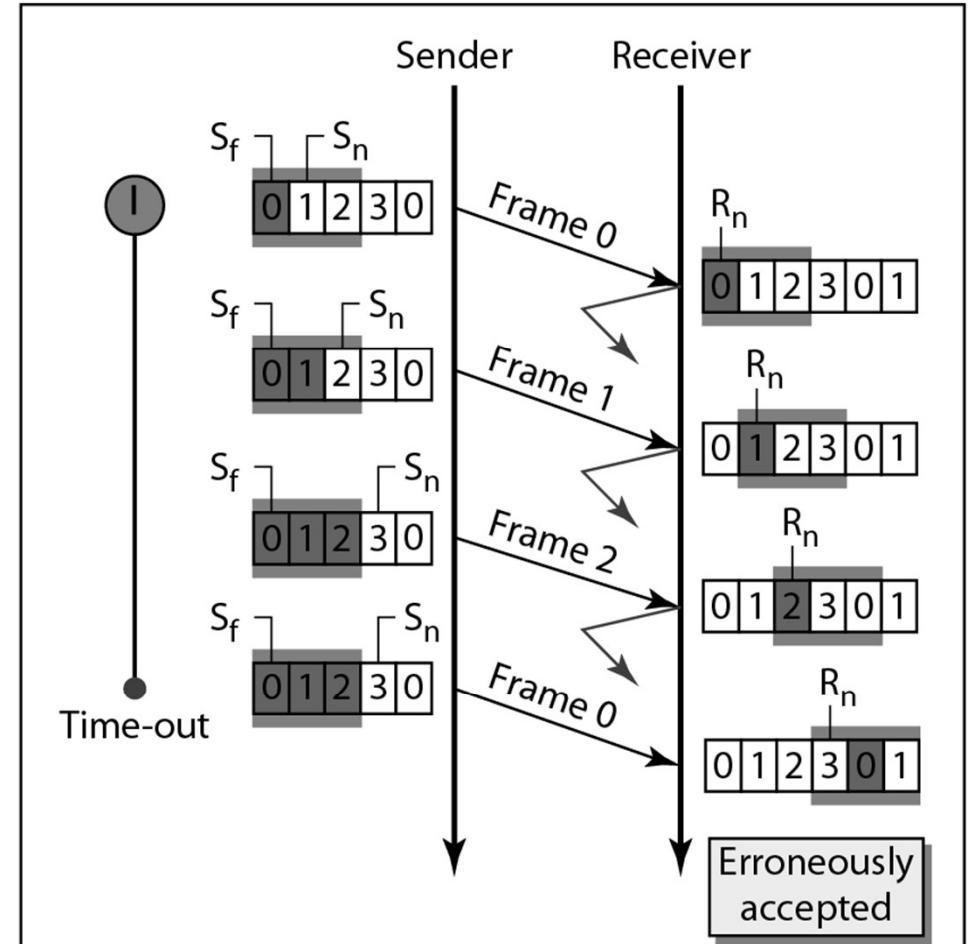
Selective Repeat Automatic Repeat Request



Selective Repeat Automatic Repeat Request



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



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

```

1 Sw = 2m-1 ;
2 Sf = 0;
3 Sn = 0;
4
5 while (true) //Repeat forever
6 {
7     WaitForEvent();
8     if(Event(RequestToSend)) //There is a packet to send
9     {
10         if(Sn-Sf >= Sw) //If window is full
11             Sleep();
12         GetData();
13         MakeFrame(Sn);
14         StoreFrame(Sn);
15         SendFrame(Sn);
16         Sn = Sn + 1;
17         StartTimer(Sn);
18     }
19
20     if(Event(ArrivalNotification)) //ACK arrives
21     {
22         Receive(frame); //Receive ACK or NAK
23         if(corrupted(frame))
24             Sleep();
25         if (FrameType == NAK)
26             if (nakNo between Sf and Sn)
27             {
28                 resend(nakNo);
29                 StartTimer(nakNo);
30             }
31         if (FrameType == ACK)
32             if (ackNo between Sf and Sn)
33             {
34                 while(sf < ackNo)
35                 {
36                     Purge(sf);
37                     StopTimer(sf);
38                     Sf = Sf + 1;
39                 }
40             }
41     }
42
43     if(Event(TimeOut(t))) //The timer expires
44     {
45         StartTimer(t);
46         SendFrame(t);
47     }
48 }

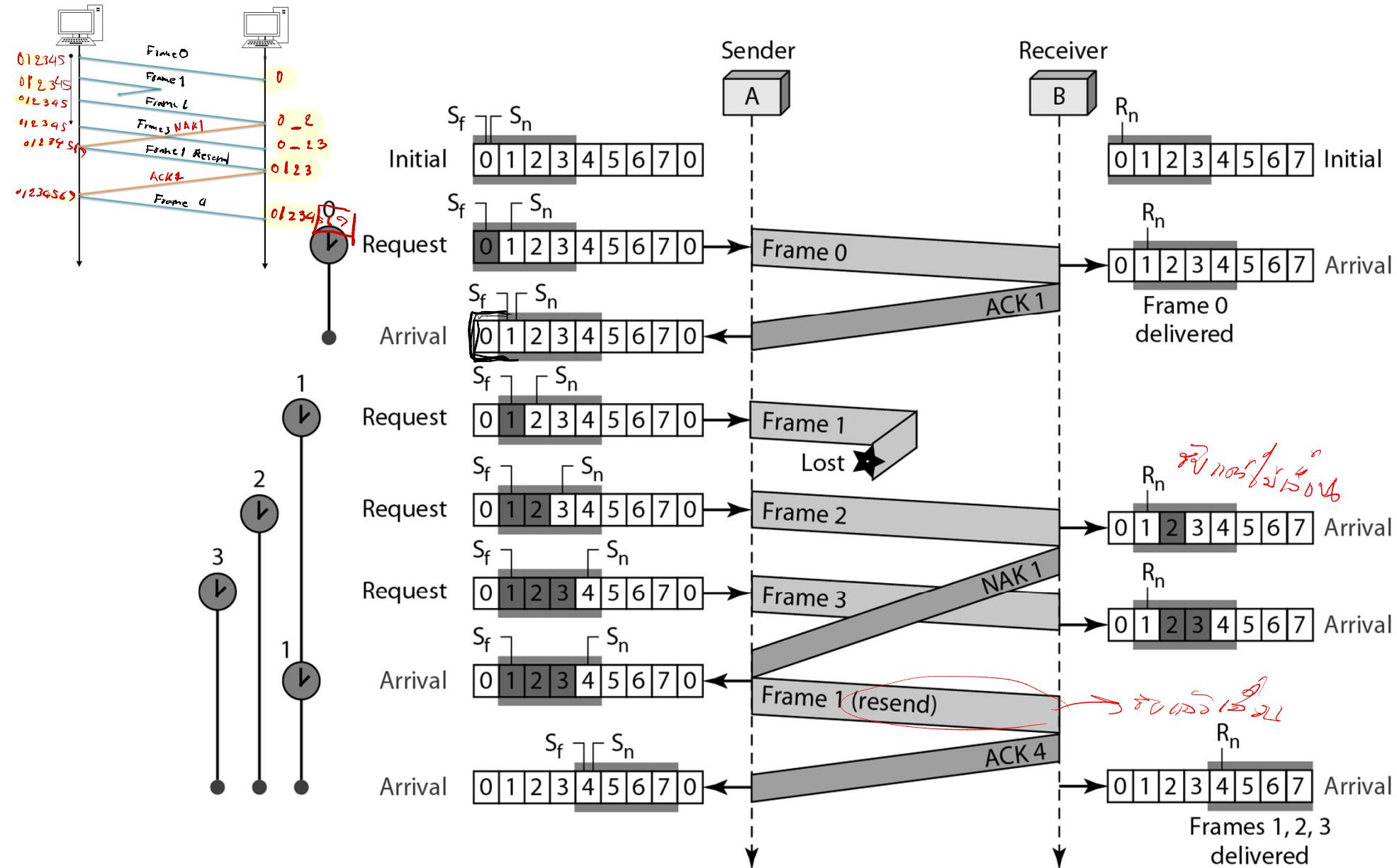
```

```

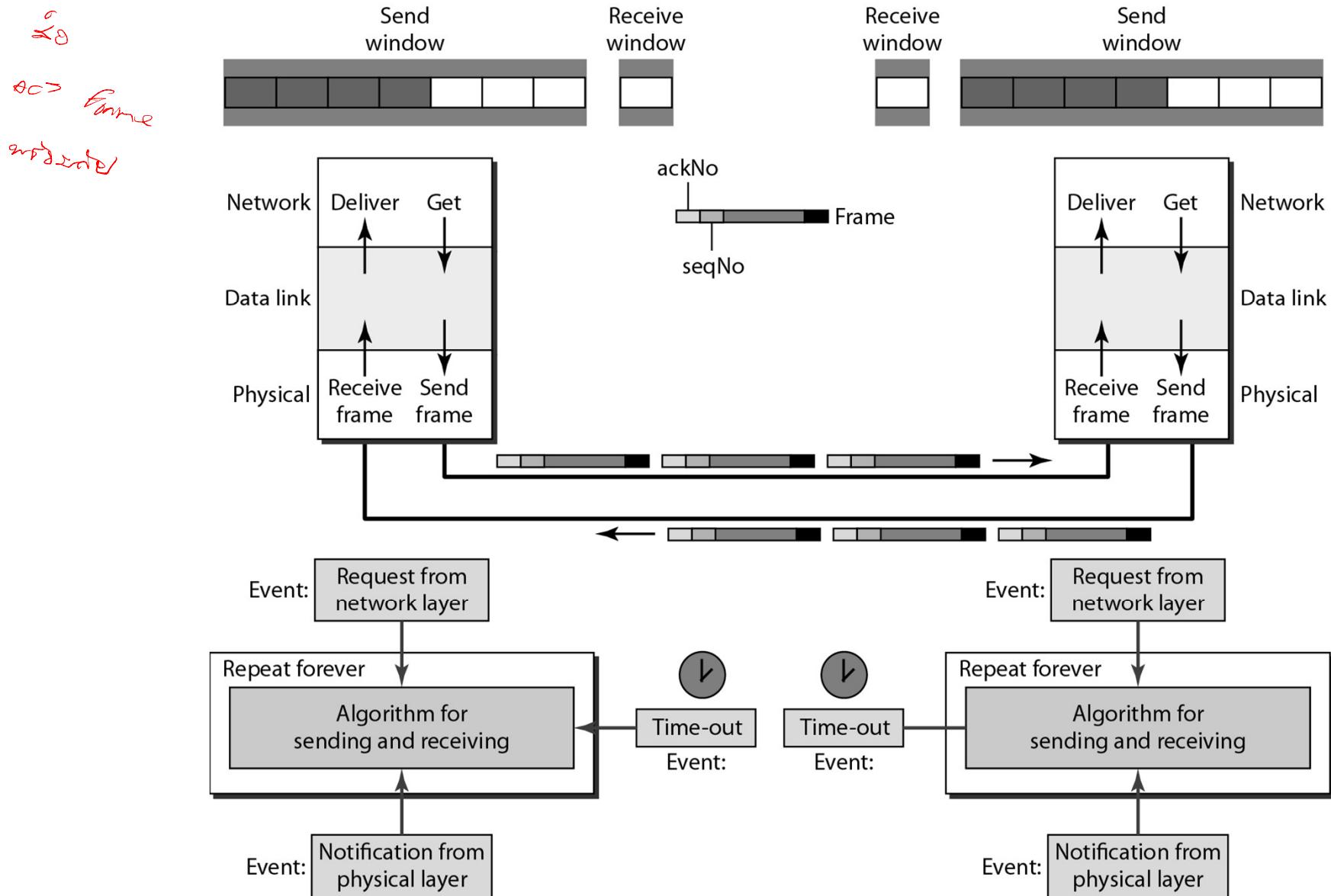
1 Rn = 0;
2 NakSent = false;
3 AckNeeded = false;
4 Repeat(for all slots)
5     Marked(slot) = false;
6
7 while (true) //Repeat forever
8 {
9     WaitForEvent();
10
11     if(Event(ArrivalNotification)) //Data frame arrives
12     {
13         Receive(Frame);
14         if(corrupted(Frame))&& (NOT NakSent)
15         {
16             SendNAK(Rn);
17             NakSent = true;
18             Sleep();
19         }
20         if(seqNo <> Rn)&& (NOT NakSent)
21         {
22             SendNAK(Rn);
23             NakSent = true; }
24         if ((seqNo in window)&& (!Marked(seqNo)))
25         {
26             StoreFrame(seqNo)
27             Marked(seqNo)= true;
28             while(Marked(Rn))
29             {
30                 DeliverData(Rn);
31                 Purge(Rn);
32                 Rn = Rn + 1;
33                 AckNeeded = true;
34             }
35             if(AckNeeded);
36             {
37                 SendAck(Rn);
38                 AckNeeded = false;
39                 NakSent = false;
40             }
41         }
42     }
43 }
44 }

```

Selective Repeat Automatic Repeat Request



Piggybacking in Go-Back-N ARQ



Question

- สาเหตุที่ต้องมีการใช้งาน Protocol ในการจัดการในการส่งข้อมูล (Data link control) สำหรับ node-to-node comm. ?
- ความสำคัญของ Flow control คือ?
- ความสำคัญของ Error control คือ?

Protocols comparison

Protocol	Flow control	Error control	Sender	Receiver
Simplest	✗	✗	-	-
Stop-and-Wait	✓		Wait ACK	send ACK
Stop-and-Wait ARQ	✓	✓	+ sequence No. 0 Time Out	ACK
Go-Back-N ARQ	✓	✓	Sequence No. 2^n Send window $2^m - 1$	
Selective Repeat ARQ	✓	✓	S window 2^{n-1} \Rightarrow R window 2^{m-1} Time out per frame	