Data Link Control
FRAMING

• The data link layer needs to pack bits into **frames**, so that each frame is distinguishable from another.
• Our postal system practices a type of framing.
• The simple act of inserting a letter into an envelope separates one piece of information from another; the envelope serves as the delimiter.

**Topics discussed in this section:**

- Fixed-Size Framing
- Variable-Size Framing
Fixed-Size Framing: No need of defining the Boundaries of the frame.

Variable-Size Framing: LAN

- **Character-oriented protocol**—popular when only text was exchange

A frame in a character-oriented protocol

- **Bit-oriented protocol**
Byte stuffing is the process of adding 1 extra byte whenever there is a flag or escape character in the text.
Figure 11.2  Byte stuffing and unstuffing

Escape character (ESC), which has a predefined bit pattern.
Figure 11.3  A frame in a bit-oriented protocol
Bit stuffing is the process of adding one extra 0 whenever five consecutive 1s follow a 0 in the data, so that the receiver does not mistake the pattern 0111110 for a flag.
Figure 11.4 Bit stuffing and unstuffing

Data from upper layer
0001111111001111101000

Frame sent
Flag | Header | 000111110110011111001000 | Trailer | Flag

Stuffed

Extra 2 bits

Frame received
Flag | Header | 000111110110011111001000 | Trailer | Flag

Unstuffed

Data to upper layer
00011111111001111101000

11.8
The most important responsibilities of the data link layer are flow control and error control. Collectively, these functions are known as data link control.

Topics discussed in this section:

Flow Control
Error Control
Flow control refers to a set of procedures used to restrict the amount of data that the sender can send before waiting for acknowledgment.

*Note*
Error control in the data link layer is based on automatic repeat request (ARQ), which is the retransmission of data.
11-3 PROTOCOLS

Now let us see how the data link layer can combine framing, flow control, and error control to achieve the delivery of data from one node to another.

The protocols are normally implemented in software by using one of the common programming languages.
Figure 11.5  Taxonomy of protocols discussed in this chapter
Let us first assume we have an ideal channel in which no frames are lost, duplicated, or corrupted. We introduce two protocols for this type of channel.

**Topics discussed in this section:**
Simplest Protocol
Stop-and-Wait Protocol
Figure 11.6  The design of the simplest protocol with no flow or error control
The sender sends a sequence of frames without even thinking about the receiver.
Figure 11.8 Design of Stop-and-Wait Protocol
The sender sends one frame and waits for feedback from the receiver. When the ACK arrives, the sender sends the next frame.
Although the Stop-and-Wait Protocol gives us an idea of how to add flow control to its predecessor, noiseless channels are nonexistent. We discuss three protocols in this section that use error control.

Topics discussed in this section:
- Stop-and-Wait Automatic Repeat Request
- Go-Back-N Automatic Repeat Request
- Selective Repeat Automatic Repeat Request
Error control in the data link layer is based on automatic repeat request, which is the retransmission of data.

We shall be doing three flow and error control protocols:

# Stop-and-Wait ARQ

# Go-Back-N ARQ

# Selective Repeat (ARQ: Automatic Repeat Request)
Stop-and-Wait ARQ

**Operation**

**Bidirectional Transmission**
# The sending device keeps a copy of the last frame transmitted until it receives an acknowledgement for that frame.

# Both, data frames and acknowledgement (ACK) frames, are numbered alternatively 0 and 1.

# A data frame 0 is acknowledged by an ACK 1 frame, indicating that the receiver has received data frame 0 and is now expecting data frame 1.

The sender has a control variable, $S$, that holds the number of the recently sent frame (0 or 1).

The receiver has a control variable, $R$, that holds the number of the next frame expected.
When the receiver receives a damaged frame, it discards it, which essentially means a frame is lost.

The receiver remains silent about a lost frame and keeps its value of $R$.

After the timer at the sender site expires, another copy of frame 1 is sent.
If the sender receives a damaged acknowledgement, it discards it.

When the timer for frame expires, the sender retransmits the last transmitted frame.

As the receiver has already received this particular frame, it silently discards it and resend the acknowledgement previously sent.
In Stop-and-Wait ARQ, numbering frames prevents the retaining of duplicate frames.
Reasons: An ACK can be delayed at the receiver or by some problem with the link.

Figure below shows the operation
Until now, we have assumed unidirectional transmission. But transmission can be bidirectional if the two parties have two separate channels for full-duplex transmission or share the same channel for half-duplex transmission.

**Piggybacking**: It is a method to combine a data frame with an acknowledgement.

---

### Diagram

**Nodes**
- **A**: Source
- **B**: Destination

**Channels**
- Full-duplex or half-duplex

**Frames and acknowledgments**
- **Frame 0, ACK 0**
- **Frame 0, ACK 1**
- **Frame 1, ACK 1**
- **Frame 1, ACK 0**

**States**
- R = 0
- R = 1

**Time**
- Linear progression of frames and acknowledgments.

---

**Legend**
- **R**: Receiver
- **S**: Source
- **ACK**: Acknowledgment
Note:

In Stop-and-Wait ARQ, at any point in time for a sender, there is only one frame, the outstanding frame, that is sent and waiting to be acknowledged, wasting transmission medium capacity. To improve, multiple frames should be in transition while waiting for acknowledgement. Go-Back-N ARQ and Selective Repeat ARQ use this concept.
Go-Back-N ARQ

Sequence Number

Sender and Receiver Sliding Window

Control Variables and Timers

Acknowledgment

Resending Frames

Operation
Frames from a sender are numbered sequentially. But we need to set a limit.

If the header of the frame allows m bits for the sequence number, the sequence number ranges from 0 to $2^m - 1$.

Therefore, for $m=3$, we shall have sequence numbers from 0 to 7.
At the sender site, to hold the outstanding frames until they are acknowledged, we use the concept of window.

**Working:**

Frames to the left have already been acknowledged and can be purged. Those to the right cannot be sent until window slides over them.

The size of the window is at most $2^m - 1$. 

---

**Diagram:**

![Diagram of Sender Sliding window](image)

**Window size = 7**

**a. Before sliding**

![Diagram of window before sliding](image)

**b. After sliding two frames**

![Diagram of window after sliding two frames](image)
Go-Back-N ARQ

Receiver Sliding window

**Working:**

The size of the window at the receiver site in this protocol is always 1.

The receiver is always looking for a specific frame to arrive in a specific order.

Any frame arriving out of order is discarded and needs to be resent.

---

**Diagram:**

```
| 5 | 6 | 7 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 0 |
```

- **a. Before sliding**
- **b. After sliding**
Go-Back-N ARQ

Control Variables and Timer

S is sequence number of the recently sent frame

SF is sequence number of the first frame in the window

SL is sequence number of the last frame in the window

W = SL - SF + 1 where W is Window size.

The sender sets a timer for each frame sent. The receiver has no timer.

---

\[ S_F, S, S_L \]

---

a. Sender window

---

\[ R \]

---

b. Receiver window
Go-Back-N ARQ

Acknowledgement and resending

**Acknowledgement:**

# The receiver sends acknowledgement if a frame has arrived safe.

# If a frame is damaged or is received out of order, the receiver is silent and will discard all subsequent frames until it receives the one it is expecting.

# The silence of receiver causes the timer of the unacknowledged frame to expire.

# This causes the sender to go back and resend all frames, beginning with the one with the expired timer.

# The receiver does not have to acknowledge each frame received. It can send one cumulative acknowledgement for several frames.

**Resending Frames:**

Suppose the sender has already sent frame 6, but the timer for frame 3 expires. So the sender goes back and sends frames 3, 4, 5, 6 again. That’s why it is called Go-Back-N ARQ.
Go-Back-N ARQ Operation

Damaged or Lost Frame

Sender

Receiver

Frame 0

Frame 1

ACK 2

Frame 2

Lost

Frame 3

Discarded

Frame 3 discarded, not in the window

Time-out

Time
In this case we have two situations:

**Situation 1:** If the next acknowledgement arrives before the expiration of any timer, there is no need for retransmission of frames because acknowledgements are cumulative. For example, ACK4 means ACK 1 to ACK 4.

**Situation 2:** If the next ACK arrives after the time-out, the frames and all the frames after that are resent.

**Note:** Receiver never resends an Acknowledgement.

**Delayed ACK**

A delayed acknowledgement also triggers the resending of frames.
Why the size of sender’s window is $2^n - 1$?

- **Frame 0**: Sender sends frames 0, 1, 2, 3. Receiver accepts frames 0, 1, 2, 3.
- **Frame 1**: Sender sends frame 4. Receiver does not acknowledge. Sender timeout.
- **Frame 2**: Sender sends frame 5. Receiver does not acknowledge. Sender timeout.
- **Frame 3**: Sender sends frame 6. Receiver does not acknowledge. Sender timeout.
- **Frame 4**: Sender sends frame 7. Receiver acknowledges frame 7.

**a. Window size < $2^n$**

**b. Window size = $2^n$**

- Time-out
- Correctly discarded
- Time-out
- Erroneously accepted
Go-Back-N ARQ Operation

Bidirectional and Piggybacking

Go-Back-N can also be bidirectional. We can also use piggybacking to improve the efficiency of the transmission.

Each direction needs both a sender window and a receiver window.
Selective-Repeat ARQ

Sender and Receiver Windows
Operation
Sender Window Size
Bidirectional Transmission
Bandwidth-Delay Product
Pipelining
Previous protocols were inefficient.

In this protocol, we do not resend all frames when just one frame is damaged; only damaged frame is resent.

This mechanism is called selective repeat. In this processing at the receiver end is more complex.
Selective Repeat ARQ

Sender and Receiver Windows

# The sender and its control variables are same as in Go-Back-N.

# Window Size is half of $2^m$.

# Receiver also have a window with the same size as of sender.

# The receivers window specifies the range of the accepted received frame. Therefore receiver is looking for a range of sequence numbers.

# This protocol also defines negative ACK (NAK) that reports the sequence number of a damaged frame before the timer expires.

---

**a. Sender window**

- Frame acknowledged: $S_F$
- Frames waiting to be sent: $S_L$

**b. Receiver window**

- Frames received and acknowledged: $R_F$
- Frames that cannot be accepted: $R_L$
Selective Repeat ARQ

Figure below explains the concept.

Operation Normal and Lost Frame
Selective Repeat ARQ can also be bidirectional. We can also use piggybacking to improve the efficiency of the transmission.

Each direction needs both a sender window and a receiver window.