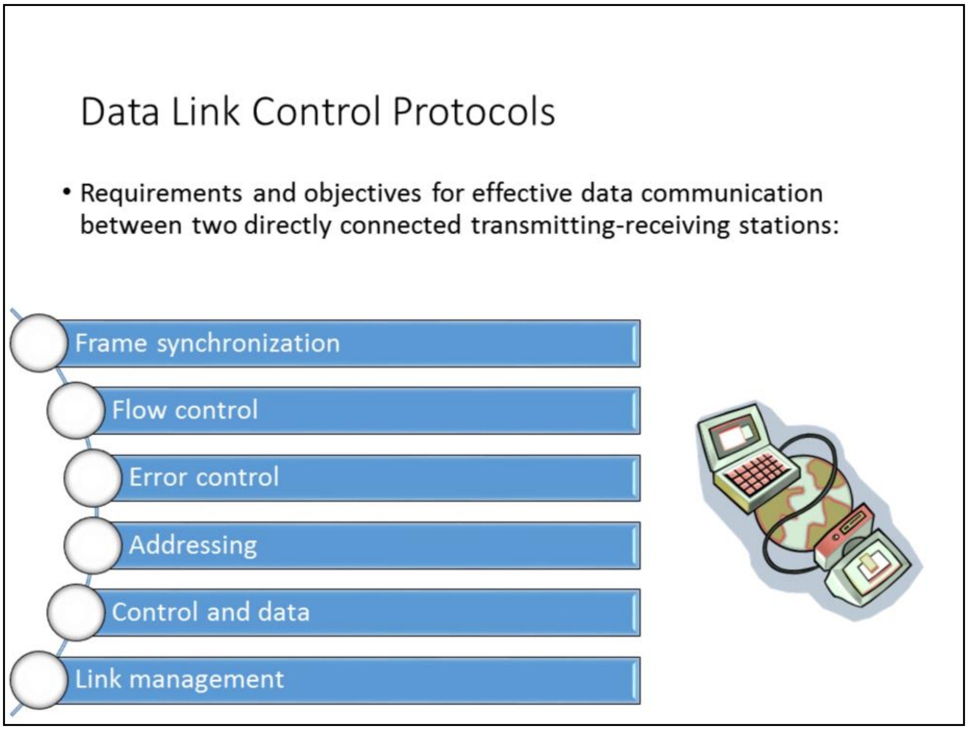


To see the need for data link control, we list some of the requirements and objectives for effective data communication between two directly connected
transmitting-receiving stations:

• Frame synchronization: Data are sent in blocks called frames. The beginning and end of each frame must be recognizable. We briefly introduced this topic with the discussion of synchronous frames (Figure 6.2).

• Flow control: The sending station must not send frames at a rate faster than the receiving station can absorb them.

• Error control: Bit errors introduced by the transmission system should be corrected.

• Addressing: On a shared link, such as a local area network (LAN), the identity of the two stations involved in a transmission must be specified.

• Control and data on same link: It is usually not desirable to have a physically separate communications path for control information. Accordingly, the receiver must be able to distinguish control information from the data being transmitted.

• Link management: The initiation, maintenance, and termination of a sustained data exchange require a fair amount of coordination and cooperation among stations. Procedures for the management of this exchange are required.

Flow Control

- Technique for assuring that a transmitting entity does not over-whelm a receiving entity with data
  - The receiving entity typically allocates a data buffer of some maximum length for a transfer
  - When data are received, the receiver must do a certain amount of processing before passing the data to the higher-level software
- In the absence of flow control, the receiver's buffer may fill up and overflow while it is processing old data

Flow control is a technique for assuring that a transmitting entity does not overwhelm a receiving entity with data. The receiving entity typically allocates a data buffer of some maximum length for a transfer. When data are received, the receiver must do a certain amount of processing before passing the data to the higher-level software. In the absence of flow control, the receiver's buffer may fill up and overflow while it is processing old data.

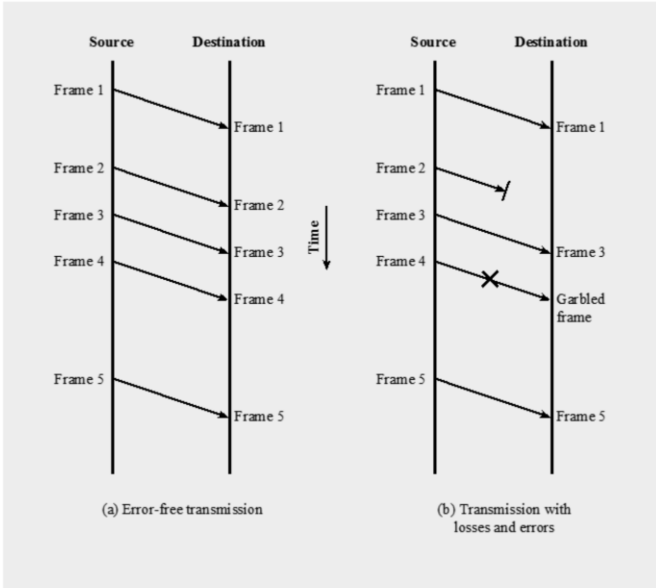


Figure 7.1 Model of Frame Transmission

To begin, we examine mechanisms for flow control in the absence of errors.

The model we will use is depicted in Figure 7.1a, which is a vertical time sequence

diagram. It has the advantages of showing time dependencies and illustrating the

correct send–receive relationship. Each arrow represents a single frame transiting a

data link between two stations. The data are sent in a sequence of frames, with each

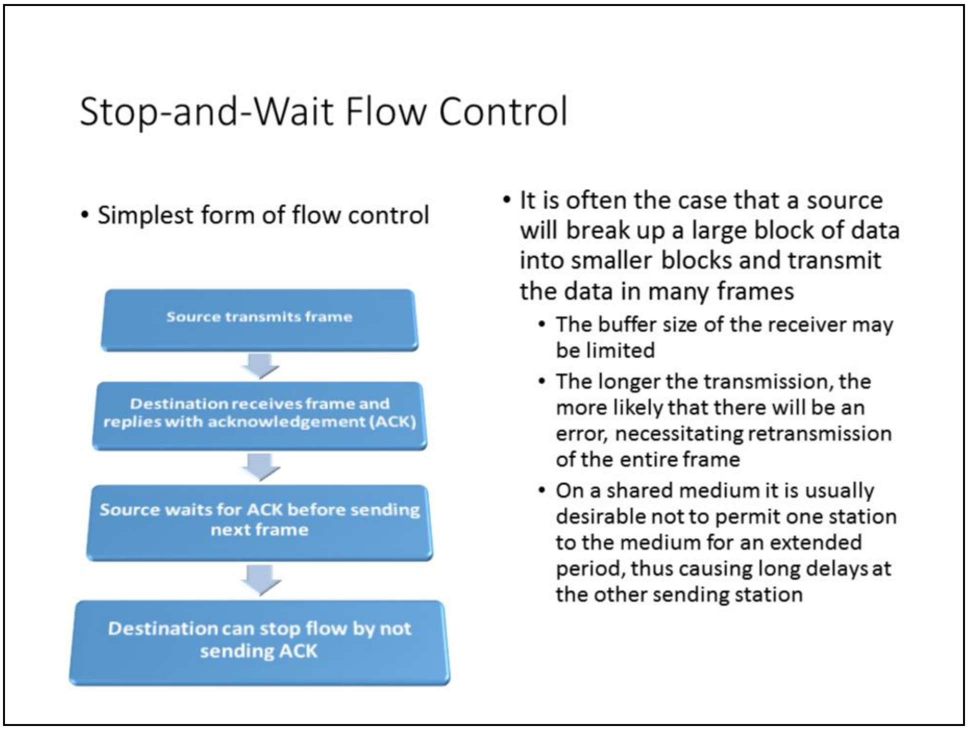
frame containing a portion of the data and some control information. The time it

takes for a station to emit all of the bits of a frame onto the medium is the transmission

time ; this is proportional to the length of the frame. The propagation time is the

time it takes for a bit to traverse the link between source and destination. For this

section, we assume that all frames that are transmitted are successfully received; no frames are lost and none arrive with errors. Furthermore, frames arrive in the same order in which they are sent. However, each transmitted frame suffers an arbitrary and variable amount of delay before reception.



The simplest form of flow control, known as stop-and-wait flow control, works as follows. A source entity transmits a frame. After the destination entity receives the frame, it indicates its willingness to accept another frame by sending back an acknowledgment to the frame just received. The source must wait until it receives the acknowledgment before sending the next frame. The destination can thus stop the flow of data simply by withholding acknowledgment. This procedure works fine and, indeed, can hardly be improved upon when a message is sent in a few large frames. However, it is often the case that a source will break up a large block of data into smaller blocks and transmit the data in many frames. This is done for the following reasons:

• The buffer size of the receiver may be limited.

• The longer the transmission, the more likely that there will be an error, necessitating retransmission of the entire frame. With smaller frames, errors are detected sooner, and a smaller amount of data needs to be retransmitted.

• On a shared medium, such as a LAN, it is usually desirable not to permit one station to occupy the medium for an extended period, thus causing long delays at the other sending stations.

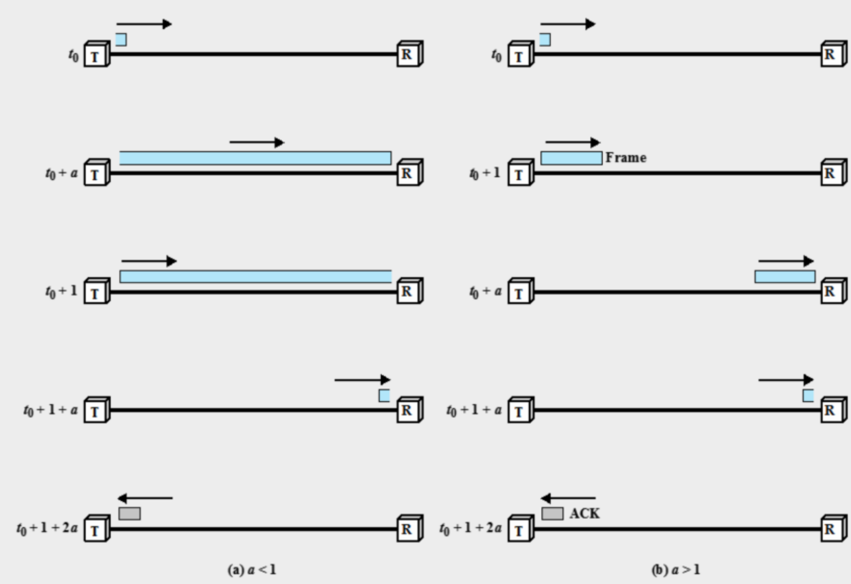


Figure 7.2 Stop-and-Wait Link Utilization (transmission time = 1; propagation time = a)

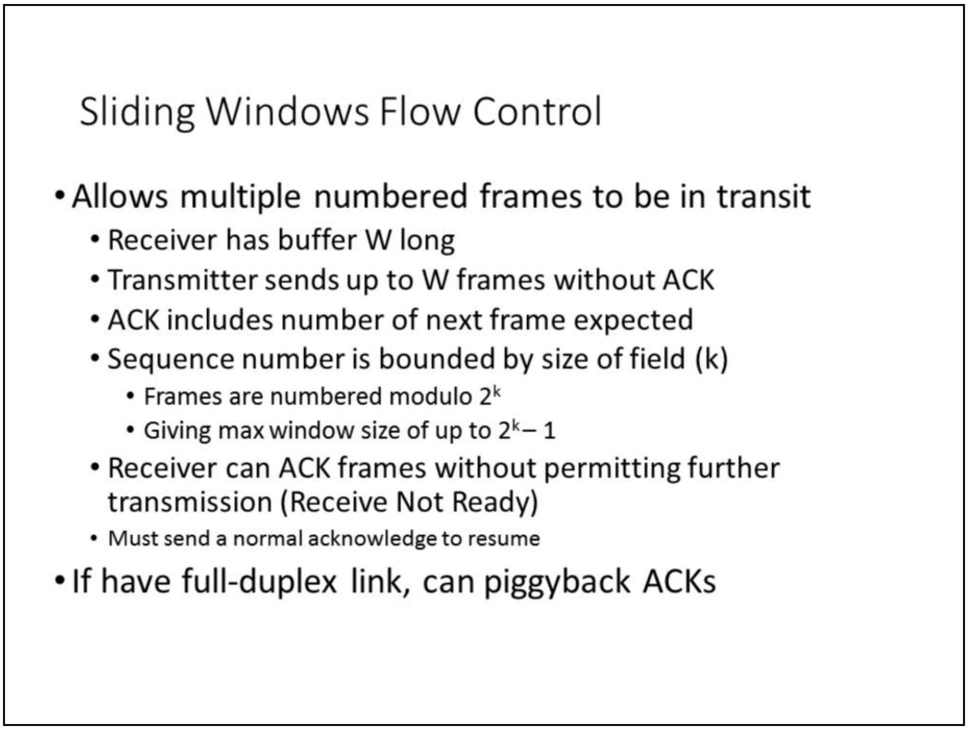
In situations where the bit length of the link is greater than the frame length, serious inefficiencies result. This is illustrated in Figure 7.2.

In the figure, the transmission time is normalized to one, and the propagation delay is expressed as the variable **a .**

When a is less than 1, the propagation time is less than the transmission time. In this case, the frame is sufficiently long that the first bits of the frame have arrived

When a is greater than 1, the propagation time is greater than the transmission time. In this case, the sender completes transmission of the entire frame before the leading bits of that frame arrive at the receiver. Put another way, larger values of a are consistent with higher data rates and/or longer distances between stations. Chapter 16 discusses a and data link performance.

Both parts of Figure 7.2 (a and b) consist of a sequence of snapshots of the transmission process over time. In both cases, the first four snapshots show the process of transmitting a frame containing data, and the last snapshot shows the return of a small acknowledgment frame. Note that for a >1, the line is always underutilized and even for a<1, the line is inefficiently utilized. In essence, for very high data rates, for very long distances between sender and receiver, stop-and-wait flow control provides inefficient line utilization.



The essence of the problem described so far is that only one frame at a time can be in transit. Efficiency can be greatly improved by allowing multiple frames to be in transit at the same time.

Consider two stations, A and B, connected via a full-duplex link. Station B allocates buffer space for *W* frames. Thus, B can accept *W* frames, and A is allowed to send *W* frames without waiting for any acknowledgments. To keep track of which frames have been acknowledged, each is labeled with a a *k*-bit sequence number. This gives a range of sequence numbers of 0 through $2^k - 1$, and frames are numbered modulo 2^k, with a maximum window size of $2^k - 1$. The window size need not be the maximum possible size for a given sequence number length k. B acknowledges a frame by sending an acknowledgment that includes the sequence number of the next frame expected. This scheme can also be used to acknowledge multiple frames, and is referred to as **sliding-window flow control**. Most data link control protocols also allow a station to cut off the flow of frames from the other side by sending a Receive Not Ready (RNR) message, which acknowledges former frames but forbids transfer of future frames. At some subsequent point, the station must send a normal acknowledgment to reopen the window. If two stations exchange data, each needs to maintain two windows, one for transmit and one for receive, and each side needs to send the data and acknowledgments to the other. To provide efficient support for this requirement, a feature known as **piggybacking** is typically provided. Each **data frame** includes a field that holds the sequence number of that frame plus a field that holds the sequence number used for acknowledgment.

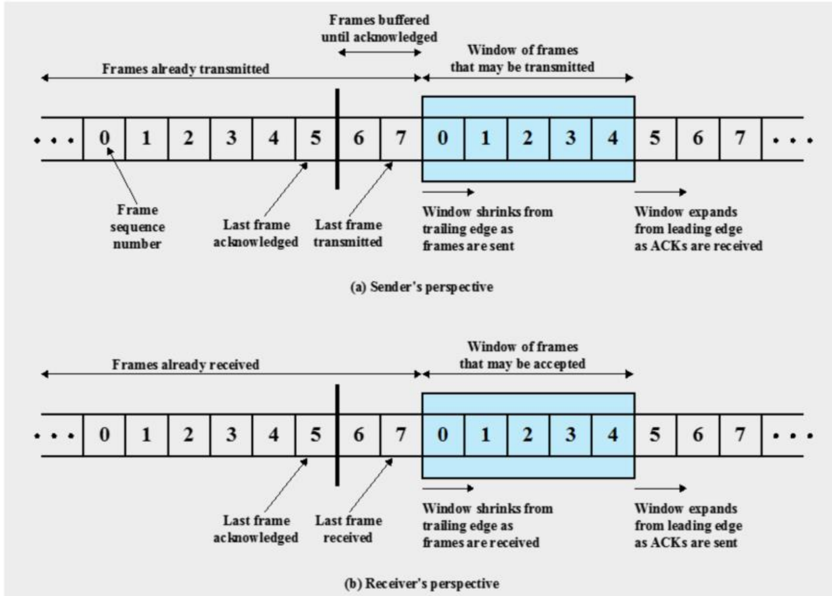


Figure 7.3 Sliding-Window Depiction

Figure 7.3 is a useful way of depicting the sliding-window process. It assumes the use of a 3-bit sequence number, so that frames are numbered sequentially from 0 through 7, and then the same numbers are reused for subsequent frames. The shaded rectangle indicates the frames that may be sent; in this figure, the sender may transmit five frames, beginning with frame 0. Each time a frame is sent, the shaded window shrinks; each time an acknowledgment is received, the shaded window grows. Frames between the vertical bar and the shaded window have been sent but not yet acknowledged. As we shall see, the sender must buffer these frames in case they need to be retransmitted.

The window size need not be the maximum possible size for a given sequence number length. For example, using a 3-bit sequence number, a window size of 5 could be configured for the stations using the sliding-window flow control protocol.

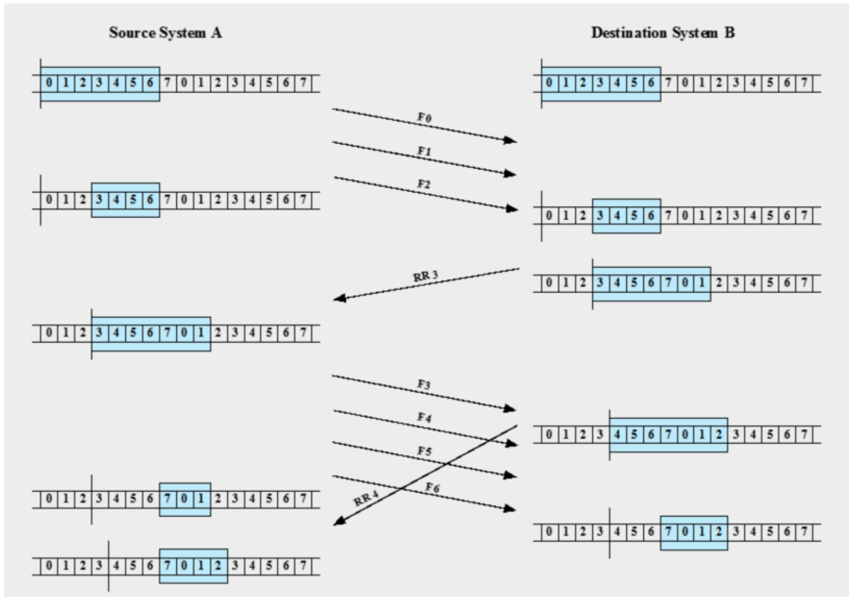
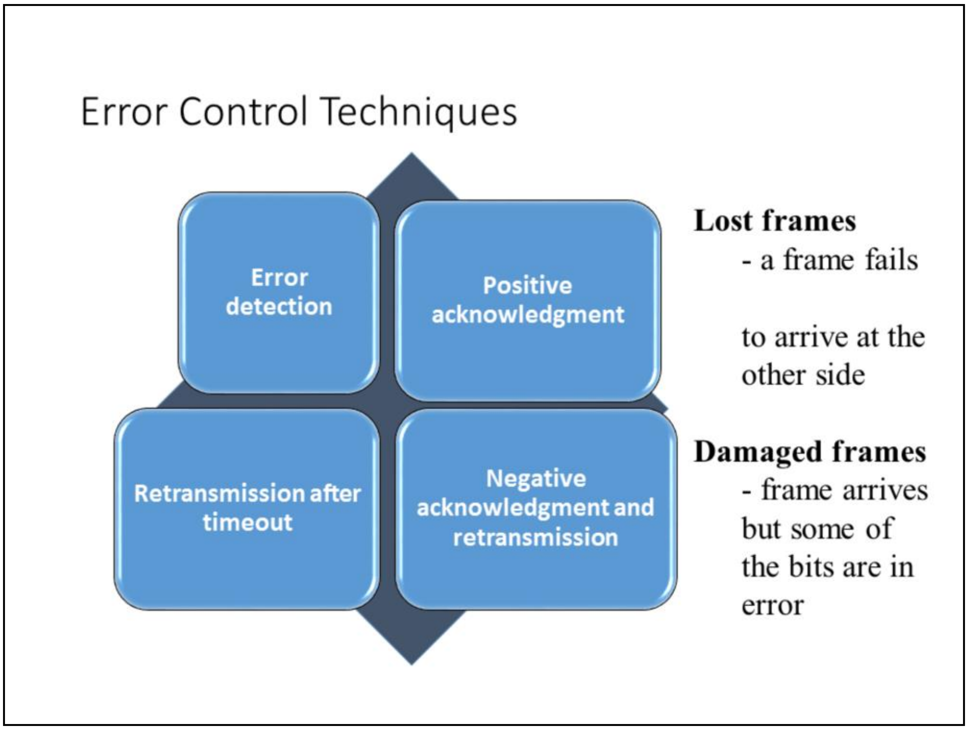


Figure 7.4 Example of a Sliding-Window Protocol

An example is shown in Figure 7.4. The example assumes a 3-bit sequence number field and a maximum window size of seven frames. Initially, A and B have windows indicating that A may transmit seven frames, beginning with frame 0 (F0). After transmitting three frames (F0, F1, F2) without acknowledgment, A has shrunk its window to four frames and maintains a copy of the three transmitted frames. The window indicates that A may transmit four frames, beginning with frame number 3. B then transmits an RR (receive ready) 3, which means "I have received all frames up through frame number 2 and am ready to receive frame number 3; in fact, I am prepared to receive seven frames, beginning with frame number 3." With this acknowledgment, A is back up to permission to transmit seven frames, still beginning with frame 3; also A may discard the buffered frames that have now been acknowledged. A proceeds to transmit frames 3, 4, 5, and 6. B returns RR 4, which acknowledges F3, and allows transmission of F4 through the next instance of F2. By the time this RR reaches A, it has already transmitted F4, F5, and F6, and therefore A may only open its window to permit sending four frames beginning with F7.

Sliding-window flow control is potentially much more efficient than stop-and-wait flow control. The reason is that, with sliding-window flow control, the transmission link is treated as a pipeline that may be filled with frames in transit. By contrast, with stop-and-wait flow control, only one frame may be in the pipe at a time. Chapter 16 quantifies the improvement in efficiency.



Error control refers to mechanisms to detect and correct errors that occur in the transmission of frames. The model that we will use, which covers the typical case, is

illustrated in Figure 7.1b. As before, data are sent as a sequence of frames; frames

arrive in the same order in which they are sent; and each transmitted frame suffers

an arbitrary and potentially variable amount of delay before reception. In addition,

we admit the possibility of two types of errors:

• Lost frame: A frame fails to arrive at the other side. In the case of a network, the network may simply fail to deliver a frame. In the case of a direct point-to-point data link, a noise burst may damage a frame to the extent that the receiver is not aware that a frame has been transmitted.

• Damaged frame: A recognizable frame does arrive, but some of the bits are in error (have been altered during transmission).

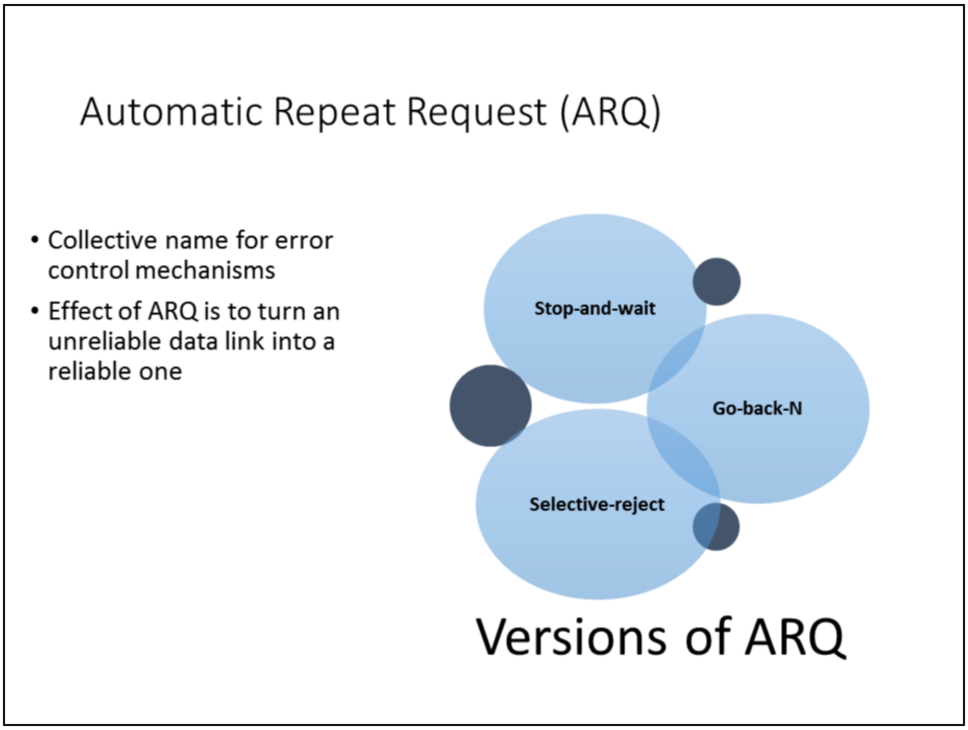
The most common techniques for error control are based on some or all of the following ingredients:

• Error detection: The destination detects frames that are in error, using the techniques described in the preceding chapter, and discards those frames.

• Positive acknowledgment: The destination returns a positive acknowledgment to successfully received, error-free frames.

• Retransmission after timeout: The source retransmits a frame that has not been acknowledged after a predetermined amount of time.

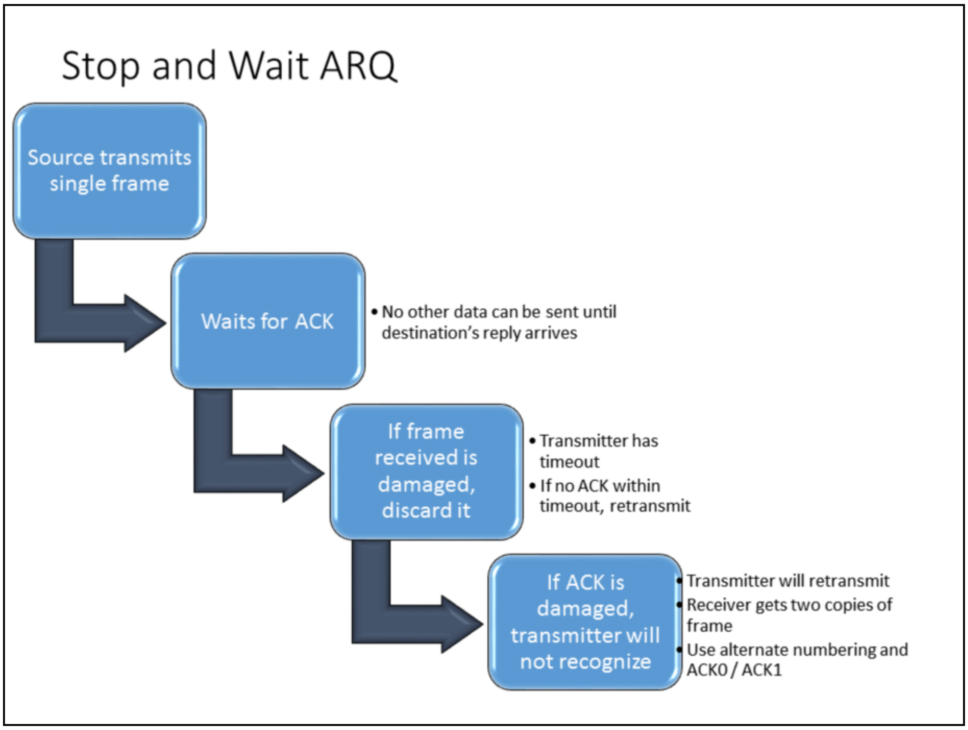
• Negative acknowledgment and retransmission: The destination returns a negative acknowledgment to frames in which an error is detected. The source retransmits such frames.



Collectively, these mechanisms are all referred to as automatic repeat request (ARQ) . The effect of ARQ is to turn a potentially unreliable data link into a reliable
one. Three versions of ARQ have been standardized:

- Stop-and-wait ARQ
- Go-back-N ARQ
- Selective-reject ARQ

All of these forms are based on the use of the flow control techniques discussed
in Section 7.1. We examine each in turn.



Stop-and-wait ARQ is based on the stop-and-wait flow control technique outlined previously. The source station transmits a single frame and then must await an acknowledgment (ACK). No other data frames can be sent until the destination station's reply arrives at the source station.

Two sorts of errors could occur. First, the frame that arrives at the destination could be damaged. The receiver detects this by using the error-detection technique referred to earlier and simply discards the frame. To account for this possibility, the source station is equipped with a timer. After a frame is transmitted, the source station waits for an acknowledgment. If no acknowledgment is received by the time that the timer expires, then the same frame is sent again. Note that this method requires that the transmitter maintain a copy of a transmitted frame until an acknowledgment is received for that frame.

The second sort of error is a damaged acknowledgment, which is not recognizable by A, which will therefore time out and resend the same frame. This duplicate frame arrives and is accepted by B. B has therefore accepted two copies of the same frame as if they were separate. To avoid this problem, frames are alternately labeled with 0 or 1, and positive acknowledgments are of the form ACK0 and ACK1. In keeping with the sliding-window convention, an ACK0 acknowledges receipt of a frame numbered 1 and indicates that the receiver is ready for a frame numbered 0.

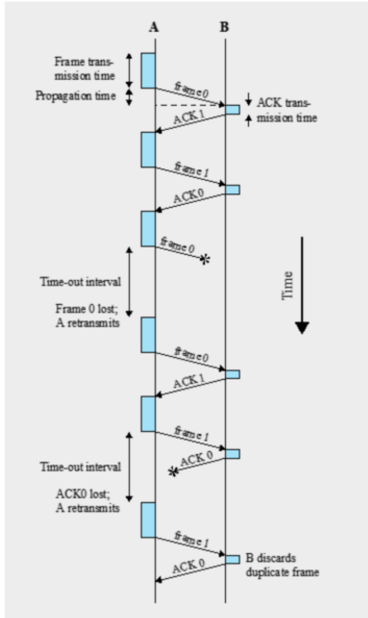


Figure 7.5 Stop-and-Wait ARQ

Figure 7.5 gives an example of the use of stop-and-wait ARQ, showing the transmission of a sequence of frames from source A to destination B.2 The figure shows the two types of errors just described. The third frame transmitted by A is lost or damaged and therefore B does not return an ACK. A times out and retransmits the frame. Later, A transmits a frame labeled 1 but the ACK0 for that frame is lost. A times out and retransmits the same frame. When B receives two frames in a row with the same label, it discards the second frame but sends back an ACK0 to each.

The principal advantage of stop-and-wait ARQ is its simplicity. Its principal disadvantage, as discussed in Section 7.1, is that stop-and-wait is an inefficient mechanism.

The sliding-window flow control technique can be adapted to provide more efficient line use; in this context, it is sometimes referred to as continuous ARQ .

Go-Back-N ARQ

- Most commonly used error control
- Based on sliding-window
- Use window size to control number of outstanding frames
- While no errors occur, the destination will acknowledge incoming frames as usual
  - RR=receive ready, or piggybacked acknowledgment
- If the destination station detects an error in a frame, it may send a negative acknowledgment
  - REJ=reject
  - Destination will discard that frame and all future frames until the frame in error is received correctly
  - Transmitter must go back and retransmit that frame and all subsequent frames

The form of error control based on sliding-window flow control that is most commonly used is called go-back-N ARQ. In this method, a station may send a series of frames sequentially numbered modulo some maximum value. The number of unacknowledged frames outstanding is determined by window size, using the sliding-window flow control technique. While no errors occur, the destination will acknowledge incoming frames as usual (RR = receive ready, or piggybacked acknowledgment). If the destination station detects an error in a frame, it may send a negative acknowledgment (REJ = reject) for that frame, as explained in the following rules. The destination station will discard that frame and all future incoming frames until the frame in error is correctly received. Thus, the source station, when it receives a REJ, must retransmit the frame in error plus all succeeding frames that were transmitted in the interim.

Selective-Reject (ARQ)

- Also called selective retransmission
- Only rejected frames are retransmitted
- Subsequent frames are accepted by the receiver and buffered
- Minimizes retransmission
- Receiver must maintain large enough buffer
- More complex logic in transmitter
  - Less widely used
- Useful for satellite links with long propagation delays

With selective-reject ARQ, the only frames retransmitted are those that receive a negative acknowledgment, in this case called SREJ, or those that time out. Selective reject would appear to be more efficient than go-back-N, because it minimizes the amount of retransmission. On the other hand, the receiver must maintain a buffer large enough to save post-SREJ frames until the frame in error is retransmitted and must contain logic for reinserting that frame in the proper sequence. The transmitter, too, requires more complex logic to be able to send a frame out of sequence. Because of such complications, select-reject ARQ is much less widely used than go-back-N ARQ. Selective reject is a useful choice for a satellite link because of the long propagation delay involved.

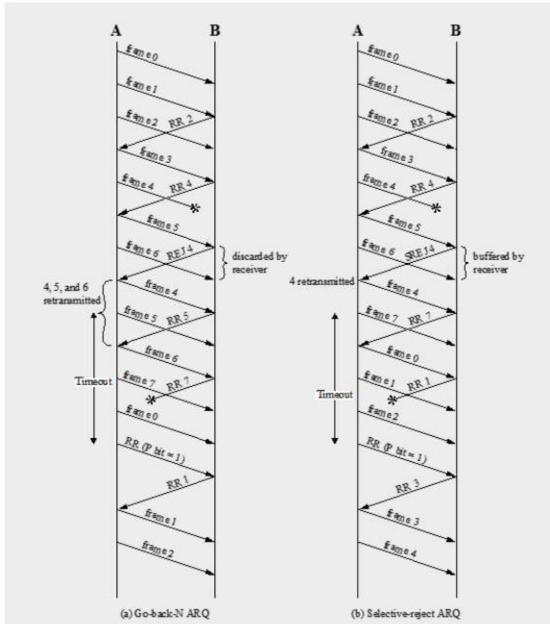


Figure 7.6 Sliding-Window ARQ Protocols

Figure 7.6a is an example of the frame flow for go-back-N ARQ.

Because of the propagation delay on the line, by the time that an acknowledgment

(positive or negative) arrives back at the sending station, it has already

sent at least one additional frame beyond the one being acknowledged. In this

example, frame 4 is damaged. Frames 5 and 6 are received out of order and are

discarded by B. When frame 5 arrives, B immediately sends a REJ 4. When the

REJ to frame 4 is received, not only frame 4 but frames 5 and 6 must be retransmitted.

Note that the transmitter must keep a copy of all unacknowledged

frames. Figure 7.6a also shows an example of retransmission after timeout. No

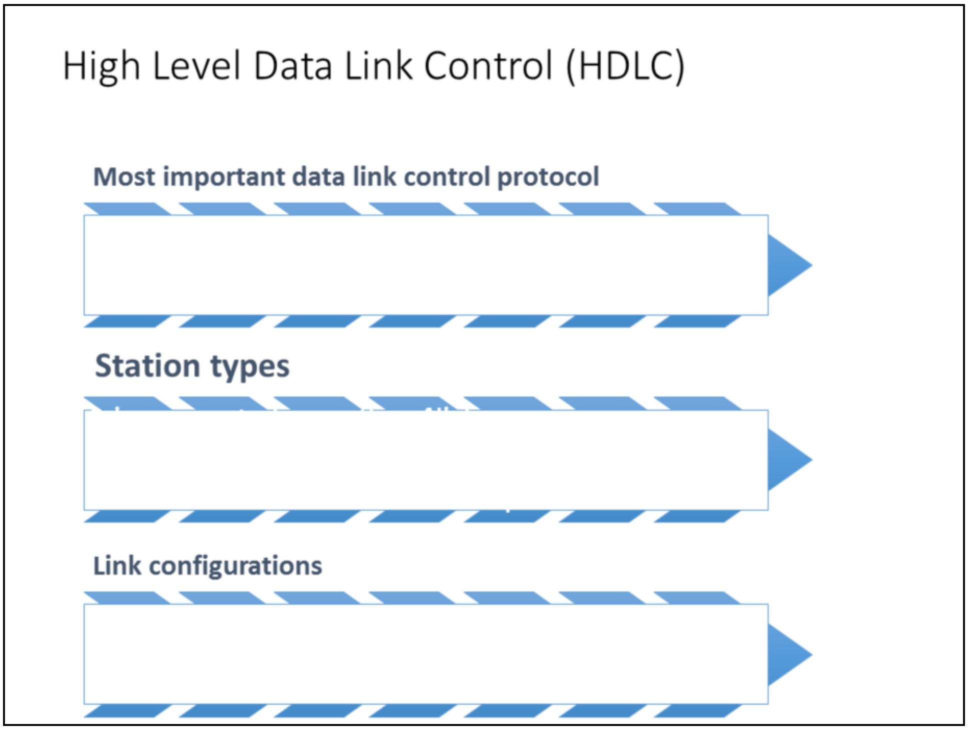
acknowledgment is received for frame 5 within the timeout period, so A issues

an RR to determine the status of B.

Figure 7.6b illustrates Selective-Reject ARQ. When frame 5 is received

out of order, B sends a SREJ 4, indicating that frame 4 has not been received.

However, B continues to accept incoming frames and buffers them until a valid frame 4 is received. At that point, B can place the frames in the proper order for delivery to higher-layer software.



The most important data link control protocol is HDLC (ISO 3009, ISO 4335). Not only is HDLC widely used, but it is the basis for many other important data link control protocols, which use the same or similar formats and the same mechanisms as employed in HDLC.

To satisfy a variety of applications, HDLC defines: three station types:

• **Primary station:** Responsible for controlling the operation of the link. Frames issued by the primary are called commands.

• **Secondary station:** Operates under the control of the primary station. Frames issued by a secondary are called responses. The primary maintains a separate logical link with each secondary station on the line.

• **Combined station:** Combines the features of primary and secondary. A combined station may issue both commands and responses.

It also defines two link configurations:

• **Unbalanced configuration:** Consists of one primary and one or more secondary stations and supports both full-duplex and half-duplex transmission.

• **Balanced configuration:** Consists of two combined stations and supports both full-duplex and half-duplex transmission.

HDLC Data Transfer Modes

Normal Response Mode (NRM)

- **Used with an unbalanced configuration**
- **Primary initiates transfer**

Asynchronous Balanced Mode (ABM)

- **Used with a balanced configuration**
- **Either station initiates transmission**
- **Has no polling overhead**
- **Most widely used**

Asynchronous Response Mode (ARM)

- **Used with unbalanced configuration**
- **Secondary may transmit without permission from primary**
- **Rarely used**

HDLC defines three data transfer modes:

- **Normal response mode (NRM):** Used with an unbalanced configuration. The primary may initiate data transfer to a secondary, but a secondary may only transmit data in response to a command from the primary. NRM is used on multi-drop lines, in which a number of terminals are connected to a host computer.

- **Asynchronous balanced mode (ABM):** Used with a balanced configuration. Either combined station may initiate transmission without receiving permission from the other combined station. ABM is the most widely used of the three modes; it makes more efficient use of a full-duplex point-to-point link because there is no polling overhead.

- **Asynchronous response mode (ARM):** Used with an unbalanced configuration. The secondary may initiate transmission without explicit permission of the primary. The primary still retains responsibility for the line, including initialization, error recovery, and logical disconnection. ARM is rarely used; it is applicable to some special situations in which a secondary may need to initiate transmission.

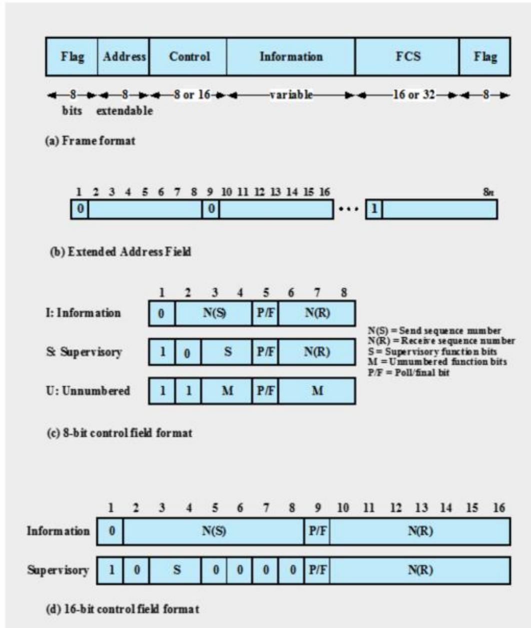


Figure 7.7 HDLC Frame Structure

HDLC uses synchronous transmission. All transmissions are in the form of frames,

and a single frame format suffices for all types of data and control exchanges.

Figure 7.7 depicts the structure of the HDLC frame. The flag, address, and

control fields that precede the information field are known as a header . The frame

check sequence and flag fields following the data field are referred to as a trailer .

Original Pattern:

111111111111011111101111110

After bit-stuffing

1111101111101101111101011111010

Figure 7.8 Bit Stuffing

Flag fields delimit the frame at both ends with the unique pattern 01111110. A single flag may be used as the closing flag for one frame and the opening flag for the next. On both sides of the user-network interface, receivers are continuously hunting for the flag sequence to synchronize on the start of a frame. While receiving a frame, a station continues to hunt for that sequence to determine the end of the frame. Because the protocol allows the presence of arbitrary bit patterns (i.e., there are no restrictions on the content of the various fields imposed by the link protocol), there is no assurance that the pattern 01111110 will not appear somewhere inside the frame, thus destroying synchronization. To avoid this problem, a procedure known as bit stuffing is used. For all bits between the starting and ending flags, the transmitter inserts an extra 0 bit after each occurrence of five 1s in

the
frame. After detecting a starting flag, the receiver monitors the bit stream. When a pattern of five 1s appears, the sixth bit is examined. If this bit is 0, it is deleted. If the sixth bit is a 1 and the seventh bit is a 0, the combination is accepted as a flag. If the sixth and seventh bits are both 1, the sender is indicating an abort condition.

With the use of bit stuffing, arbitrary bit patterns can be inserted into the data field of the frame. This property is known as data transparency .

Figure 7.8 shows an example of bit stuffing. Note that in the first two cases, the extra 0 is not strictly necessary for avoiding a flag pattern but is necessary for the operation of the algorithm.

Address Field

- Identifies secondary station that transmitted or will receive frame
- Usually 8 bits long
- May be extended to multiples of 7 bits
  - Leftmost bit indicates if is the last octet (1) or not (0)
- Address 11111111 allows a primary to broadcast a frame for reception by all secondaries

The address field identifies the secondary station that transmitted or is to receive the frame. This field is not needed for point-to-point links but is always included for the sake of uniformity. The address field is usually 8 bits long but, by prior agreement, an extended format may be used in which the actual address length is a multiple of 7 bits. The leftmost bit of each octet is 1 or 0 according as it is or is not the last octet of the address field. The remaining 7 bits of each octet form part of the address. The single-octet address of 11111111 is interpreted as the all-stations address in both basic and extended formats. It is used to allow the primary to broadcast a frame for reception by all secondaries.

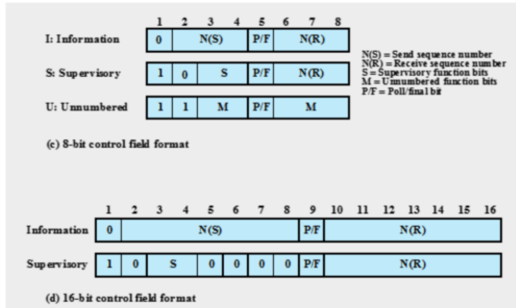
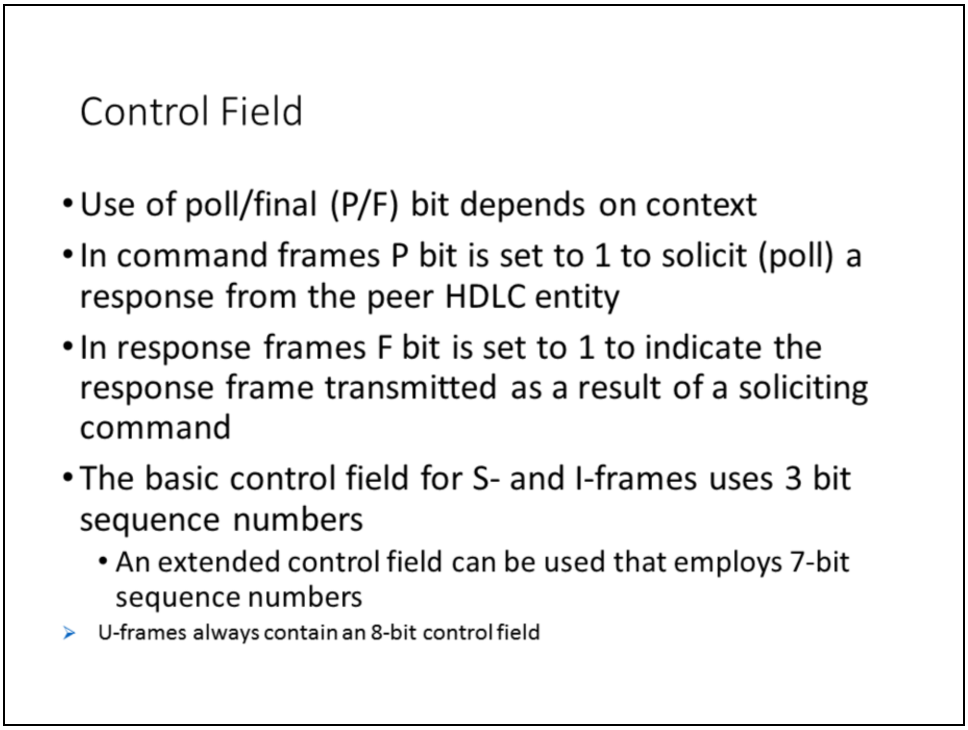


Figure 7.7 HDLC Frame Structure

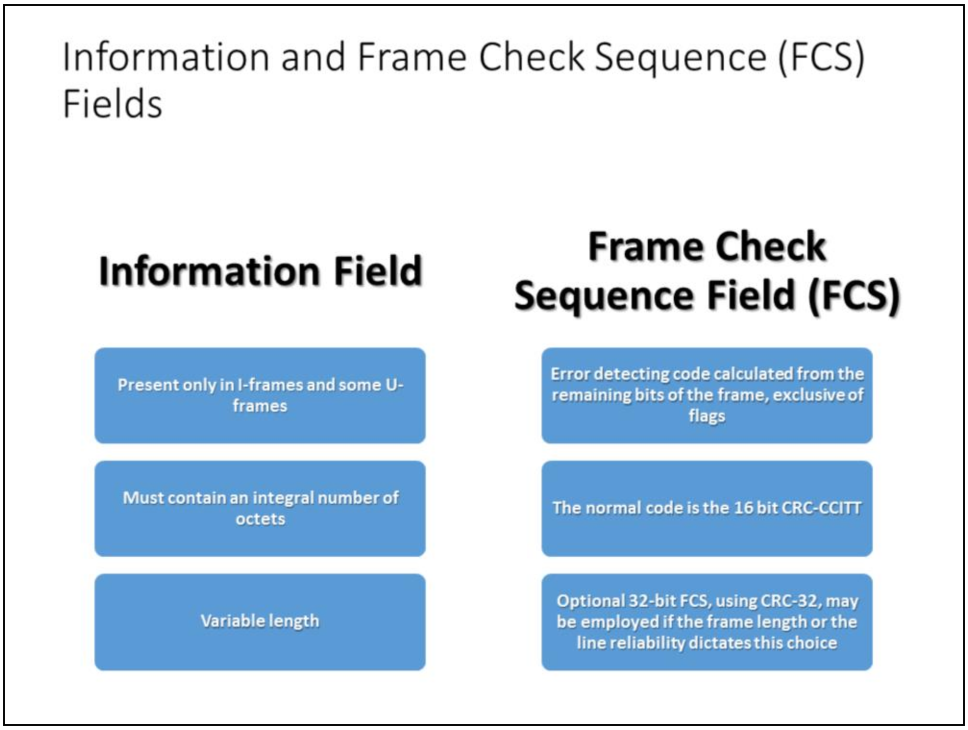
- HDLC defines three types of frames, each with a different control field format
  - Information frames (I-frames)
    - Carry the data to be transmitted for the user
    - Flow and error control data, using the ARQ mechanism, are piggybacked on an information frame
  - Supervisory frames (S-frames)
    - Provide the ARQ mechanism when piggybacking is not used
  - Unnumbered frames (U-frames)
    - Provide supplemental link control functions

HDLC defines three types of frames, each with a different control field format. Information frames (I-frames) carry the data to be transmitted for the user (the logic above HDLC that is using HDLC). Additionally, flow and error control data, using the ARQ mechanism, are piggybacked on an information frame. Supervisory frames (S-frames) provide the ARQ mechanism when piggybacking is not used. Unnumbered frames (U-frames) provide supplemental link control functions. The first one or two bits of the control field serves to identify the frame type. The remaining bit positions are organized into subfields as indicated in Figures 7.7c and d. Their use is explained in the discussion of HDLC operation later in this chapter.



All of the control field formats contain the poll/final (P/F) bit. Its use depends on context. Typically, in command frames, it is referred to as the P bit and is set to 1 to solicit (poll) a response frame from the peer HDLC entity. In response frames, it is referred to as the F bit and is set to 1 to indicate the response frame transmitted as a result of a soliciting command.

Note that the basic control field for S- and I-frames uses 3-bit sequence numbers. With the appropriate set-mode command, an extended control field can be used for S- and I-frames that employs 7-bit sequence numbers. U-frames always contain an 8-bit control field.



The information field is present only in I-frames and some U-frames. The field can contain any sequence of bits but must consist of an integral number of octets. The length of the information field is variable up to some system-defined maximum.

The frame check sequence (FCS) is an error-detecting code calculated from the remaining bits of the frame, exclusive of flags. The normal code is the 16-bit CRC-CCITT defined in Section 6.3. An optional 32-bit FCS, using CRC-32, may be employed if the frame length or the line reliability dictates this choice.

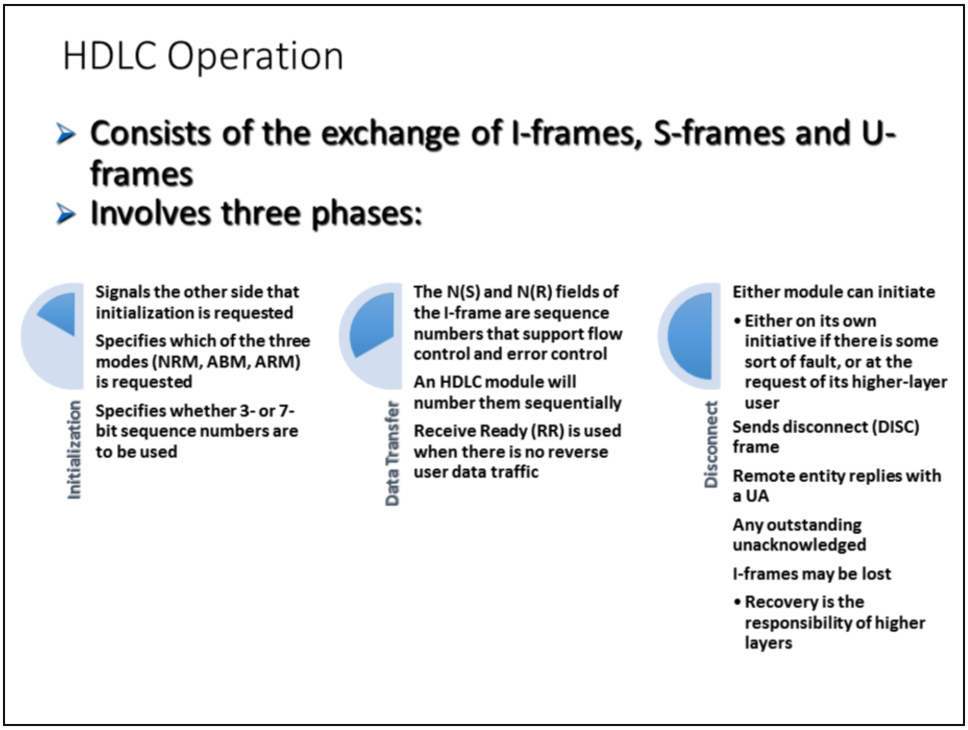
Name	Command/ Response	Description
Information (I)	C/R	Exchange user data
Supervisory (S)		
Receive ready (RR)	C/R	Positive acknowledgment; ready to receive I-frame
Receive not ready (RNR)	C/R	Positive acknowledgment; not ready to receive
Reject (REJ)	C/R	Negative acknowledgment; go back N
Selective reject (SREJ)	C/R	Negative acknowledgment; selective reject
Unnumbered (U)		
Set normal response/extended mode (SNRM/SNRME)	C	Set mode; extended = 7-bit sequence numbers
Set asynchronous response/extended mode (SARM/SARME)	C	Set mode; extended = 7-bit sequence numbers
Set asynchronous balanced/extended mode (SABM, SABME)	C	Set mode; extended = 7-bit sequence numbers
Set initialization mode (SIM)	C	Initialize link control functions in addressed station
Disconnect (DISC)	C	Terminate logical link connection
Unnumbered Acknowledgment (UA)	R	Acknowledge acceptance of one of the set-mode commands
Disconnected mode (DM)	R	Responder is in disconnected mode
Request disconnect (RD)	R	Request for DISC command
Request initialization mode (RIM)	R	Initialization needed; request for SIM command
Unnumbered information (UI)	C/R	Used to exchange control information
Unnumbered poll (UP)	C	Used to solicit control information
Reset (RSET)	C	Used for recovery; resets N(R), N(S)
Exchange identification (XID)	C/R	Used to request/report status
Test (TEST)	C/R	Exchange identical information fields for testing
Frame reject (FRMR)	R	Report receipt of unacceptable frame

Table 7.1

HDLC Commands and Responses

(Table can be found on page 230 in the textbook)

HDLC operation consists of the exchange of I-frames, S-frames, and U-frames
between two stations. The various commands and responses defined for these
frame
types are listed in Table 7.1.



The operation of HDLC involves three phases. First, one side or another initializes the data link so that frames may be exchanged in an orderly fashion. During this phase, the options that are to be used are agreed upon. After initialization, the two sides exchange user data and the control information to exercise flow and error control. Finally, one of the two sides signals the termination of the operation.

Either side may request initialization by issuing one of the six set-mode commands. This command serves three purposes:

1. It signals the other side that initialization is requested.

2. It specifies which of the three modes (NRM, ABM, ARM) is requested.

3. It specifies whether 3- or 7-bit sequence numbers are to be used.

If the other side accepts this request, then the HDLC module on that end transmits an unnumbered acknowledged (UA) frame back to the initiating side. If the request is rejected, then a disconnected mode (DM) frame is sent.

When the initialization has been requested and accepted, then a logical connection is established. Both sides may begin to send user data in I-frames, starting with sequence number 0. The N(S) and N(R) fields of the I-frame are sequence numbers that support flow control and error control. An HDLC module sending a sequence of I-frames will number them sequentially, modulo 8 or 128, depending on whether 3- or 7-bit sequence numbers are used, and place the sequence number in N(S). N(R) is the acknowledgment for I-frames received; it enables the HDLC module to indicate which number I-frame it expects to receive next.

S-frames are also used for flow control and error control. The receive ready (RR) frame acknowledges the last I-frame received by indicating the next I-frame expected. The RR is used when there is no reverse user data traffic (I-frames) to carry an acknowledgment. Receive not ready (RNR) acknowledges an I-frame, as with RR, but also asks the peer entity to suspend transmission of I-frames. When the entity that issued RNR is again ready, it sends an RR. REJ initiates the go-back-N ARQ. It indicates that the last I-frame received has been rejected and that retransmission of all I-frames beginning with number N(R) is required. Selective reject (SREJ) is used to request retransmission of just a single frame.

Either HDLC module can initiate a disconnect, either on its own initiative if there is some sort of fault, or at the request of its higher-layer user. HDLC issues a disconnect by sending a disconnect (DISC) frame. The remote entity must accept the disconnect by replying with a UA and informing its layer 3 user that the connection has been terminated. Any outstanding unacknowledged I-frames may be lost, and their recovery is the responsibility of higher layers.

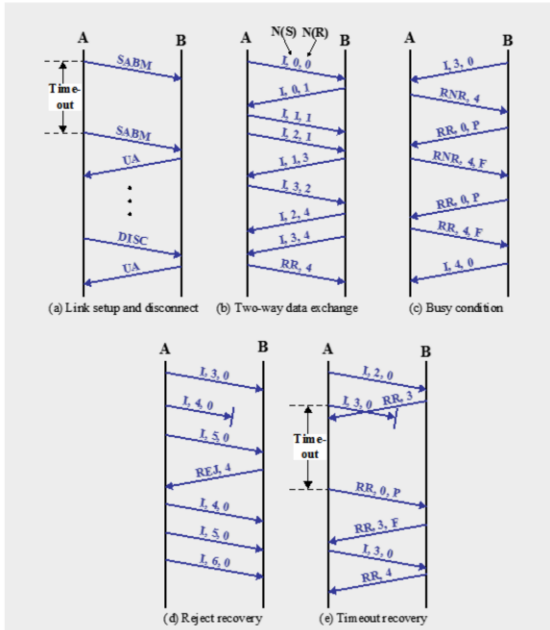


Figure 7.9 Examples of HDLC Operation

To better understand HDLC operation, several examples

are presented in Figure 7.9. In the example diagrams, each arrow includes a

legend that specifies the frame name, the setting of the P/F bit, and, where appropriate,

the values of N(R) and N(S). The setting of the P or F bit is 1 if the designation

is present and 0 if absent.

Figure 7.9a shows the frames involved in link setup and disconnect. The HDLC

protocol entity for one side issues an SABM command to the other side and starts

a timer. The other side, upon receiving the SABM, returns a UA response and sets

local variables and counters to their initial values. The initiating entity receives the

UA response, sets its variables and counters, and stops the timer. The logical connection

is now active, and both sides may begin transmitting frames. Should the timer expire without a response to an SABM, the originator will repeat the SABM, as illustrated. This would be repeated until a UA or DM is received or until, after a given number of tries, the entity attempting initiation gives up and reports failure to a management entity. In such a case, higher-layer intervention is necessary. The same figure (Figure 7.9a) shows the disconnect procedure. One side issues a DISC command, and the other responds with a UA response.

Figure 7.9b illustrates the full-duplex exchange of I-frames. When an entity sends a number of I-frames in a row with no incoming data, then the receive sequence number is simply repeated (e.g., I,1,1; I,2,1 in the A-to-B direction). When an entity receives a number of I-frames in a row with no outgoing frames, then the receive sequence number in the next outgoing frame must reflect the cumulative activity (e.g., I,1,3 in the B-to-A direction). Note that, in addition to I-frames, data exchange may involve supervisory frames.

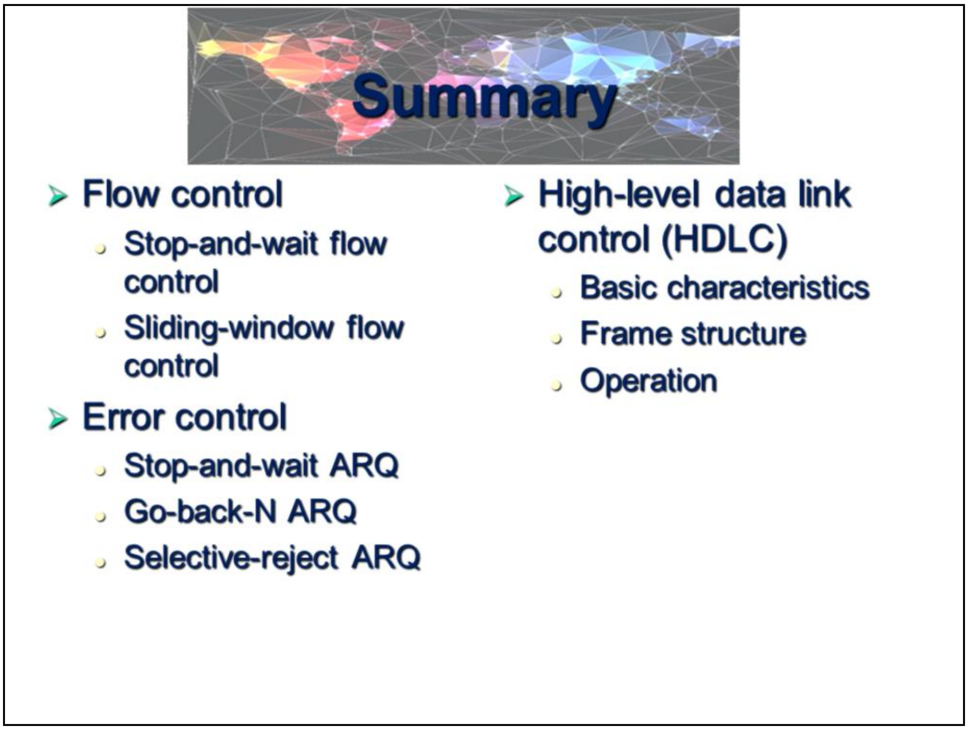
Figure 7.9c shows an operation involving a busy condition. Such a condition may arise because an HDLC entity is not able to process I-frames as fast as they are arriving, or the intended user is not able to accept data as fast as they arrive in I-frames. In either case, the entity's receive buffer fills up and it must halt the incoming flow of I-frames, using an RNR command. In this example, A issues an RNR, which requires B to halt transmission of I-frames. The station receiving the RNR will usually poll the busy station at some periodic interval by sending an RR with the P bit set. This requires the other side to respond with either an RR or an RNR. When the busy condition has cleared, A returns an RR, and I-frame transmission from B can resume.

An example of error recovery using the REJ command is shown in Figure 7.9d. In this example, A transmits I-frames numbered 3, 4, and 5. Number 4 suffers an error and is lost. When B receives I-frame number 5, it discards this frame because it is out of order and sends an REJ with an N(R) of 4. This causes A to initiate retransmission

of I-frames previously sent, beginning with frame 4. A may continue to send additional frames after the retransmitted frames.

An example of error recovery using a timeout is shown in Figure 7.9e. In this example, A transmits I-frame number 3 as the last in a sequence of I-frames. The frame suffers an error. B detects the error and discards it. However, B cannot send an REJ, because there is no way to know if this was an I-frame. If an error is detected in a frame, all of the bits of that frame are suspect, and the receiver has no way to act upon it. A, however, would have started a timer as the frame was transmitted.
This timer has a duration long enough to span the expected response time.
When the timer expires, A initiates recovery action. This is usually done by polling the other side with an RR command with the P bit set to determine the status of the other side. Because the poll demands a response, the entity will receive a frame containing an N(R) field and be able to proceed. In this case, the response indicates that frame 3 was lost, which A retransmits.

These examples are not exhaustive. However, they should give the reader a good feel for the behavior of HDLC.



Chapter 7 summary.