Data Link Control

A layer of logic above the physical interfacing, with transmission medium referred to as data link

Flow control enables a receiver to regulate data flow from sender to avoid buffer overflow; sender may not send data faster than receiver can absorb

Error detection is performed by receiver to check for errors in received code and to recover actual bits

Error control is performed by sender to retransmit damaged frames not acknowledged by receiver, or ones explicitly requested by receiver

Frame synchronization allows beginning and end of each block of data to be recognized

Addressing used to specify the identity of sender and receiver in a multipoint network (LAN)

Control and data must be distinguished from each other, specially when they are being transmitted over the same physical link

Link management refers to procedures for coordination and cooperation between stations to initiate, maintain, and terminate a sustained data exchange

Flow control

- Standard producer-consumer problem
- Error free flow control
  - Figure 7.1a, with vertical time sequence
- Stop-and-wait flow control
  - Simplest form of flow control
  - Source entity transmits a frame
  - Destination entity sends back a signal to acknowledge receipt and to indicate that it can receive another frame
  - Source waits for acknowledgement before sending another frame
  - Destination can stop flow of data by withholding acknowledgement
  - Works well for sending a few large blocks
  - Preferable to send smaller blocks
    - Buffer size of receiver may be limited
    - Shorter transmission time; damaged frame can be retransmitted in short time
    - One station does not monopolize shared transmission medium (LAN) for extended time
  - Stop-and-wait not very good for multiple frames for a single message
    - Only one frame at a time can be in transit
    - If bit length of link is greater than frame length, we can have serious inefficiencies
      - Bit length is the number of bits on the link when stream of bits fully occupies the link
      - Bit length is $R \times \frac{d}{V}$ where R is the data rate in bps, d is the distance of link in meters, and V is the velocity of propagation in m/s
    - Figure 7.2
      - Transmission time per frame normalized to 1
      - Propagation delay per bit expressed as variable a

        - Larger values of a are consistent with higher data rates and/or longer distances between stations

- Sliding-window flow control
  - Problem with only one frame being in transit at any time
  - Efficiency can be improved by having multiple frame in transit
  - Two stations A and B connected via a full-duplex link
    - Station A is the sender; and can send W frames without waiting for acknowledgement
    - Station B can receive up to W frames
    - Each frame labeled with a sequence number
    - Acknowledgement from B is the sequence number of next frame expected
    - Acknowledgement implicitly tells that B can receive W frames starting at the specified sequence number and can include multiple frames
    - B can also withhold acknowledgement until a certain frame has been received
    - A maintains a list of sequence numbers it is prepared to send
    - B maintains a list of sequence numbers it is prepared to receive
    - The two lists are *windows* of frames
  - Sequence number
    - Occupies a field in frame and must be of bounded size
    - With n bits for sequence number, frames must be numbered modulo 2^n
  - Figure 7.3
    - 3-bit sequence number
    - W is 5 (up to 5 frames in transit at any time)
    - Actual window size may not be W
    - Frames not yet acknowledged must be buffered in sender for possible retransmission
  - Figure 7.4
    - 3-bit sequence number
    - Withhold acknowledgement at times
  - Return Not Ready (RNR)
    - Allows receiver to acknowledge frames with an indication that it is not yet ready to accept more frames
    - At some future point, receiver must send a Return Ready (RR) acknowledgement to allow transmission of more frames
  - Transmission in both directions
    - Allowed for two stations to exchange data
    - Both need to maintain two windows – one for transmission and the other for receiving
    - Acknowledgement can be *piggybacked* onto normal frame
      - Each frame contains a field for its sequence number plus another field used for sequence number of a frame being acknowledged
    - If there is no frame to be sent, an RNR or RR message can be sent by itself
    - If there is data to be sent but no acknowledgement, the last acknowledgement can be repeated

Error detection

- Errors occur due to change in one or more bits during transmission
- Different probabilities related to error

- P_b – probability of single bit error; bit error rate (BER)
- P_1 – probability of frame arriving with no bit error
- P_2 – probability of one or more undetected bit errors in frame
- P_3 – probability of one or more detected bit errors but no undetected bit errors

- No effort is made to detect errors
  - $P_3 = 0$ – probability of detected bit errors is zero
  - Assume that probability of any bit being in error is constant, or $P_b = c$, and independent for each bit
  - With F bits per frame, we have

$$\begin{aligned} P_1 &= (1 - P_b)^F \\ P_2 &= 1 - P_1 \end{aligned}$$

  - Probability of no bit errors P_1 decreases as probability of a single bit error P_b increases
  - Probability of no bit error P_1 decreases with increasing frame length

- Example – ISDN connection
  - Aims at BER on 64 kbps channel to be less than 10^{-6} on at least 90% of observed one minute interval
  - User requires at most one frame with an undetected bit error per day on a continuously used 64-kbps channel
  - Frame length – 1000 bits
  - Number of frames transmitted in a day is: $24 \times 60 \times 60 \times 64 = 5.53 \times 10^6$
  - Desired frame error rate $P_2 = \frac{1}{5.53 \times 10^6} = 0.18 \times 10^{-6}$
  - Assume $P_b = 10^{-6}$
  - $P_1 = (0.999999)^{1000} = 0.999$, giving $P_2 = 10^{-3}$, which is about three times more than our requirement

- Additional bits needed to allow for error detection
  - Code calculated as function of other transmitted bits
  - Receiver can check the code and detect error
  - P_3 is the probability that errors in the frame will be detected by receiver
  - P_2 is the residual error rate and the probability that an error will go undetected despite the use of error-detection scheme

- Parity check

- Cyclic redundancy check (CRC)
  - Given a message of k bits
  - Generate an n bit sequence, called *frame check sequence* (FCS)
  - Resulting frame of $k + n$ bits is exactly divisible by some predetermined number
  - Case I: Modulo 2 arithmetic
    * Uses binary addition with no carry, or exclusive-OR operation
    * Define

T	$(k + n)$ bit frame to be transmitted, $n < k$
M	k bit message, in the first k positions of T
F	n-bit FCS, the last n bits of T
P	Pattern of $n + 1$ bits; predetermined divisor

    * The condition to be satisfied is: $T \% P = 0$

  - It is easy to see that

    $$T = 2^n M + F$$

    as M is in the high-order bits, or in the terminology of C

    ```
    T = M << n + F
    ```

  - Dividing $2^n M$ by P, we have quotient Q and remainder R as

    $$\frac{2^n M}{P} = Q + \frac{R}{P}$$

  - For modulo 2 division, remainder is always at least 1 bit less than divisor, to be used as FCS

    $$T = 2^n M + R$$

  - Does R satisfy our condition?

    $$\begin{aligned} T &= 2^n M + R \\ \frac{T}{P} &= \frac{2^n M + R}{P} \\ &= Q + \frac{R}{P} + \frac{R}{P} \\ &= Q + \frac{2R}{P} \\ &= Q \qquad\qquad (2x\%2 = 0) \end{aligned}$$

  - To generate FCS, divide $2^n M$ by P and use the remainder as FCS
  - Example: $M = 10\ 1000\ 1101$ (10 bits); $P = 11\ 0101$ (6 bits)
    Multiplying M by 2^5 yields 101 0001 1010 0000
    Dividing by P, we have

    $$\begin{aligned} Q &= 11\ 0101\ 0110 \\ R &= 0\ 1110 \end{aligned}$$

    Transmitted message T is: 101 0001 1010 1110
    Upon receipt, message is divided by P and $R = 0$ indicates no transmission error
  - Pattern P is chosen to be 1 bit longer than the desired FCS; both the high- and low-order bits of P must be 1
  - Errors in an $n + k$ bit frame can be represented by an $n + k$ bit field with 1 in each error position; resulting frame T_r is

    $$T_r = T \text{ xor } E$$

  - Receiver will fail to detect an error only if T_r is divisible by P, which happens if E is divisible by P

- Case II: Polynomials
  - Express all values as binary polynomial coefficients
  - Coefficients correspond to bits in a binary number

    $$\begin{aligned} M = 110011 \quad &\Rightarrow \quad M(X) = X^5 + X^4 + X + 1 \\ P = 11001 \quad &\Rightarrow \quad P(X) = X^4 + X^3 + 1 \end{aligned}$$

  - Arithmetic operations are modulo 2
  - CRC process is

    $$\begin{aligned} \frac{X^n M(X)}{P(X)} &= Q(X) + \frac{R(X)}{P(X)} \\ T(X) &= X^n M(X) + R(X) \end{aligned}$$

  - An error $E(X)$ will only be undetectable if it is divisible by $P(X)$; the following errors are detectable

    - All single-bit errors
    - All double-bit errors as long as $P(X)$ has at least three 1s
    - Any odd number of errors, as long as $P(X)$ contains a factor $(X+1)$
    - Any burst error for which length of burst is less than length of divisor polynomial ($\leq$ FCS)
    - Most larger burst errors
  - If all error patterns are considered equally likely, then for a burst of length $r+1$, probability of an undetected error is $\frac{1}{2^{r-1}}$; for a larger burst, probability is $\frac{1}{2^r}$, r being the length of FCS
  - Four widely used $P(X)$

CRC-12	$X^{12}+X^{11}+X^3+X^2+X+1$
CRC-16	$X^{16}+X^{15}+X^2+1$
CRC-CCITT	$X^{16}+X^{12}+X^5+1$
CRC-32	$X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^8+X^7+X^5+X^4+X^2+X+1$

  - CRC-12 is used for transmission of streams of 6 bit characters and generates a 12-bit FCS
  - CRC-16 and CRC-CCITT are used for 8-bit characters in US and Europe, giving 16-bit FCS
  - CRC-32 is specified as an option in some point-to-point synchronous transmission standards
- Digital logic
  - Implemented as a dividing circuit with exclusive-OR gates and a shift register
  - Shift register
    - String of 1-bit storage devices
    - Each device has an output line to show the value currently stored, plus an input line
    - At discrete clock times, value in storage device is replaced by value indicated on input line
    - Entire register is clocked simultaneously, causing a 1-bit shift in entire register
  - Circuit implementation
    - Register contains n bits, equal to FCS length
    - Up to n exclusive-OR gates
    - Presence or absence of a gate corresponds to the presence or absence of a term in the divisor polynomial $P(X)$, excluding the X^n term
  - Figure 7.6

$$\begin{array}{rcllcl} \text{Message } M & = & 10\ 1000\ 1101 & M(X) & = & X^9+X^7+X^3+X^2+1 \\ \text{Divisor } P & = & 11\ 0101 & P(X) & = & X^5+X^4+X^2+1 \end{array}$$

    - Five shift register storage devices corresponding to $P(X)$
    - Initialize by clearing all registers (make zero)
    - Enter message one bit at a time, starting with most significant bit
    - No feedback till the most significant bit arrives at left end of register; everything is shift up to that point
    - Whenever a 1 arrives at left end of rigister, 1 is subtracted from second (C_3), fourth (C_1), and sixth (input) bits on next shift; resulting in binary long division
    - Process continues through all bits of message, plus five zero bits (for shifting)
    - After processing last bit, shift register contains remainder (FCS)
  - Same logic is repeated at receiver end; no errors will be indicated by 0 in shift register

Error control

- Mechanism to detect and *correct* errors in transmitted frames
- Two types of errors during transmission
  1. Lost frames

    - Frame did not arrive at receiver end; possible address damage due to noise
  2. Damaged frame
    - Some bits in the received frame are altered during transmission

- **Automatic Repeat Request** or ARQ: Techniques for error control
  - Error detection
  - Positive acknowledgement
    * Receiver returns an acknowledgement that it received an error-free frame
  - Retransmission after timeout
    * Source retransmits frame if acknowledgement not received within predetermined time
  - Negative acknowledgement and retransmission
    * Receiver returns negative acknowledgement for frames with errors, effectively asking for retransmission
- Stop-and-wait ARQ
  - Very simple process
  - Based on stop-and-wait flow control technique
  - Source sends a frame and waits for ACK
  - No other data frames are sent till receiver's ACK is received
  - Error possibilities
    1. Frame damaged in transit
       * Receiver detects damaged frame using error-detection techniques, and discards the frame
       * No ACK is sent to sender
       * Sender retransmits after timeout expiration
       * Sender must maintain a copy of waht was sent til an ACK is received
    2. Damaged ACK
       * Frame correctly received by receiver but ACK damaged in transit
       * ACK not recognized by sender who sends a second copy of same frame
       * Receiver ends up with two copies of same frame
       * Problem solved by labeling frames with 0 or 1
       * Positive acknowledgement of the form ACK0 (for frame 1) or ACK1 (for frame 0), using sliding window convention
  - Figure 7.8
- Go-back-n ARQ
  - Based on sliding window flow control
  - Sender may send a series of frames numbered sequentially modulo some max value
  - Number of unacknowledged outstanding frames determined by window size using sliding window flow control
  - In case of no errors, receiver acknowledges RR
  - If an error is detected, receiver sends a reject (REJ) for that frame
  - Receiver discards that frame and future frames till frame is correctly received
  - Sender must retransmit the frame in error and all succeeding frames that were transmitted
  - Go-back-N takes into account following contingencies
    1. Damaged frame
       * Received frame is erroneous as determined by receiver

* Receiver discards the frame, leading to two possibilities
  (a) Sender sends next frame
    · Receiver receives next frame out of order and sends an REJ for previous (damaged frame)
    · Sender retransmits damaged frame and all subsequent frames
  (b) Sender does not send additional frames right away
    · Receiver receives nothing and does not return either RR or REJ
    · Upon timeout, sender transmits an RR frame with a P bit set to 1
    · Receiver interprets the RR frame with a P bit of 1 as a command to be acknowledged by sending an RR, with the next frame expected
    · When sender receives the RR, it retransmits the lost frame

2. Damaged RR has two subcases
   (a) Receiver receives frame and sends an RR which is lost in transit
     * It is possible that a subsequent acknowledgement arrives before the timeout at sender, causing no problem
   (b) Sender's timeout expires
     * Sender transmits an RR frame with P bit as 1
     * A P-bit timer is set
     * If receiver fails to respond, the P-bit timer expires
     * Sender issues a new RR and resets the P-bit timer
     * The above procedure is repeated a number of times, and finally, the sender initiates a reset procedure
3. Damaged REJ
   * Handled as case 1b

– Figure 7.9a

- Selective-Reject ARQ
  – Only those frames are retransmitted that receive a negative acknowledgement, or that time out
  – Figure 7.9b
  – SREJ is sent when a frame is received out of order
  – Receiver continues to accept incoming frames and buffers them till the lost frame is received
  – Receiver must maintain a large enough buffer to save post-SREJ frames
  – Transmitter also requires more complex logic, limiting the use of this technique

High-level Data link Control (HDLC)

- Most important data link protocol; widely used
- Basic characteristics
  – Three types of stations
    1. Primary station
       * Responsible for controlling the operation of link
       * Frames issued here are called commands
    2. Secondary station
       * Operates under control of primary station
       * Frames issued here are called responses
       * Primary maintains a separate logical link with each secondary station on the line

3. Combined station
   * May issue both commands and responses
- Two link configurations
  1. Unbalanced configuration
     * One primary and one or more secondary stations
     * Supports both full duplex and half duplex transmission
  2. Balanced configuration
     * Two combined stations
     * Supports both full duplex and half duplex transmission
- Three data transfer modes
  1. Normal response mode (NRM)
     * Used with unbalanced configuration
     * Primary may initiate data transfer to secondary
     * Secondary may only transmit data in response to a command from primary
     * Used on multidrop lines in which a number of terminals are connected to host computer
       · Host polls each terminal for input
     * Also used on point-to-point links, particularly if the link connects a terminal or other peripheral to computer
  2. Asynchronous balanced mode (ABM)
     * Most widely used of the three modes
     * Used with balanced configuration
     * Either combined station may initiate transmission without receiving permission from the other combined station
     * Makes more efficient use of a full-duplex point-to-point link as there is no polling overhead
  3. Asynchronous response mode (ARM)
     * Used with unbalanced configuration
     * Secondary station may initiate transmission without explicit permission from the primary
     * Primary retains responsibility for the line, including initialization, error recovery, and logical disconnection
     * Rarely used
     * Applicable in cases where secondary may need to initiate transmission

- Frame structure
  - HDLC uses synchronous transmission in the form of frames
  - All frames (data and control) are of same format
  - Figure 7.10a
  - Header fields
    * Precede the information field
    * 8-bit flag, 8-bit (extendible) address, and 8- or 16-bit control
  - Trailer fields
    * Follow the information field
    * 16- or 32-bit FCS field
    * 8-bit flag
  - Flag fields
    * Delimit the frame at both ends with unique pattern 0111 1110
    * Single flag can be used as a closing flag for one frame and opening flag for next

    * Used by receivers to synchronize on the start of frame
    * While frame is being received, receiver looks for the flag to determine the end of frame
    * Presence of the flag pattern within the information carried by the frame can destroy synchronization
    * Problem fixed by *bit stuffing*
      · Between starting and ending flags, transmitter always inserts an extra 0 after each occurrence of five consecutive 1s
      · Receiver monitors the bit stream and examines sixth bit after getting five 1s; if it is 0, it is ignored, if 1, it looks for the next bit which if 0, indicates the end of frame
      · If sixth and seventh bit are both 1s, sender is assumed to have indicated an abort condition
    * *Data transparency*
      · Bit stuffing can be used to insert arbitrary bit patterns in information fields
    * Figure 7.11
  – Address field
    * Identifies the secondary station that transmitted or is to receive the frame
    * Not needed for point-to-point links, but included for uniformity
    * Generally 8-bit long but can be longer (multiple of 7 bits) by prior agreement
      · In extended format, leftmost bit of each octet is 1 if it is the last octet; otherwise, the MSB is 0
      · Remaining 7 bits form part of the address
      · Single octet with all 1s is interpreted as "all stations" in both basic and extended formats, for broadcast to all stations
  – Control field
    * Identifies three different types of frames defined in HDLC
      1. Information frames (I-frames)
         · Carry data to be transmitted for the user
         · Flow and error control data, using ARQ, are piggybacked on the I-frames
      2. Supervisory frames (S-frames)
         · Provide the ARQ mechanism when piggybacking is not used
      3. Unnumbered frames (U-frames)
         · Provide supplemental link control functions
         · First one or two bits specify the frame type
           0 by itself defines I-frame
           10 defines S-frame
           11 defines U-frame
         · Remaining bit positions are organized into subfields as per Figures 7.10c and 7.10d
    * All control field formats contain the poll/final (P/F) bit, with its use defined by context
      · In command frames, it is referred to as P bit and is set to 1 to poll a response frame from peer HDLC entity
      · In response frames, it is referred to as F bit and is set to 1 to indicate the response frame transmitted as a result of polling command
  – Information field
    * Present only in I-frames and some U-frames
    * Can contain any number of bits but integral number of octets
    * Length is variable depending on some system defined maximum
  – Frame check sequence field
    * Error-detection code computed from remaining bits of the frame, exclusive of flags
    * Any of the CRC

- Operation
  - Consists of the exchange of I-, S-, and U-frames between two stations
  - Involves three phases
    1. Initialization
       - Requested by either side by issuing one of six set-mode commands to achieve the following
         (a) Signal the other side that initialization is requested
         (b) Specifies the requested mode (NRM, ARM, ABM)
         (c) Specifies whether 3- or 7-bit sequence numbers are to be used
       - Upon acceptance, HDLC module on the other side transmits an unnumbered acknowledgement (UA) frame back to initiator
       - If request is denied, a disconnected mode (DM) frame is sent
    2. Data transfer
       - After initialization, a logical connection is established
       - Both sides can send user data in I-frames, starting with sequence number 0
       - $N(S)$ and $N(R)$ fields in I-frame are sequence numbers to support flow control and error control
       - HDLC module numbers I-frames sequentially, modulo 8 or 128, depending on 3-bit or 7-bit sequence numbers in use, putting sequence numbers in $N(S)$
       - $N(R)$ is the acknowledgement for received I-frames; informs about the number of I-frame expected next
       - S-frames can be used for flow and error control as well
         - RR frame acknowledges the last I-frame received by indicating the next I-frame expected
         - RR is used when there is no reverse user data traffic (I-frames) to carry acknowledgement
         - RNR acknowledges an I-frame but asks to suspend transmission of I-frames, with transmission to be resumed through an RR frame
         - REJ initiates go-back-n-ARQ
         - SREJ is used to request retransmission of just a single frame
    3. Disconnect
       - Either HDLC module can initiate a disconnect
       - Initiated by sending a disconnect (DISC) frame
       - Remote entity must accept the disconnect by replying with a UA and informing the upper layer about the disconnection
       - Any outstanding I-frame may be lost, with responsibility for recovery resting with higher layers
  - Figure 7.12 – Examples
    - Figure 7.12a
      - Frames involved in link setup and disconnect
      - SABM initiates the setup
      - UA finally sets it up, and makes logical connection active
      - To disconnect, one side issues a DISC command and the other side responds with a UA
    - Figure 7.12b
      - Full-duplex exchange of I-frames
      - $N(S)$ and $N(R)$ carry the sequence number and response number
      - We may send an S-frame if the I-frame is not being sent
    - Figure 7.12c
      - Busy condition
      - Incoming flow of I-frames halted by RNR
    - Figure 7.12d
      - Recovery using REJ
    - Figure 7.12e
      - Error recovery using timeout