Data Link Control Protocols

Flow Control
Flow Control

- Ensuring the sending entity does not overwhelm the receiving entity
  - Preventing buffer overflow

Transmission time
- Time taken to emit all bits of frame onto medium
- Proportional to length of frame (and data rate)

Propagation time
- Time for a bit to traverse link between source and destination (a function of velocity and distance)

Model of Frame Transmission

(a) Error-free transmission  
(b) Transmission with losses and errors
Stop-and-Wait

- Simplest form of flow control
- Basic behavior
  - Source transmits frame
  - Destination receives frame and replies with acknowledgement (ACK)
  - Source waits for ACK before sending next frame
  - Destination can stop flow by not sending ACK
- Works well for a few large frames

Fragmentation

- Large block of data may be split into small frames
- Why?
  - Limited buffer size at receiver
  - Errors detected sooner (when flawed frame received)
  - On error, retransmission of smaller frames is needed
  - Prevents one station from occupying medium for long periods on shared medium
- S&W often becomes inadequate due to inherent inefficiency of link utilization
  - Particularly bad for high data rates and/or over long distances (i.e. where prop. time > trans. time)
S&W Link Utilization

Allow multiple frames to be in transit
- Transmission link is treated as a pipeline
- Receiver has buffer that is W frames long
- Transmitter can send up to W frames w/o ACK
- Each frame is numbered
- ACK includes number of next frame expected
  (RR = Receive Ready)
- Range of seq. numbers bounded by size of field
  - $k$-bit field => seq. #s 0 thru $2^k-1$
  - Actual window size may be < max. possible size
  - Frames are numbered modulo $2^k$

Note: Transmission time of ACK here is assumed negligible.
Sliding-Window Diagram

(a) Sender's perspective

(b) Receiver's perspective

Sliding-Window Example

Source System A

Destination System B

Here we have 3-bit seq. number, window size of 7 frames
Sliding-Window Enhancements

- Receiver can acknowledge frames without permitting further transmission (Receive Not Ready = RNR)
  - But, must later send a normal ACK to resume
- If duplex, can use piggybacking
  - Each data frame includes both seq. # of that frame plus seq. # used for ACK of reverse direction
  - If no data to send, use acknowledgement frame
  - If data but no acknowledgement to send, send last acknowledgement number again

Data Link Control Protocols

  Error Control
Error Control

- Detection and correction of errors
- What do we worry about?
  - Lost frames (fail to arrive)
  - Damaged frames (some bits garbled)
- Automatic repeat request (ARQ)
  - Error detection (e.g. CRC)
  - Positive acknowledgment (from receiver on success)
  - Retransmission after timeout (when timer expires)
  - Negative acknowledgement and retransmission
    - Instead of making sender wait for timeout to retransmit, receiver returns NACK to frames where error detected

Automatic Repeat Request (ARQ)

- Three versions are standardized
  - Stop-and-Wait ARQ
  - Go-Back-N ARQ
  - Selective-Reject ARQ
- Each is an extension to S&W or sliding-window schemes for flow control that we previously studied
  - Extended for error control
Stop-and-Wait ARQ

- Source transmits single frame
- Waits for ACK
- If received frame damaged, receiver discards it
  - Sender has timeout (equipped with a timer)
  - If no ACK within timeout, sender retransmits
- If ACK is damaged, sender will not recognize it
  - Sender will retransmit
  - Receiver gets two copies of frame
  - How to solve duplication at receiver?
    - Frames alternately numbered with 0 or 1 by sender
    - Positive ACKs from receiver use ACK0 and ACK1 alternately
      - e.g. like sliding-window, ACK0 means “#1 good, send me #0”!
    - Receiver recognizes duplicate and discards one

Example: Stop-and-Wait ARQ
Stop-and-Wait ARQ Tradeoffs

- Pro
  - Simple

- Con
  - Inefficient

Go-Back-N ARQ

- Based on sliding-window flow control
- Use window to control # of outstanding frames
- If no error, ACK from receiver as usual with # of next frame expected (via RR or piggyback)
- If error detected, reply with rejection (REJ)
  - Destination *discards that frame and all future frames* until error frame received correctly
  - Source must go back and *retransmit that frame and all subsequent frames*
Several Contingencies

Go-back-N ARQ accounts for several contingencies
- Damaged/lost data frame
- Damaged/lost RR
- Damaged/lost REJ

Note
- Distinction between *damaged* and *lost* frame is minor here, since with only error detection all bits in frame are suspect
- Thus, we cannot tell if damaged frame was a garbled data frame, RR, REJ, etc.

Damaged/lost Data Frame

Case (a)
- Sender transmits frame *i*
- Frame *i* is lost or damaged in transit
  - If latter, detected and discarded by receiver
- Sender soon after sends *i+1*
- Receiver gets frame *i+1* out of sequence
- Receiver sends REJ-*i*
- Sender retransmits frame *i* and all subsequent frames

Case (b)
- Frame *i* lost or damaged/discharded and no additional frames are sent
- Receiver gets nothing, returns neither ack. nor rejection
- When sender’s timer expires, it transmits RR including special P bit set
  - P=1 tells destination this is a command that must be acknowledged
- Receiver responds with RR-*i* and sender then retransmits frame *i*

Alternative
- Receiver could send REJ-*x* (*x* the next frame expected) when it discards an error frame, but receiver does not know the purpose of damaged frame and thus might be rejecting a good frame currently in flight ☹️
Damaged/lost RR

- **Case (a)**
  - Receiver gets frame \( i \) and sends RR \((i + 1)\)
  - RR is lost in transit
  - Since acks. are cumulative, a subsequent RR for subsequent frame may arrive before sender’s timer expires on frame \( i \)
  - Thus, problem worked itself out on its own 😊

- **Case (b)**
  - If sender times out, it sends RR to receiver with P=1 as before
  - Sender sets another timer (P-bit timer)
  - If receiver fails to respond or response is damaged, P-bit timer expires
  - If so, sender again sends RR with P=1 and resets P-bit timer
  - This procedure can be repeated # of times before some limit is reached and reset procedure is initiated

Damaged/lost REJ

- Same procedure as lost data frame
  - If sender sends more frames, REJ-\( i \) repeated
  - If sender does not, it times out
    - Sends RR with P=1 to receiver as before
    - Receiver responds with RR-\( i \)
    - Sender then retransmits frame \( i \)
Example: Go-Back-N ARQ

Selective-Reject ARQ

- Also called *selective retransmission*
- **Key features**
  - Only selectively rejected frames (via SREJ) are retransmitted
  - Subsequent frames are accepted by the receiver and buffered
  - Receiver orders frames before presenting to higher layer
- **Pros and cons**
  - Minimizes retransmission
  - Receiver more complex
    - Must maintain buffer big enough to save post-SREJ frames until frame in error is retransmitted
    - Extra logic for reinserting retransmitted frame in proper sequence
  - Sender more complex
    - Extra logic for sending a frame out of sequence
Example: Selective-Reject ARQ

Maximum window size? (1)

- Assume a $k$-bit sequence field
- $W$ limited by interaction b/w error control and ack.

**Go-Back-N ARQ**

- $W \leq 2^k - 1$
  - e.g. $k = 3 \Rightarrow W \leq 7$
- Why can’t we let $W = 2^k$?
  - Example ($k = 3$, $W = 2^3$) involving hosts A and B
    - A sends frame 0 then gets back an RR1 from B
    - A sends frames 1, 2, 3, 4, 5, 6, 7, 0
    - A gets back an RR1 from B
    - But what does RR1 mean? Could be cumulative ack. or could be repeat of previous RR1 with all 8 frames lost in transit?
  - Problem solved by limiting $W \leq 2^k - 1$
Maximum window size? (2)

**Selective-Reject ARQ**
- \( W \leq 2^{k-1} \)
  - e.g. \( k = 3 \Rightarrow W \leq 4 \)
- Why more restricted than with GBN?
  - Example \((k = 3, W = 2^{k-1})\) involving hosts A and B
    - A sends frames 0, 1, 2, 3, 4, 5, 6 to B
    - B receives all frames then cumulatively acks. with RR7
    - RR7 is lost in transit
    - A times out and retransmits frame 0
    - However, B has already advanced its window to accept next frames \((7, 0, 1, 2, 3, 4, 5)\)
    - Thus, upon receiving retransmitted frame 0, B assumes it is a new frame 0 and accepts it as such assuming that frame 7 must have been lost
  - Problem caused by overlap b/w sending and receiving windows
    - Solved by limiting \( W \leq 2^{k-1} \)

---

**Data Link Control Protocols**

High-Level Data Link Control (HDLC)
High-Level Data Link Control

- HDLC
- Important data-link control protocol
  - Widely used
  - Also, many others based on HDLC with various alterations
- ISO 33009, ISO 4335

HDLC Station Types

- Primary station
  - Controls operation of link
  - Frames issued are called *commands*
  - Maintains separate logical link to each secondary station

- Secondary station
  - Under control of primary station
  - Frames issued called *responses*

- Combined station
  - May issue commands and responses
HDLC Link Configurations

- **Unbalanced**
  - One primary and one or more secondary stations
  - Supports full duplex and half duplex

- **Balanced**
  - Two combined stations
  - Supports full duplex and half duplex

HDLC Transfer Modes (1)

- **Normal Response Mode (NRM)**
  - *Unbalanced* configuration
  - Primary initiates transfer to secondary
  - Secondary may only transmit data in response to command from primary
  - Used on multi-drop lines
  - Host computer as primary
  - Terminals as secondary
HDLC Transfer Modes (2)

Asynchronous Response Mode (ARM)
- *Unbalanced* configuration
- Secondary may initiate transmission without permission from primary
- Like NRM, primary responsible for line
  - Initialization, error recovery, logical disconnection
- Rarely used

HDLC Transfer Modes (3)

Asynchronous Balanced Mode (ABM)
- *Balanced* configuration
- Either station may initiate transmission without receiving permission
- No polling overhead
- *Most widely used*
HDLC Frame Structure

- Synchronous transmission
- All transmissions in frames
- Single frame format for all data and control exchanges

<table>
<thead>
<tr>
<th>Flag</th>
<th>Address</th>
<th>Control</th>
<th>Information</th>
<th>FCS</th>
<th>Flag</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>8</td>
<td>8 or 16</td>
<td>variable</td>
<td>16</td>
<td>8</td>
</tr>
</tbody>
</table>

Flag Fields

- Delimit frame at both ends
- Flag = 01111110
- May use to close one frame and open another
- Receiver hunts for flag sequence to synchronize
- Bit stuffing used to avoid confusion with data containing same pattern 01111110
  - 0 inserted after every sequence of five 1s
  - If receiver detects five 1s, it checks next bit
  - If 0, it is deleted
  - If 1 and seventh bit is 0, accept as flag
  - If sixth and seventh bits 1, sender is indicating abort
Bit Stuffing

Original Pattern:

11111111111101111110111110

After bit-stuffing

1111011111011011111011111010

(a) Example

Flag Flag

Transmitted Frame

Flag Flag Flag

Received Frame

(b) An inverted bit splits a frame in two

Flag Flag

Transmitted Frame

Flag Flag Flag

Received Frame

(c) An inverted bit merges two frames

Address Field

- Identifies secondary station that sent or will receive frame
  - Not needed for pt-to-pt links but included for uniformity
  - Field is usually 8 bits long
- Extensible by prior agreement, multiples of 7 bits
  - Leftmost bit of each octet indicates whether it is last octet (1) or not (0)
- All ones (11111111) is broadcast
Control Field

- Different for each frame type
  - Information frame (I-frame)
    - Data to be transmitted to user (next layer up)
    - Flow and error control piggybacked on information frames
  - Supervisory frame (S-frame)
    - Provides ARQ when piggyback not used
  - Unnumbered frame (U-frame)
    - Supplementary link control
- First 1-2 bits of control field identify frame type
- Remaining bits specify the control characteristics

Control Field Diagram

(c) 8-bit control field format

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>N(S)</td>
<td>P/F</td>
<td>N(R)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| I: Information
| S: Supervisory
| U: Unnumbered |

N(S) = Send sequence number
N(R) = Receive sequence number
S = Supervisory function bits
M = Unnumbered function bits
P/F = Poll/final bit

(d) 16-bit control field format

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td>N(S)</td>
<td>P/F</td>
<td></td>
<td></td>
<td></td>
<td>N(R)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| Information
| Supervisory |

Flag Address Control Information IFS P/S
Poll/Final (P/F) Bit

- Use depends on context
- Command frame
  - P bit
  - 1 to solicit (poll) response from peer
- Response frame
  - F bit
  - 1 indicates response to soliciting command

Information Field

- Only in I-frames and some U-frames
- Must contain integral number of octets
- Variable length up to some system-defined maximum
FCS Field

- Frame Check Sequence
- Error detection
- 16-bit CRC
  - CRC-CCITT: \( P(X) = X^{16} + X^{12} + X^5 + 1 \)
- Optional 32-bit CRC (CRC-32)

HDLC Operation

- Exchange of information, supervisory, and unnumbered frames
- Three phases
  - Initialization
  - Data transfer
  - Disconnect
- See Table 7.1 for HDLC Commands and Responses
  - Many types of U-frames including 6 set-mode commands
  - 4 types of S-frames
### Table 7.1 HDLC Commands and Responses

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

### Examples of Operation (1)

(a) Link setup and disconnect
(b) Two-way data exchange
(c) Busy condition
Examples of Operation (2)

(d) Reject recovery

(e) Timeout recovery

Other DLC Protocols (e.g. LLC)

- **Logical Link Control (LLC)**
  - **IEEE 802.2**
  - Different frame format
  - Link control split between two sublayers, medium access layer (MAC) and LLC (on top of MAC)
  - No primary and secondary - all stations are peers
  - Two MAC addresses needed
    - Sender and receiver
  - Error detection at MAC sub-layer
    - 32-bit CRC
  - Destination and source access points (DSAP, SSAP) at LLC sub-layer
  - LLC control field same as HDLC but only 7-bit seq. #s