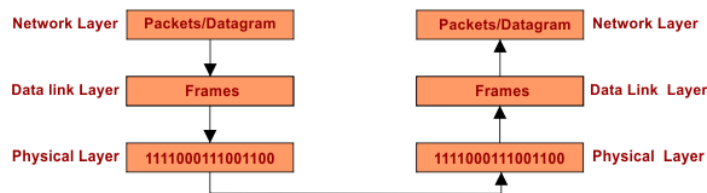


Chapter 4

Data Link layer

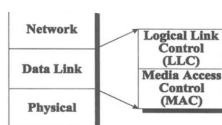
The physical layer allows a stream of bits to be transmitted between two remote systems. The link layer (layer no. 2) must ensure the correction of the transmitted bits and gather them into packets to pass them to the Network layer. The link layer retrieves data packets from the network layer, wraps them in frames which sends them one by one to the physical layer.



To make transmission reliable, the link layer must ensure:

- The delimitation of the exchanged data blocks;
- Control of the integrity of the data received;
- The organization and control of the exchange.
- Access control to a shared channel

In fact, the link layer consists of two sublayers: LLC (Logical Link Control) and MAC (Media Access Control). The LLC sublayer provides the first three functions while the MAC sublayer provides the last.

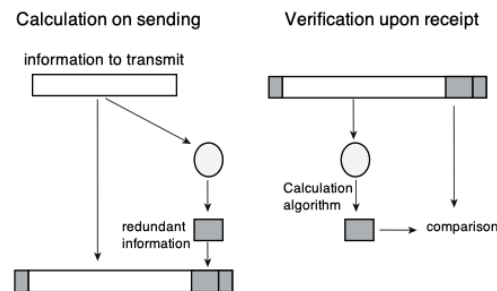


The advantage of flags is that they allow you to always find synchronization and send frames of any size.

4.2 Error Detection and Correction

Generally speaking, when transmitting data, we must ensure that the data received has not been altered during transmission. Several factors can modify the data content such as interference caused by electromagnetic radiation or distortion of transmission cables.

Protection methods exploit data redundancy by adding control bits to data bits. The control bits are calculated, at the transmitter, by an algorithm specified in the protocol from the data block. On reception, we execute the same algorithm to check if the redundancy is consistent. If this is the case, it is considered that there is no transmission error and the information received is processed; otherwise, we are certain that the information is invalid.



Several error protection methods can be used:

4.2.1 Data duplication

An example of control bits is the duplication of transmitted bits (repetition detection). The coded message is a duplicate copy of the initial message, the receiver knows that there has been an error if the copies are not identical, it then requests retransmission of the message. If the same error occurs on both copies, the error will not be detected.

If the message is sent in triplicate, the receiver can even correct the error by taking the values of the two identical copies without requesting retransmission from the transmitter.

4.2.2 Parity check code

It is a code in which a bit (the parity bit) is added to the initial word to ensure parity. Its yield is low when k is small.

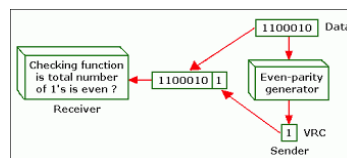
Example: Transmission of characters using a representation code (7-bit ASCII code).

7 bits of data	(count of 1-bits)	8 bits including parity	
		even	odd
0000000	0	00000000	00000001
(Q) 1010001	3	10100011	10100010
(i) 1101001	4	11010010	11010011
1111111	7	11111111	11111110

This code is able to detect all odd number errors but it does not detect even number errors. It can detect a parity error, but cannot locate it.

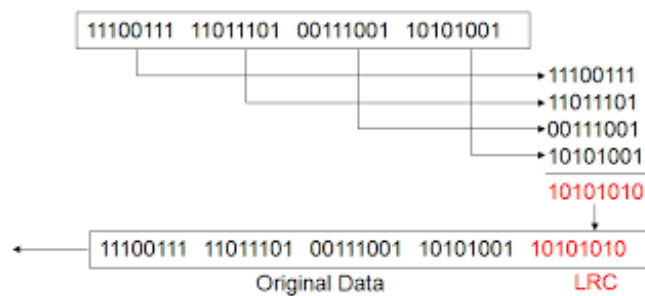
4.2.2.1 Vertical parity

To each character we add a bit (vertical redundancy bit or parity bit, VRC: Vertical Redundancy Check)



4.2.2.2 Longitudinal parity

To each character block, we add an additional control field (LRC: Longitudinal Redundancy Check)



Longitudinal parity was initially used for magnetic tapes to supplement vertical parity error detection.

4.2.2.3 Longitudinal and vertical parity

The data block is arranged in a matrix form ($k = a \bullet b$). Parity is applied to each row and each column. We obtain a matrix $(a + 1, b + 1)$. An LRC character is added to the transmitted block. Each bit of the LRC character corresponds to the parity of the bits of each character of the same rank: the first bit of the LRC is the parity of all 1^{er} bits of each character, the second of all 2^e bits... The character thus formed is added to the message. The LRC is itself protected by a parity bit (VRC).

	H	E	L	L	O	LRC →
bit 0	0	1	0	0	1	0
bit 1	0	0	0	0	1	1
bit 2	0	1	1	1	1	0
bit 3	1	0	1	1	1	0
bit 4	0	0	0	0	0	0
bit 5	0	0	0	0	0	0
bit 6	1	1	1	1	1	1
VRC ↓	0	1	1	1	1	0

1001000	0	1000101	1	1001100	1	1001100	1	1001111	1	1000010	0
H		E		L		L		O		LRC	

4.2.3 Polynomial codes

The polynomial code method (or CRC: Cyclic Redundant Coding) is the most used method for detecting grouped errors. Before transmission, control bits are added. If errors are detected upon reception, the message must be retransmitted.

In this code, information of n bits is considered as the list of binary coefficients of a polynomial of n terms, therefore of degree $n - 1$.

Example:

- $1101 \rightarrow x^3 + x^2 + 1$
- $110001 \rightarrow x^5 + x^4 + 1$
- $11001011 \rightarrow x^7 + x^6 + x^3 + x + 1$

To calculate the control bits, a certain number of operations are carried out with these polynomials with binary coefficients. All these operations are carried out modulo 2. This is how, in additions and subtractions, we do not take into account the carry: Any addition and any subtraction are therefore identical to an XOR operation. For example:

$$\begin{array}{r}
 1 \ 0 \ 0 \ 1 \ 1 \ 0 \ 1 \ 1 \\
 + \ 1 \ 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \\
 \hline
 = \ 0 \ 1 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1
 \end{array}$$

$$\begin{array}{r}
1 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \\
- \ 1 \ 0 \ 1 \ 0 \ 0 \ 1 \ 1 \ 0 \\
\hline
= \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 1 \ 0
\end{array}$$

The source and destination choose the same polynomial $G(x)$ called generator because it is used to generate the control bits (Checksum).

The algorithm for calculating the message to send is as follows. Let $M(x)$ be the polynomial corresponding to the original message, and let r be the degree of the generator polynomial $G(x)$ chosen:

- Multiply $M(x)$ by x^r , which amounts to adding r zeros to the end of the original message
- Perform the following division modulo 2:

$$\frac{M(x)x^r}{G(x)} = Q(x) + R(x)$$

- The quotient $Q(x)$ is ignored. The remainder $R(x)$ (Checksum) contains r bits (degree of the remainder $r - 1$). We then carry out the subtraction modulo 2:

$$M(x).x^r - R(x) = T(x)$$

The polynomial $T(x)$ is the cyclic polynomial: it is the message ready to be sent. The cyclic polynomial is a multiple of the generator polynomial $T(x) = Q(x).G(x)$

At reception, we carry out the following division:

$$\frac{T(x)}{G(x)}$$

- If the remainder = 0, there is no error
- If the remainder $\neq 0$, there is an error, so we must retransmit

By choosing $G(x)$ judiciously, we can detect any error on 1 bit, 2 consecutive bits, a sequence of n bits and beyond n bits with a very high probability.

Example

Either transmit the message 1011011 using the generator polynomial $G(x) = x^4 + x + 1$. We proceed as follows to calculate the message to be transmitted

1. original message = 1011011 $\Rightarrow M(x) = x^6 + x^4 + x^3 + x^1 + 1$

2. $G(x) = x^4 + x + 1$

3. $M(x).x^4 = x^{10} + x^8 + x^7 + x^5 + x^4$

4. Calculate $R(x)$ by polynomial division

$$\begin{array}{r|l}
 x^{10} + x^8 + x^7 + x^5 + x^4 & x^4 + x + 1 \\
 \underline{x^{10} + x^7 + x^6} & x^6 + x^4 + x^2 \\
 x^8 + x^6 + x^5 + x^4 & \\
 \underline{x^8 + x^5 + x^4} & \\
 x^6 & \\
 \underline{x^6 + x^3 + x^2} & \\
 x^3 + x^2 &
 \end{array}$$

5. $R(x) = x^3 + x^2 = (1100)_2$

6. The message to send $T(x) = M(x).x^r - R(x) = x^{10} + x^8 + x^7 + x^5 + x^4 - x^3 - x^2 = (10110111100)_2$

On reception, a similar calculation is carried out on the word received, but here the remainder must be zero. Otherwise, an error occurred along the way.

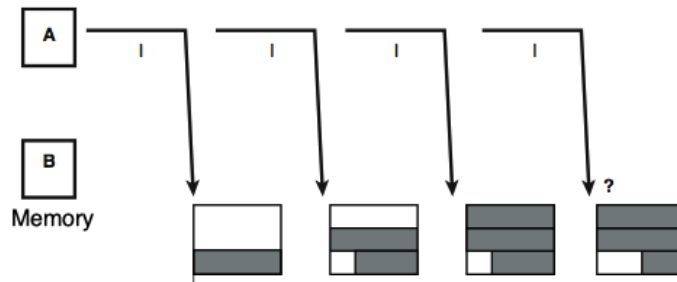
Polynomial codes used

The main generating polynomials (divisors) are:

- LRCC-8: $x^8 + 1$
- LRCC-16: $x^{16} + 1$
- CRC 12: $x^{12} + x^{11} + x^3 + x^2 + x + 1$
- CRC 16 Forward: $x^{16} + x^{15} + x^2 + 1$
- CRC 16 Backward: $x^{16} + x^{14} + x + 1$
- CRC CITT Forward: $x^{16} + x^{12} + x^5 + 1$
- CRC CITT Backward: $x^{16} + x^{11} + x^4 + 1$

4.3 Flow Control

In a transmission of information from a transmitter A to a receiver B, if the transmitter produces the data at a speed significantly higher than the consumption speed of the receiver, the latter will be congested (saturated or overloaded) and the information transmitted will be lost. To solve this problem, we can consider providing the receiver with a buffer memory allowing it to store messages while waiting for their processing. We can quickly see that whatever the size of the memory used, it can be saturated.



Flow control is used to set up a mechanism to control the rate at which information is sent to the receiver. It can be achieved by several methods:

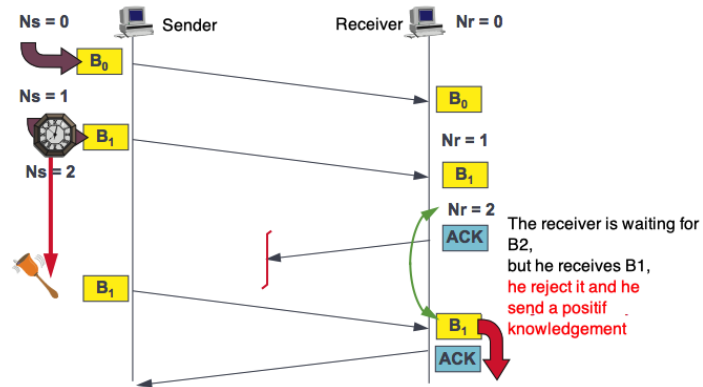
4.3.1 Sent and Wait

After sending a block of information, the transmitter stops waiting for an acknowledgment of receipt. Upon receipt of the acknowledgment, denoted ACK for Acknowledge, the transmitter sends the following block.

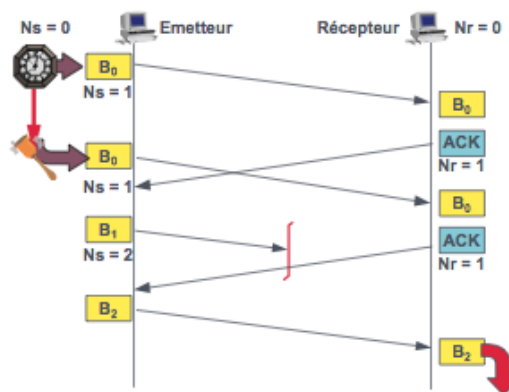
In the event of a transmission error, the received block is rejected. The block is said to be lost, it is not acknowledged. The transmitter then remains on standby. To avoid blocking transmission, when sending each data block, the transmitter sets a timer. At the end of the allotted time (Time Out), if no acknowledgment of receipt (ACK) has been received, the transmitter retransmits the unacknowledged block.

A difficulty arises if the loss concerns the ACK. Indeed, although the data has been correctly received, the transmitter retransmits it on a delay. The information is thus received twice. To avoid data duplication, it is necessary to identify the blocks. For this purpose, the transmitter and the receiver maintain counters N_s (N_s , Number transmitted, s for send) and N_r (Number of the block to receive, r for receive). Both counters are initialized to zero. The content of the counter N_s is transmitted with the block, the receiver compares this number with the content of its counter N_r . If the two values are

identical the block is deemed valid and accepted. If the values differ, the block received is not the expected one. It is rejected and acknowledged if it corresponds to a block already received.



In cases where consumption delays are longer, the data may not be acknowledged on time. For example, if A transmits a block B₀ and B delays in its processing, A will retransmit B₀ before receiving the acknowledgment. If A transmits a new block B₁ and is lost, it will consider the acknowledgment of the second B₀ as an acknowledgment of B₁.



To avoid this confusion of interpretation, it is also necessary to number the ACKs.

The wait time for acknowledgments makes the send and wait method inefficient. Additionally, it is unidirectional.

4.3.2 Look-ahead window

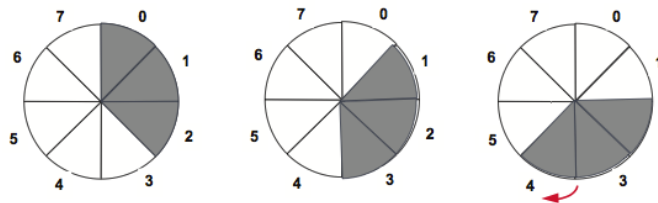
To improve the previous protocol and reduce the waiting time for acknowledgments, we send several blocks without waiting for the ACKs, this process is called anticipation. Thus, an acknowledgment no longer acknowledges a single frame but a set of frames which

follow one another without error. The number of successive frames that can be sent without receiving an acknowledgment is limited by a value denoted W , called window. The principle is to authorize the transmitter to send frames with a sequence number between the number r of the next expected frame (communicated by the receiver) and $r + W - 1$:

$$r \leq Ns \leq r + W - 1$$

Note: $W = 1$ in the case of a Send-and-Wait procedure.

To be able to resend the frames in the event of an error, the transmitter puts the unacknowledged blocks in W buffers. When receiving an acknowledgment of a frame, the transmitter frees the corresponding memory and sends a new frame.



Problem: what happens if, temporarily, the receiver is not ready to receive the W information frames from the window? The use of an anticipation window may require the implementation of an additional all-or-nothing regulation mechanism. The idea is that a particular control frame is used to indicate that the receiver is momentarily unable to continue receiving. The transmitter receiving this control frame immediately stops all transmission (even if it has not used its entire transmission "window"). Another control frame is then necessary to indicate to the transmitter that the receiver has returned to a normal state and is therefore ready to receive new frames.

4.4 Data management procedures

Data management procedures are link layer protocols that implement the previous techniques (frame delineation, error correction and flow control). They come from two families:

1. Character-oriented procedures: which generally work alternately (send and wait type).
2. Bit-oriented procedures: designed for simultaneous bidirectional transmissions at high speeds.

4.4.1 HDLC procedure

The HDLC (High-level Data Link Control) procedure is a bit-oriented procedure, developed by IBM and standardized by the ITU in 1976. HDLC is a full duplex point-to-point and multipoint procedure, using frames separated by flags of value 01111110 (7E). Three modes can be operated by HDLC:

1. Normal Response Mode (NRM): the secondary station must wait for an explicit order from the primary before being able to transmit.
2. asynchronous response mode (Asynchronous Response Mode or ARM): the secondary station has the right to transmit data without waiting for the invitation from the primary. This mode of operation is also known as LAP (Link Access Protocol). It assumes that both stations have both primary and secondary status.
3. the balanced or symmetrical asynchronous response mode (Asynchronous Balanced Mode or ABM): the link must be point-to-point; as for LAP, the two stations have both primary and secondary status. This mode of operation is also known as the LAP-B (Link Access Protocol-Balanced) protocol. Nowadays, this is the only mode used.

4.4.1.1 HDLC frame types

HDLC uses three frame types:

1. diinformation frames or **I** frames: ensure data transfer;
2. supervision frames or **S** frames (Supervisor): ensure the transmission of supervision commands (acknowledgment of receipt...),
3. unnumbered frames or **U** (Unnumbered) frames: supervise the connection (connection, disconnection).

4.4.1.2 HDLC frame structure

The HDLC frame is organized as follows:

8 bits	8 bits	8 bits	n bits	16 bits	8 bits
flag	address	control	information	FCS	flag

- Pennant (flag): 01111110

- . delimits frames: all frames must start and end with a flag.
 - . allows frame synchronization: all stations attached to the link must constantly search for this sequence;
 - . the same flag can serve as a closing flag for a frame and as an opening flag for the following frame;
 - . flag transparency mechanism by stuffing bits: in transmission, a 0 is inserted as soon as five consecutive 1s appear outside the F fields; these 0s are removed on reception. If seven 1s appear anywhere in a frame, it is declared in error.
- Address field: identifies the frame as a command or response. In ABM mode, the values that this field can take are predefined. Four values are sufficient to distinguish commands and responses in both directions of transmission (eg: 11000000, 10000000, 11110000 and 1110000).
 - Control field: it indicates the frame type with the necessary parameters.
 - FCS (Frame Check Sequence): calculated on the address, order and of information, from the polynomial code V.41 ($x^{16} + x^{12} + x^5 + 1$).

4.4.1.3 Control field and frame formats

There are three frame formats which correspond to different codings of the control field:

	1	2	3	4	5	6	7	8
Information	0	Ns			P/F	Nr		
Supervision	1	0	S	S	P/F	Nr		
Unnumbered	1	1	M	M	P/F	M	M	M

- Ns: sequence number of the transmitted frame,
- Nr: number of the next expected frame (acknowledgment in the data),
- P/F (Poll/Final): This bit is set to 1 by the primary when it requests an immediate response from the secondary.

The meaning of type S frames depends on the two SS bits according to the following table:

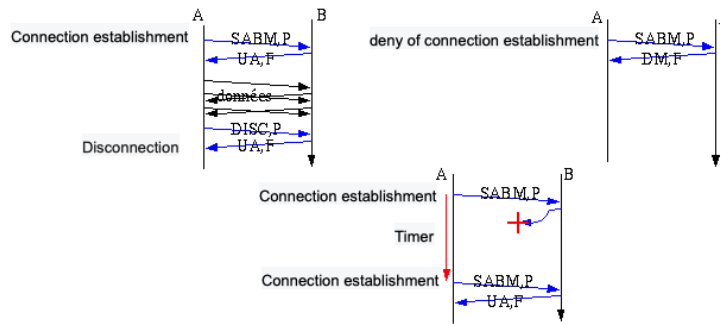
S	S	Command	Meaning
0	0	RR (Receiver Ready)	The station is ready to receive the frame number Nr and positively acknowledges the reception of frames up to (Nr - 1)
0	1	RNR (Receiver not Ready)	The station is not ready to receive frames but and positively acknowledges the reception of frames up to (Nr - 1)
1	0	REJ (Reject)	The station rejects the frames from number Nr. The sender is obliged to retransmit (P/F = 1)
1	11	SREJ (Reject)	= REJ but only for frame number Nr.

The meaning of type U frames depends on the two M bits according to the following table:

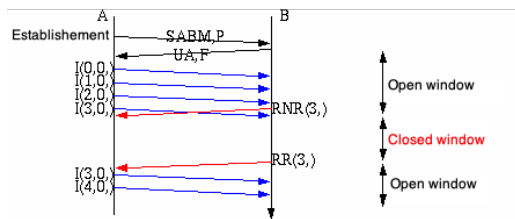
Frame	Command	Meaning
11111100	SABM	Set ABM requests establishment in ABM mode
11110000	DM	Disconnect Mode indicates station is located in offline mode
11001010	DISC	Disconnect releases link
11000110	UA	Unnumbered Acknowledge indicates reception acceptance of an unnumbered order

4.4.1.4 Examples of exchanges

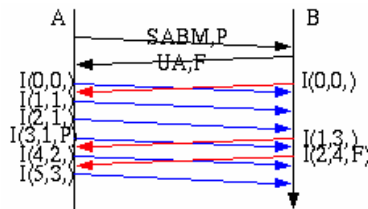
The following figure illustrates examples of exchanges between two stations using the HDLC procedure to establish the connection.



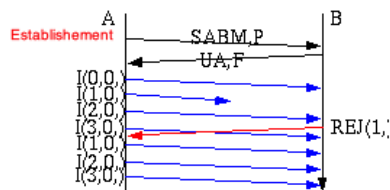
The following example illustrates a case of unidirectional exchange.



The following example illustrates a case of bidirectional exchange.



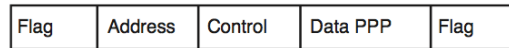
The following example illustrates a case of unidirectional exchange with loss of frames.



4.5 PPP Protocol

PPP is the point-to-point link protocol used in the Internet. It uses the subscriber's telephone lines to access the network (the connection typically involves a personal computer

and the Internet service provider). This is a very simplified version of HDLC which does not include flow control or error recovery mechanisms. The structure of a PPP frame is given in the following figure.



The 8 bits of the Address field are 1 (the link being point-to-point, a single address value is sufficient). The Control field has the same meaning as in HDLC. The Data PPP field begins with two bytes (the protocol field), which identify the higher-level protocol for which the frame is intended; it ends with an FCS field whose calculation mode is identical to that of an HDLC frame. The only frame carrying data on a reliable link is a U frame of type UI (Unnumbered Information). This frame contains an information field but is not numbered (because there is no flow control). The absence of an error recovery mechanism does not mean that the circuit is reliable: the FCS field is used to validate the frames received.

PPP includes three main components:

- A method for encapsulating datagrams from multiple protocols.
- A Link Control Protocol (LCP) intended to establish, configure, and test the data link.
- A family of Network Control Protocols (NCPs) for establishing and configuring multiple network layer protocols.

Two very well-known variants of the PPP protocol:

- **PPPoA** (point-to-point protocol over ATM) is a protocol used by ADSL and cable broadband connections intended for individuals.
- **PPPoE** (point-to-point protocol over Ethernet) is a PPP encapsulation protocol over Ethernet. It allows you to benefit from the advantages of PPP, in particular its compatibility with authentication protocols (PAP, CHAP, etc.) and connection control (speed, etc.), on an Ethernet network. It is widely used in broadband Internet connections via ADSL and cable intended for individuals.