

# High-Level Data Link Control

**High-Level Data Link Control** (HDLC) is a communication protocol used for transmitting data between devices in telecommunication and networking. Developed by the International Organization for Standardization (ISO), it is defined in the standard ISO/IEC 13239:2002.

HDLC ensures reliable data transfer, allowing one device to understand data sent by another. It can operate with or without a continuous connection between devices, making it versatile for various network configurations.

Originally, HDLC was used in multi-device networks, where one device acted as the master and others as slaves, through modes like Normal Response Mode (NRM) and Asynchronous Response Mode (ARM). These modes are now rarely used. Currently, HDLC is primarily employed in point-to-point connections, such as between routers or network interfaces, using a mode called Asynchronous Balanced Mode (ABM).

## High-Level Data Link Control

	<u>Communication protocol</u>
Abbreviation	HDLC
Purpose	Data framing
Developer(s)	International Organization for Standardization (ISO)
Introduction	1979
Based on	<u>SDLC</u>
OSI layer	<u>Data link layer</u>

## History

HDLC is based on IBM's SDLC protocol, which is the layer 2 protocol for IBM's Systems Network Architecture (SNA). It was extended and standardized by the ITU as LAP (Link Access Procedure), while ANSI named their essentially identical version ADCCP.

The HDLC specification does not specify the full semantics of the frame fields. This allows other fully compliant standards to be derived from it, and derivatives have since appeared in innumerable standards. It was adopted into the X.25 protocol stack as LAPB, into the V.42 protocol as LAPM, into the Frame Relay protocol stack as LAPF and into the ISDN protocol stack as LAPD.

The original ISO standards for HDLC are the following:

- ISO 3309-1979 – Frame Structure
- ISO 4335-1979 – Elements of Procedure
- ISO 6159-1980 – Unbalanced Classes of Procedure
- ISO 6256-1981 – Balanced Classes of Procedure

ISO/IEC 13239:2002, the current standard, replaced all of these specifications.

HDLC was the inspiration for the IEEE 802.2 LLC protocol, and it is the basis for the framing mechanism used with the PPP on synchronous lines, as used by many servers to connect to a WAN, most commonly the Internet.

A similar version is used as the control channel for E-carrier (E1) and SONET multichannel telephone lines. Cisco HDLC uses low-level HDLC framing techniques but adds a protocol field to the standard HDLC header.

## Framing

HDLC frames can be transmitted over synchronous or asynchronous serial communication links. Those links have no mechanism to mark the beginning or end of a frame, so the beginning and end of each frame has to be identified. This is done by using a unique sequence of bits as a frame delimiter, or *flag*, and encoding the data to ensure that the flag sequence is never seen inside a frame. Each frame begins and ends with a frame delimiter. A frame delimiter at the end of a frame may also mark the start of the next frame.

On both synchronous and asynchronous links, the flag sequence is binary "01111110", or hexadecimal 0x7E, but the details are quite different.

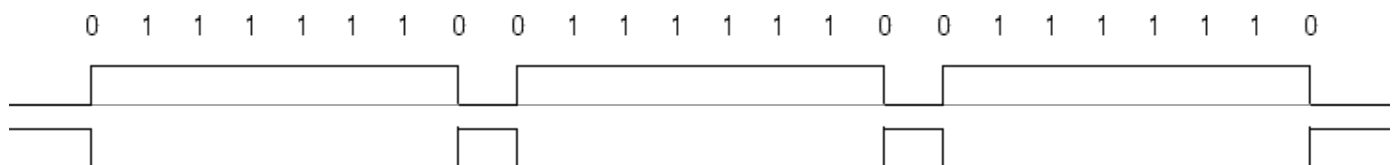
## Synchronous framing

Because a flag sequence consists of six consecutive 1-bits, other data is coded to ensure that it never contains more than five 1-bits in a row. This is done by bit stuffing: any time that five consecutive 1-bits appear in the transmitted data, the data is paused and a 0-bit is transmitted.

The receiving device knows that this is being done, and after seeing five 1-bits in a row, a following 0-bit is stripped out of the received data. If instead the sixth bit is 1, this is either a flag (if the seventh bit is 0), or an error (if the seventh bit is 1). In the latter case, the frame receive procedure is aborted, to be restarted when a flag is next seen.

This bit-stuffing serves a second purpose, that of ensuring a sufficient number of signal transitions. On synchronous links, the data is NRZI encoded, so that a 0-bit is transmitted as a change in the signal on the line, and a 1-bit is sent as no change. Thus, each 0 bit provides an opportunity for a receiving modem to synchronize its clock via a phase-locked loop. If there are too many 1-bits in a row, the receiver can lose count. Bit-stuffing provides a minimum of one transition per six bit times during transmission of data, and one transition per seven bit times during transmission of a flag.

When no frames are being transmitted on a simplex or full-duplex synchronous link, a frame delimiter is continuously transmitted on the link. This generates one of two continuous waveforms, depending on the initial state:



The HDLC specification allows the 0-bit at the end of a frame delimiter to be shared with the start of the next frame delimiter, i.e. "011111101111110". Some hardware does not support this.

For half-duplex or multi-drop communication, where several transmitters share a line, a receiver on the line will see continuous idling 1-bits in the inter-frame period when no transmitter is active.

HDLC transmits bytes of data with the least significant bit first (not to be confused with little-endian order, which refers to byte ordering within a multi-byte field).

## Asynchronous framing

When using asynchronous serial communication such as standard RS-232 serial ports, synchronous-style bit stuffing is inappropriate for several reasons:

- Bit stuffing is not needed to ensure an adequate number of transitions, as start and stop bits provide that,
- Because the data is NRZ encoded for transmission, rather than NRZI encoded, the encoded waveform is different,
- RS-232 sends bits in groups of 8, making adding single bits very awkward, and
- For the same reason, it is only necessary to specially code flag *bytes*; it is not necessary to worry about the bit pattern straddling multiple bytes.

Instead asynchronous framing uses "control-octet transparency", also called "byte stuffing" or "octet stuffing". The frame boundary octet is 01111110, (0x7E in hexadecimal notation). A "control escape octet", has the value 0x7D (bit sequence '10111110', as RS-232 transmits least-significant bit first). If either of these two octets appears in the

transmitted data, an escape octet is sent, followed by the original data octet with bit 5 inverted. For example, the byte 0x7E would be transmitted as 0x7D 0x5E ("10111110 01111010"). Other reserved octet values (such as XON or XOFF) can be escaped in the same way if necessary.

The "abort sequence" 0x7D 0x7E ends a packet with an incomplete byte-stuff sequence, forcing the receiver to detect an error. This can be used to abort packet transmission with no chance the partial packet will be interpreted as valid by the receiver.

## Structure

---

The contents of an HDLC frame are shown in the following table:

Flag	Address	Control	Information	FCS	Flag
8 bit	8 or more bits	8 or 16 bits	Variable length, $8 \times n$ bits	16 or 32 bits	8 bits

Note that the end flag of one frame may be (but does not have to be) the beginning (start) flag of the next frame.

Data is usually sent in multiples of 8 bits, but only some variants require this; others theoretically permit data alignments on other than 8-bit boundaries.

The frame check sequence (FCS) is a 16-bit CRC-CCITT or a 32-bit CRC-32 computed over the Address, Control, and Information fields. It provides a means by which the receiver can detect errors that may have been induced during the transmission of the frame, such as lost bits, flipped bits, and extraneous bits. However, given that the algorithms used to calculate the FCS are such that the probability of certain types of transmission errors going undetected increases with the length of the data being checked for errors, the FCS can implicitly limit the practical size of the frame.

If the receiver's calculation of the FCS does not match that of the sender's, indicating that the frame contains errors, the receiver can either send a negative acknowledge packet to the sender, or send nothing. After either receiving a negative acknowledge packet or timing out waiting for a positive acknowledge packet, the sender can retransmit the failed frame.

The FCS was implemented because many early communication links had a relatively high bit error rate, and the FCS could readily be computed by simple, fast circuitry or software. More effective forward error correction schemes are now widely used by other protocols.

## Types of stations (computers) and data transfer modes

---

Synchronous Data Link Control (SDLC) was originally designed to connect one computer with multiple peripherals via a multidrop bus. The original "normal response mode" is a master-slave mode where the computer (or **primary terminal**) gives each peripheral (**secondary terminal**) permission to speak in turn. Because all communication is either to or from the primary terminal, frames include only one address, that of the secondary terminal; the primary terminal is not assigned an address. There is a distinction between **commands** sent by the primary to a secondary, and **responses** sent by a secondary to the primary, but this is not reflected in the encoding; commands and responses are indistinguishable except for the difference in the direction in which they are transmitted.

**Normal response mode** allows the secondary-to-primary link to be shared without contention, because it has the primary give the secondaries permission to transmit one at a time. It also allows operation over half-duplex communication links, as long as the primary is aware that it may not transmit when it has given permission to a secondary.

**Asynchronous response mode** is an HDLC addition<sup>[1]</sup> for use over full-duplex links. While retaining the primary/secondary distinction, it allows the secondary to transmit at any time. Thus, there must be some other mechanism to ensure that multiple secondaries do not try to transmit at the same time (or only one secondary).

**Asynchronous balanced mode** adds the concept of a *combined terminal* which can act as both a primary and a secondary. Unfortunately, this mode of operation has some implementation subtleties. While the most common frames sent do not care whether they are in a command or response frame, some essential ones do (notably most unnumbered frames, and any frame with the P/F bit set), and the address field of a received frame must be examined to determine whether it contains a command (the address received is ours) or a response (the address received is that of the other terminal).

This means that the address field is not optional, even on point-to-point links where it is not needed to disambiguate the peer being talked to. Some HDLC variants extend the address field to include both source and destination addresses, or an explicit command/response bit.

## HDLC operations and frame types

Three fundamental types of HDLC frames may be distinguished:

- Information frames, or **I-frames**, transport user data from the network layer. They can also include flow and error control information piggybacked on data.
- Supervisory frames, or **S-frames**, are used for flow and error control whenever piggybacking is impossible or inappropriate, such as when a station does not have data to send. S-frames **do not** have information fields.
- Unnumbered frames, or **U-frames**, are used for various miscellaneous purposes, including link management. Some U-frames contain an information field, depending on the type.

### Control field

The general format of the control field is:

HDLC control fields

7	6	5	4	3	2	1	0	
N(R) Receive sequence no.		P/F	N(S) Send sequence no.		0	1		<b>I-frame</b>
N(R) Receive sequence no.		P/F	type		0	1		<b>S-frame</b>
type		P/F	type		1	1		<b>U-frame</b>

There are also extended (two-byte) forms of I and S frames. Again, the least significant bit (rightmost in this table) is sent first.

Extended HDLC control fields

15	14	13	12	11	10	9	8		7	6	5	4	3	2	1	0	
N(R) Receive sequence no.						P/F	N(S) Send sequence no.						0	1			<b>Extended I-frame</b>
N(R) Receive sequence no.						P/F	0	0	0	0	type	0	1				<b>Extended S-frame</b>

### P/F bit

Poll/Final is a single bit with two names. It is called Poll when part of a command (set by the primary station to obtain a response from a secondary station), and Final when part of a response (set by the secondary station to indicate a response or the end of transmission). In all other cases, the bit is clear.

The bit is used as a token that is passed back and forth between the stations. Only one token should exist at a time. The secondary only sends a Final when it has received a Poll from the primary. The primary only sends a Poll when it has received a Final back from the secondary, or after a timeout indicating that the bit has been lost.

- In NRM, possession of the poll token also grants the addressed secondary permission to transmit. The secondary sets the F-bit in its last response frame to give up permission to transmit. (It is equivalent to the word "Over" in radio voice procedure.)
- In ARM and ABM, the P bit forces a response. In these modes, the secondary need not wait for a poll to transmit, so the final bit may be included in the *first* response after the poll.
- If no response is received to a P bit in a reasonable period of time, the primary station times out and sends P again.
- The P/F bit is at the heart of the basic **checkpoint retransmission** scheme that is required to implement HDLC; all other variants (such as the REJ S-frame) are optional and only serve to increase efficiency. Whenever a station receives a P/F bit, it may assume that any frames that it sent before it last transmitted the P/F bit and not yet acknowledged will never arrive, and so should be retransmitted.

When operating as a combined station, it is important to maintain the distinction between P and F bits, because there may be two checkpoint cycles operating simultaneously. A P bit arriving in a command from the remote station is not in response to our P bit; only an F bit arriving in a response is.

## N(R), the receive sequence number

Both I and S frames contain a receive sequence number N(R). N(R) provides a positive acknowledgement for the receipt of I-frames from the other side of the link. Its value is always the first frame not yet received; it acknowledges that all frames with N(S) values up to N(R)-1 (modulo 8 or modulo 128) have been received and indicates the N(S) of the next frame it expects to receive.

N(R) operates the same way whether it is part of a command or response. A combined station only has one sequence number space.

## N(S), the sequence number of the sent frame

This is incremented for successive I-frames, modulo 8 or modulo 128. Depending on the number of bits in the sequence number, up to 7 or 127 I-frames may be awaiting acknowledgment at any time.

## I-Frames (user data)

Information frames, or **I-frames**, transport user data from the network layer. In addition they also include flow and error control information piggybacked on data. The sub-fields in the control field define these functions.

The least significant bit (first transmitted) defines the frame type. 0 means an I-frame. Except for the interpretation of the P/F field, there is no difference between a command I frame and a response I frame; when P/F is 0, the two forms are exactly equivalent.

## S-frames (control)

Supervisory Frames, or 'S-frames', are used for flow and error control whenever piggybacking is impossible or inappropriate, such as when a station does not have data to send. S-frames in HDLC **do not** have information fields, although some HDLC-derived protocols use information fields for "multi-selective reject".

The S-frame control field includes a leading "10" indicating that it is an S-frame. This is followed by a 2-bit type, a poll/final bit, and a 3-bit sequence number. (Or a 4-bit padding field followed by a 7-bit sequence number.)

The first (least significant) 2 bits mean it is an S-frame. All S frames include a P/F bit and a receive sequence number as described above. Except for the interpretation of the P/F field, there is no difference between a command S frame and a response S frame; when P/F is 0, the two forms are exactly equivalent.

### Receive Ready (RR)

- Bit value = 00 (0x00 to match above table type field bit order<sup>[2]</sup>)
- Indicate that the sender is ready to receive more data (cancels the effect of a previous RNR).
- Send this packet if you need to send a packet but have no I frame to send.
- A primary station can send this with the P-bit set to solicit data from a secondary station.
- A secondary terminal can use this with the F-bit set to respond to a poll if it has no data to send.

### Receive Not Ready (RNR)

- Bit value = 01 (0x04 to match above table type field bit order<sup>[3]</sup>)
- Acknowledge some packets but request no more be sent until further notice.
- Can be used like RR with P bit set to solicit the status of a secondary station
- Can be used like RR with F bit set to respond to a poll if the station is busy.

### Reject (REJ)

- Bit value = 10 (0x08 to match above table type field bit order<sup>[4]</sup>)
- Requests immediate retransmission starting with N(R).
- Sent in response to an observed sequence number gap; e.g. after seeing I1/I2/I3/I5, send REJ4.
- Optional to generate; a working implementation may use only RR.

### Selective Reject (SREJ)

- Bit value = 11 (0x0c to match above table type field bit order)
- Requests retransmission of only the frame N(R).
- Not supported by all HDLC variants.
- Optional to generate; a working implementation may use only RR, or only RR and REJ.

## U-Frames

Unnumbered frames, or **U-frames**, are primarily used for link management, although a few are used to transfer user data. They exchange session management and control information between connected devices, and some U-frames contain an information field, used for system management information or user data. The first 2 bits (11) mean it is a U-frame. The five type bits (2 before P/F bit and 3 bit after P/F bit) can create 32 different types of U-frame. In a few cases, the same encoding is used for different things as a command and a response.

### Mode setting

The various modes are described in § [Link configurations](#). Briefly, there are two non-operational modes (initialization mode and disconnected mode) and three operational modes (normal response, asynchronous response, and asynchronous balanced modes) with 3-bit or 7-bit (extended) sequence numbers.

### Disconnected mode (DM) response

When the secondary is disconnected (the default state on power-up), it sends this generic response to any poll (command frame with the poll flag set) except an acceptable mode setting command. It may alternatively give a FRMR response to an unacceptable mode set command.

#### **Unnumbered acknowledge (UA) response**

This is the secondary's response to an acceptable mode set command, indicating that it is now in the requested mode.

#### **Set ... mode (SNRM, SARM, SABM) command**

Place the secondary in the specified mode, with 3-bit sequence numbers (1-byte control field). The secondary acknowledges with UA. If the secondary does not implement the mode, it responds with DM or FRMR.

#### **Set ... mode extended (SNRME, SARME, SABME) command**

Place the secondary in the specified mode, with 7-bit sequence numbers (2-byte control field).

#### **Set mode (SM) command**

Generic mode set, new in ISO/IEC 13239, using an information field to select parameters. ISO/IEC 13239 added many additional options to HDLC, including 15- and 31-bit sequence numbers, which can only be selected with this command.

#### **Disconnect (DISC) command**

This command causes the secondary to acknowledge with UA and disconnect (enter disconnected mode). Any unacknowledged frames are lost.

#### **Request disconnect (RD) response**

This response requests the primary to send a DISC command. The primary should do so promptly, but may delay long enough to ensure all pending frames are acknowledged.

#### **Set initialization mode (SIM) command**

This rarely-implemented command is used to perform some secondary-specific initialization, such as downloading firmware. What happens in initialization mode is not otherwise specified in the HDLC standard.

#### **Request initialization mode (RIM) response**

This requests the primary to send SIM and initialize the secondary. It sent in lieu of DM if the secondary requires initialization.

### **Information transfer**

These frames may be used as part of normal information transfer.

#### **Unnumbered information (UI)**

This frame (command or response) communicates user data, but without acknowledgement or retransmission in case of error.

#### **UI with header check (UIH)**

This frame (command or response), a ISO/IEC 13239 addition and rarely used, is like UI but also excludes CRC protection. Only a configurable-length prefix ("header") of the frame is covered by the CRC polynomial; errors in the rest of the frame are not detected.

#### **Unnumbered poll (UP) command**

This command solicits a response from the secondary. With the poll bit set, it acts like any other poll frame, without the acknowledgement that must be included in I or S frame. With the poll bit clear, it has a special meaning in normal response mode: the secondary *may* respond, even though it has not received the poll bit. This is rarely used in HDLC, but was used in the original IBM SDLC as a substitute for the lack of asynchronous response mode; where the communication channel could handle simultaneous responses, the primary would periodically send UP to the broadcast address to collect any pending responses.

### **Error Recovery**

#### **Frame reject (FRMR) response**

The FRMR response contains a description of the unacceptable frame, in a standardized format. The first 1 or 2 bytes are a copy of the rejected control field, the next 1 or 2 contain the secondary's current send and receive sequence numbers (and a flag indicating that the frame was a response, applicable only in balanced mode), and the following 4 or 5 bits are error flags indicating the reason for the rejection. The secondary repeats the same FRMR response to every poll until the error is cleared by a mode set command or RSET. The error flags are:

- W: the frame type (control field) is not understood or not implemented.
- X: the frame type is not understood with a non-empty information field, but one was present.
- Y: the frame included an information field that is larger than the secondary can accept.
- Z: the frame included an invalid receive sequence number N(R), one which is not between the previously received value and the highest sequence number transmitted. (This error cannot be cleared by receiving RSET, but can be cleared by *sending* RSET.)
- V: the frame included an invalid send sequence number N(S), greater than the last number acknowledged plus the transmit window size. This error is only possible if a transmit window size smaller than the maximum has been negotiated.
- The error flags are normally padded with 0 bits to an 8-bit boundary, but HDLC permits frames which are not a multiple of a byte long.

### Reset (RSET) command

The RSET command causes a secondary to reset its receive sequence number so the next expected frame is sequence number 0. This is a possible alternative to sending a new mode set command, which resets both sequence numbers. It is acknowledged with UA, like a mode set command.

### Peer discovery

#### Exchange identification (XID)

An XID command includes an information field specifying the primary's capabilities; the secondary responds with an XID response specifying its capabilities. This is normally done before sending a mode set command. Systems Network Architecture defined one format for the information field, in which the most significant bit of the first byte is clear (0), but HDLC implementations normally implement the variant defined in ISO 8885, which has the most significant bit of the first byte set (1).

#### TEST

A TEST command is simply a ping command for debugging purposes. The payload of the TEST command is returned in the TEST response.

### Defined in other standards

There are several U frames which are not part of HDLC, but defined in other related standards.

#### Nonreserved (NR0, NR1, NR2, NR3)

The "nonreserved" commands and responses are guaranteed by the HDLC standard to be available for other uses.

#### Ack connectionless (AC0, AC1)

These are defined in the IEEE 802.2 logical link control standard.

#### Configure (CFGR)

This command was defined in SDLC for debugging. It had a 1-byte payload which specified a non-standard test mode for the secondary. Even numbers disabled the mode, while odd numbers enabled it. A payload of 0 disabled all test modes. The secondary normally acknowledged a configure command by echoing it in response.

#### Beacon (BCN) response

This response was defined in SDLC to indicate a communications failure. A secondary which received no frames at all for a long time would begin sending a stream of beacon responses, allowing a unidirectional fault to be located. Note that ISO/IEC 13239 assigns UIH the same encoding as BCN.

## Link configurations

---

Link configurations can be categorized as being either:

- *Unbalanced*, which consists of one primary terminal, and one or more secondary terminals.
- *Balanced*, which consists of two peer terminals.

The three link configurations are:



- *Normal Response Mode* (NRM) is an unbalanced configuration in which only the primary terminal may initiate data transfer. The secondary terminals transmit data only in response to commands from the primary terminal. The primary terminal polls each secondary terminal to give it an opportunity to transmit any data it has.
- *Asynchronous Response Mode* (ARM) is an unbalanced configuration in which secondary terminals may transmit without permission from the primary terminal. However, there is still a distinguished primary terminal which retains responsibility for line initialization, error recovery, and logical disconnect.
- *Asynchronous Balanced Mode* (ABM) is a balanced configuration in which either station may initialize, supervise, recover from errors, and send frames at any time. There is no master/slave relationship. The DTE (data terminal equipment) and DCE (data circuit-terminating equipment) are treated as equals. The initiator for Asynchronous Balanced Mode sends an SABM.

An additional link configuration is *Disconnected mode*. This is the mode that a secondary station is in before it is initialized by the primary, or when it is explicitly disconnected. In this mode, the secondary responds to almost every frame other than a mode set command with a "Disconnected mode" response. The purpose of this mode is to allow the primary to reliably detect a secondary being powered off or otherwise reset.

## HDLC Command and response repertoire

---

The minimal set required for operation are:

- Commands: I, RR, RNR, DISC, and one of SNRM, SARM or SABM
- Responses: I, RR, RNR, UA, DM, FRMR

### Basic operations

- Initialization can be requested by either side. When the primary sends one of the six mode-set commands, it:
  - Signals the other side that initialization is requested
  - Specifies the mode, NRM, ABM, ARM
  - Specifies whether 3 or 7 bit sequence numbers are in use.

The HDLC module on the other end transmits (UA) frame when the request is accepted. If the request is rejected it sends (DM) disconnect mode frame.

### Functional extensions (options)

- For Switched Circuits
  - Commands: ADD – XID
  - Responses: ADD – XID, RD
- For 2-way Simultaneous commands & responses are ADD – REJ
- For Single Frame Retransmission commands & responses: ADD – SREJ
- For Information Commands & Responses: ADD – UI
- For Initialization
  - Commands: ADD – SIM
  - Responses: ADD – RIM
- For Group Polling
  - Commands: ADD – UP
- Extended Addressing
- Delete Response I Frames

- Delete Command I Frames
- Extended Numbering
- For Mode Reset (ABM only) Commands are: ADD – RSET
- Data Link Test Commands & Responses are: ADD – TEST
- Request Disconnect. Responses are ADD – RD
- 32-bit FCS

## HDLC command and response repertoire

Type Of Frame	Name	Command/Response	Description	Info	C-Field Format							
					7	6	5	4	3	2	1	0
Information(I)		C/R	User exchange data		N(R)			P/F	N(S)			0
Supervisory (S)	Receive Ready (RR)	C/R	Positive Acknowledgement	Ready to receive I-frame N(R)	N(R)			P/F	0	0	0	1
	Receive Not Ready (RNR)	C/R	Positive Acknowledgement	Not ready to receive	N(R)			P/F	0	1	0	1
	Reject (REJ)	C/R	Negative Acknowledgement	Retransmit starting with N(R)	N(R)			P/F	1	0	0	1
	Selective Reject (SREJ)	C/R	Negative Acknowledgement	Retransmit only N(R)	N(R)			P/F	1	1	0	1

## Unnumbered frames

Unnumbered frames are identified by the low two bits being 1. With the P/F flag, that leaves 5 bits as a frame type. Even though fewer than 32 values are in use, some types have different meanings depending on the direction they are sent: as a command or as a response. The relationship between the **DISC** (disconnect) command and the **RD** (request disconnect) response seems clear enough, but the reason for making **SARM** command numerically equal to the **DM** response is obscure.

Name	Command/ Response	Description	Info	C-Field Format							
				7	6	5	4	3	2	1	0
Set normal response mode <b>SNRM</b>	C	Set mode	Use 3 bit sequence number	1	0	0	P	0	0	1	1
SNRM extended <b>SNRME</b>	C	Set mode; extended	Use 7 bit sequence number	1	1	0	P	1	1	1	1
Set asynchronous response mode <b>SARM</b>	C	Set mode	Use 3 bit sequence number	0	0	0	P	1	1	1	1
SARM extended <b>SARME</b>	C	Set mode; extended	Use 7 bit sequence number	0	1	0	P	1	1	1	1
Set asynchronous balanced mode <b>SABM</b>	C	Set mode	Use 3 bit sequence number	0	0	1	P	1	1	1	1
SABM extended <b>SABME</b>	C	Set mode; extended	Use 7 bit sequence number	0	1	1	P	1	1	1	1
Set Mode <b>SM</b>	C	Set mode, generic	New in ISO 13239	1	1	0	P	0	0	1	1
Set initialization mode <b>SIM</b>	C	Initialize link control function in the addressed station		0	0	0	P	0	1	1	1
Request initialization mode <b>RIM</b>	R	Initialization needed	Request for <b>SIM</b> command	0	0	0	F	0	1	1	1
Disconnect <b>DISC</b>	C	Terminate logical link connection	Future I and S frames return <b>DM</b>	0	1	0	P	0	0	1	1
Request disconnect <b>RD</b>	R	Solicitation for <b>DISC</b> Command		0	1	0	F	0	0	1	1
Unnumbered acknowledgment <b>UA</b>	R	Acknowledge acceptance of one of the set-mode commands.		0	1	1	F	0	0	1	1
Disconnect mode <b>DM</b>	R	Responder in disconnected mode	Mode set required	0	0	0	F	1	1	1	1
Unnumbered information <b>UI</b>	C/R	Unacknowledged data	Has a payload	0	0	0	P/F	0	0	1	1
UI with header check <b>UIH</b>	C/R	Unacknowledged data	New in ISO 13239	1	1	1	P/F	1	1	1	1
Unnumbered poll <b>UP</b>	C	Used to solicit control information		0	0	1	P	0	0	1	1
Reset <b>RSET</b>	C	Used for recovery	Resets N(R) but not N(S)	1	0	0	P	1	1	1	1
Exchange identification <b>XID</b>	C/R	Used to Request/Report capabilities		1	0	1	P/F	1	1	1	1
Test <b>TEST</b>	C/R	Exchange identical information fields for testing		1	1	1	P/F	0	0	1	1
Frame reject <b>FRMR</b>	R	Report receipt of unacceptable frame		1	0	0	F	0	1	1	1
Nonreserved 0 <b>NR0</b>	C/R	Not standardized	For application use	0	0	0	P/F	1	0	1	1
Nonreserved 1 <b>NR1</b>	C/R	Not standardized	For application use	1	0	0	P/F	1	0	1	1
Nonreserved 2 <b>NR2</b>	C/R	Not standardized	For application use	0	1	0	P/F	1	0	1	1
Nonreserved 3 <b>NR3</b>	C/R	Not standardized	For application use	1	1	0	P/F	1	0	1	1
Ack connectionless, seq 0 <b>AC0</b>	C/R	Not part of HDLC	IEEE 802.2 LLC extension	0	1	1	P/F	0	1	1	1
Ack connectionless, seq 1 <b>AC1</b>	C/R	Not part of HDLC	IEEE 802.2 LLC extension	1	1	1	P/F	0	1	1	1
Configure for test <b>CFGR</b>	C/R	Not part of HDLC	Was part of SDLC	1	1	0	P/F	0	1	1	1
Beacon <b>BCN</b>	R	Not part of HDLC	Was part of SDLC	1	1	1	F	1	1	1	1

## HDLC U frames, by binary encoding

C-Field Format								Command	Response
0	1	2	3	4	5	6	7		
1	1	0	0	P/F	0	0	0	UI	
1	1	0	0	P/F	0	0	1	SNRM	
1	1	0	0	P/F	0	1	0	DISC	RD
1	1	0	0	P/F	0	1	1	SM <sup>*</sup>	
1	1	0	0	P/F	1	0	0	UP	
1	1	0	0	P/F	1	0	1	(unused)	
1	1	0	0	P/F	1	1	0		UA
1	1	0	0	P/F	1	1	1	TEST	
1	1	0	1	P/F	0	x	x	NR0–NR3	
1	1	0	1	P/F	1	x	x	(unused)	
1	1	1	0	P/F	0	0	0	SIM	RIM
1	1	1	0	P/F	0	0	1		FRMR
C-Field Format								Command	Response
0	1	2	3	4	5	6	7		
1	1	1	0	P/F	0	1	0	(unused)	
1	1	1	0	P/F	0	1	1	CFGR <sup>†</sup>	
1	1	1	0	P/F	1	0	x	(unused)	
1	1	1	0	P/F	1	1	x	AC0–AC1 <sup>†</sup>	
1	1	1	1	P/F	0	0	0	SARM	DM
1	1	1	1	P/F	0	0	1	RSET	
1	1	1	1	P/F	0	1	0	SARME	
1	1	1	1	P/F	0	1	1	SNRME	
1	1	1	1	P/F	1	0	0	SABM	
1	1	1	1	P/F	1	0	1	XID	
1	1	1	1	P/F	1	1	0	SABME	
1	1	1	1	P/F	1	1	1	UIH <sup>*</sup>	
									BCN <sup>†</sup>

\* <sup>^</sup> <sup>^</sup> ISO/IEC 13239 addition

<sup>†</sup> <sup>^</sup> <sup>^</sup> <sup>^</sup> Not part of HDLC

The UI, UIH, XID, TEST frames contain a payload, and can be used as both commands and responses. The SM command and FRMR response also contain a payload.

- A UI frame contains user information, but unlike an I frame it is neither acknowledged nor retransmitted if lost.
- A UIH frame (an ISO/IEC 13239 addition) is like a UI frame, but additionally applies the frame check sequence only to a specified-length prefix of the frame; transmission errors after this prefix are not detected.
- The XID frame is used to exchange terminal capabilities. Systems Network Architecture defined one format, but the variant defined in ISO 8885 is more commonly used. A primary advertises its capabilities with an XID command, and a secondary returns its own capabilities in an XID response.
- The TEST frame is simply a ping command for debugging purposes. The payload of the TEST command is returned in the TEST response.
- The SM command (an ISO/IEC 13239 addition) is a generic "set mode" command which includes an information field (in the same ISO 8885 format as XID) specifying parameters. This allows parameter values (like 15- and 31-bit sequence numbers) and parameters like window sizes and maximum frame sizes not expressible by the standard six mode-set commands to be negotiated.
- The FRMR response contains a description of the unacceptable frame, in a standardized format. The first 1 or 2 bytes are a copy of the rejected control field, the next 1 or 2 contain the secondary's current send and receive sequence numbers, and the following 4 or 5 bits are error flags indicating the reason for the rejection.

## See also

- Point-to-Point Protocol
- Serial Line Internet Protocol

- Self-synchronizing code

## Notes

---

1. (Friend et al. 1988, p. 191)
2. "X.25 frame types - Martin Baker" ([http://www.euclideanspace.com/coms/protocol/x25/link/f\\_types/index.htm#rr](http://www.euclideanspace.com/coms/protocol/x25/link/f_types/index.htm#rr)).
3. "X.25 frame types - Martin Baker" ([http://www.euclideanspace.com/coms/protocol/x25/link/f\\_types/index.htm#nr](http://www.euclideanspace.com/coms/protocol/x25/link/f_types/index.htm#nr)).
4. "X.25 frame types - Martin Baker" ([http://www.euclideanspace.com/coms/protocol/x25/link/f\\_types/index.htm#rej](http://www.euclideanspace.com/coms/protocol/x25/link/f_types/index.htm#rej)).

## References

---

- Friend, George E.; Fike, John L.; Baker, H. Charles; Bellamy, John C. (1988). *Understanding Data Communications* (2nd ed.). Indianapolis: Howard W. Sams & Company. ISBN 0-672-27270-9.
- Stallings, William (2004). *Data and Computer Communications* ([https://archive.org/details/datacomputercomm00stal\\_1](https://archive.org/details/datacomputercomm00stal_1)) (7th ed.). Upper Saddle River: Pearson/Prentice Hall. ISBN 978-0-13-100681-2.
- S. Tanenbaum, Andrew (2005). *Computer Networks* ([https://archive.org/details/isbn\\_9788177581652](https://archive.org/details/isbn_9788177581652)) (4th ed.). 482,F.I.E., Patparganj, Delhi 110 092: Dorling Kindersley(India)Pvt. Ltd.,licenses of Pearson Education in South Asia. ISBN 81-7758-165-1.

## External links

---

- *PPP in a Real-time Oriented HDLC-like Framing*. RFC 2687 (<https://tools.ietf.org/html/rfc2687>).
- *PPP in HDLC-like Framing* (<https://www.rfc-editor.org/info/std51>). RFC 1662 (<https://tools.ietf.org/html/rfc1662>). STD 51.
- Data Communication Lectures of Manfred Lindner – Part HDLC ([http://www.ict.tuwien.ac.at/lva/384.081/infobase/L03-HDLC\\_v4-4.pdf](http://www.ict.tuwien.ac.at/lva/384.081/infobase/L03-HDLC_v4-4.pdf))
- HDLC packet format and other information ([http://www.acacia-net.com/wwwcla/protocol/iso\\_4335.htm#l%20format](http://www.acacia-net.com/wwwcla/protocol/iso_4335.htm#l%20format))
- 
- ISO 3309:1984 Information Processing Systems—Data Communication—High Level Data Link Control Procedures—Frame Structure (<https://law.resource.org/pub/in/bis/S04/is.11418.1.1986.pdf>) (archived (<https://archive.org/details/gov.in.is.11418.1.1986>))
- ISO 4335:1984 Data Communication—High Level Data Link Control Procedures—Consolidation of Elements of Procedures (<https://law.resource.org/pub/in/bis/S04/is.11418.2.1986.pdf>) (archived (<https://archive.org/details/gov.in.is.11418.2.1986>))
- ISO/IEC 13239:2002 (<https://www.iso.org/standard/37010.html>)

---

Retrieved from "[https://en.wikipedia.org/w/index.php?title=High-Level\\_Data\\_Link\\_Control&oldid=1223273295](https://en.wikipedia.org/w/index.php?title=High-Level_Data_Link_Control&oldid=1223273295)"

-