

Data link control requirements are discussed and summarized. A generalized structure for a data link control capable of meeting those requirements is presented. Synchronous data link control (SDLC) is given as a solution, evolving from the generalized structure, to meet the requirements stated. Finally, the significant attributes of SDLC are discussed and summarized.

Synchronous data link control: A perspective

by R. A. Donnan and J. R. Kersey

This paper presents some perspectives on synchronous data link control (SDLC), a new data link control discipline, which possesses the important property of performing its function without necessarily being apparent at higher levels of network control. Several levels of detail are treated in the paper, including a discussion of features common to a generalized structure for bit-oriented data link control procedures and also some specific aspects of SDLC.

This paper should be of interest to data communications network planners, network performance analysts, access method programmers, and network application programmers among others. One objective of the paper is to provide a common perspective from which the many people in a data processing/data communications organization may view this new data link control discipline.

Data link control requirements

The purpose of data link control is to provide a grammar by which two or more machines may converse, or exchange information, over a data link in an efficient and reliable manner. To meet this purpose, certain requirements need to be met by the data link control.

basic requirements

The basic requirements of any data link control include those requirements that are necessary to transmit information between

senders and receivers on a data link and to protect the integrity of that information. If information transfer were always between two stations over a point-to-point dedicated facility which was error free, then only a rudimentary data link control would be required. However, the data communications environment does, and will continue to, include multipoint and switched facilities, and those facilities as well as point-to-point facilities are susceptible to introducing errors. Thus, a minimum set of requirements for a data link control are considered herein to be: (1) A means of delimiting the beginning and ending of information to distinguish it from noise and to signal when the checking mechanism is to be active. (2) A means of identifying the sender and receiver from among several on a multipoint facility or the very many connectable through a switched facility. (3) A means of detecting for errors and initiating a corrective action. Typical means of detecting errors are by algorithms based on format and by various forms of redundancy such as vertical, longitudinal, or cyclic redundancy checks. The corrective action is usually done via retransmission although forward error correction using additional redundancy could be employed.

Many different data link controls could have an architecture designed to meet these basic requirements. However, merely meeting the basic requirements would not meet all the various needs existing in even a small set of today's data communication environments and applications. What is needed, then, is a data link control architecture that meets the requirements of the largest possible set of environments and applications.

Before specifying the generalized structure of a data link control, it is necessary to consider the requirements that are imposed by all the several elements of the system such as the applications, the communications facilities, and other equipment employed to economically meet the needs of those applications. Applications may have needs that directly impose requirements on the data link control. Also some applications may have needs that can only be met by certain facilities and/or equipment. These facilities and/or equipment may, in turn, impose additional requirements on the data link control. Typical application requirements are discussed below.

**general
requirements**

Some application requirements

Conversational applications are typified by relatively balanced traffic from and to an operator-oriented terminal. The traffic is composed of messages of fairly short length with operator participation on a record-by-record basis. Each exchange of records represents a completed unit which is considered by the system to be functionally independent from other such exchanges.

conversational

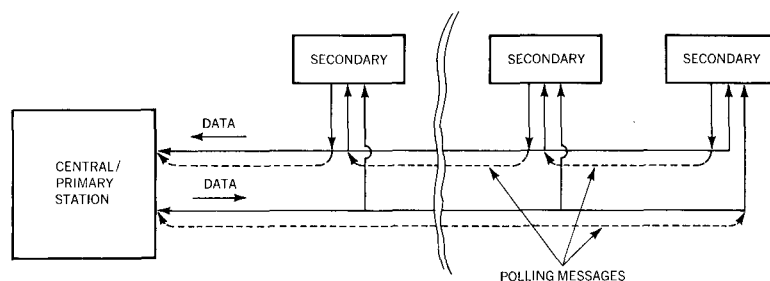
Since the amount of information used for data link control (DLC) can become a significant percentage of the total traffic, optimization of data link control overhead is an important requirement. Relatively fast response time is also a requirement, but consistency of response time is even more significant in order to maintain user satisfaction. Since the operator may become involved in procedures for recovering from system-detected errors, automatic error recovery is less of a requirement than for some other applications, say, batch applications. Whether recovery is mostly automatic or mostly reliant on the operator will depend on trade-offs between hardware and software costs versus operator convenience and human factors considerations.

inquiry Inquiry applications are typified by relatively short input messages and relatively long output messages; thus, there is a traffic imbalance. It would not be unusual for this input-to-output traffic ratio to reach one to four or even larger imbalances. The requirements imposed on the DLC by inquiry applications are much the same as for conversational applications; however, an advantage can be obtained if a second inquiry can be accepted while the response to the first inquiry is being outputted. This implies using a particular form of duplex transmission, i.e., receiving input from one station on a multi-point link while simultaneously sending output to another station. Such a form of duplex operation has been termed "multi-multipoint". Involvement by the operator in recovery procedures from system-detected errors can be expected.

batch Batch applications (e.g., terminal-to-processor remote job entry) are typified by relatively large quantities of input and/or output data per job. Because of this, it is less important that the number of DLC characters used for control (resulting in overhead) be at an absolute minimum since they represent a smaller percentage of the total traffic. Further, the effects of turnaround delays are less significant than they are for short length messages (blocks); thus, half duplex facilities are usually employed. Batch applications are usually expected to operate without direct operator participation on a record or transmission block basis. Thus, the requirement for automatic detection of and recovery from errors is greater than in applications that are more closely attended by an operator.

processor to processor In processor-to-processor applications, the traffic, in many cases, is relatively balanced; however, the traffic pattern may be highly variable and dependent on the division of function between the processors for the particular job being performed. The effects of DLC overhead tend to be less significant than in most of the other cases mentioned, but there is a high requirement for recovery from system-detected errors to be fully automatic and for operator intervention to be held to an absolute minimum.

Figure 1 Hub-go-ahead link



Some special facility requirements

As was mentioned previously, the use of certain types of facilities may be imposed by application requirements or by system economics. The nature of these facilities may, then, impose requirements on the DLC. For example, if dial-up facilities are employed, it is desirable to verify that the intended station has been reached upon completion of a connection over the switched link. This implies a requirement for an "I am —; who are you?" kind of function in the DLC.

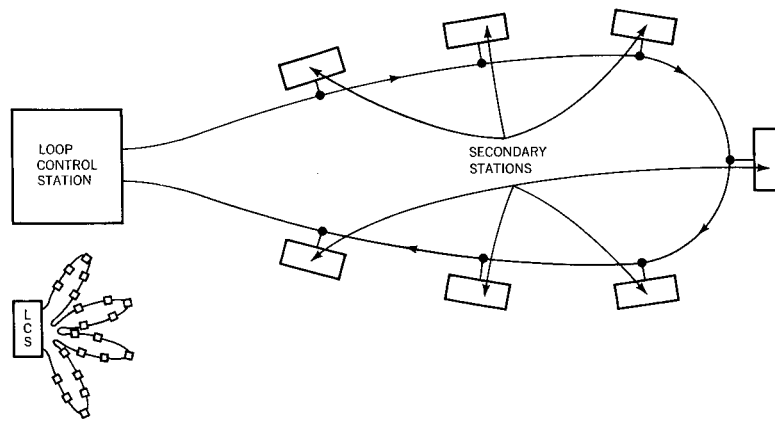
Another clear example is presented by earth-satellite communication circuits. Because of the long propagation delays inherent in such a facility, it could be mandatory that a higher modulus for transmission block numbering be employed than is needed for a typical land-based communications facility. This larger modulus, along with two-way simultaneous control capability in the DLC, would be required for efficient use of the facility.

long
propagation
delays

In certain real-time applications where fast response time to a large number of terminals on long links is desired, it is sometimes desirable to use a hub link and hub-go-ahead polling. A hub link (see Figure 1) uses a two-way simultaneous transmission facility, with each station connected to the facility via one and one-half modems such that signals transmitted by a station on the input line can be received by each "less-remote" station (with respect to the location of the central station). In hub-go-ahead, the central station polls the most remote station (via the output line); this station then transmits all messages it has accumulated since its last transmitting period. At the end of its transmission, the transmitting station sends a go-ahead signal to the next-less-remote station on the input line. This process repeats until the closest station has completed transmission. The closest station then sends a go-ahead signal to the central station which again starts the cycle of scanning with the most remote station. Meanwhile, the central station may be sending output messages to any of the stations. The main advantage of hub-go-ahead lies

hub
links

Figure 2 Loop transmission link



in lessening the number of line turnarounds and the DLC overhead for polling in a scan of the stations. This is particularly significant for applications with short messages and a fast response time requirement. The impacts on DLC requirements include:

1. A "hub-poll" type of control command to go from the primary to the secondary stations.
2. A "go-ahead" signal to be passed from a secondary station to the next "less-remote" secondary station.
3. Capability of receiving inputs from multiple stations in response to a single hub-go-ahead poll command and the resultant "go-ahead" signals from secondary to secondary.
4. Additional error-recovery procedures, etc.

loops

A loop transmission facility (see Figure 2) is one in which a communications link, originating at a loop controller station, passes through one or more secondary stations in a serial fashion, one after the other, and then terminates back at the receiving side of the loop controller station. Logically, the communications link makes a circular path between all stations connected to the link including the loop control station. The physical path of the link may, for reliability considerations, be made nodal (e.g., four-leaf clover) with several returns to the control station to facilitate link reconfiguration in the event of a line failure at some point in the loop. Loop operation also makes some requirements on the DLC. These include:

1. A "loop-poll" type of control command which may be accepted or passed on by each secondary station in succession.
2. A "go-ahead" signal.
3. Some particular procedures with regards to a station's "right-to-transmit" that are different from normal "roll-call" or "hub" poll procedures.

4. Capability of receiving and distinguishing inputs from multiple stations in response to a single loop poll command.
5. Different (from those in half-duplex, multipoint links) error recovery procedures.

Summary of major requirements

Many other examples could be given of requirements that are imposed by either applications, facilities, or other influences. The following is a summary of the major requirements. They are in no particular order of significance. Full justification for the requirements is not given.

- Control of the data link should remain with a central station to facilitate hub-go-ahead and loop operation as well as to simplify error recovery procedures on all types of links.
- Transmission block length should be independent of message or record length.
- The DLC should allow straightforward transmission of any message (bit pattern) of any number of bits in length, i.e., it should not be possible for any bit pattern within a transmission to be mistakenly construed as a control character.
- The DLC should not impose unnecessary line turnarounds.
- Operation on systems with a wide range of line speeds and propagation delays should be possible without significant loss of achievable line utilization.
- The basic DLC structure should be equally applicable to half-duplex, duplex, reverse channel, hub, or loop link configurations.
- A simple form of the DLC should be provided for use with simple stations, with more complex forms available for more intelligent stations, but upward compatibility and common link cohabitability of the different forms should be maintained. (Cohabitability is the ability of several DLC architectures to function simultaneously over a common communications facility.)
- The amount of redundancy for checking purposes should be variable to meet the needs of diverse applications and facilities.
- A common discipline should be used for recovery from detected errors in the communications link; recovery should, to the greatest degree possible, be capable of being automatic.
- The addressing and control structures of the DLC should be open-ended. That is, the structure should permit an increase in the number of stations to be addressed and new functional capabilities to be added while retaining functional compatibility and link cohabitability with prior implementations of a previous version of the basic DLC. Further, this extendability should be accomplished without affecting programs already written to operate over the link.

- The DLC should minimize the down time of all elements of the system over which DLC may exert an influence. Particularly, it should not be required to complete time-consuming error recovery procedures with a given station on a multi-drop link prior to exchanging traffic with another station on that link.
- The DLC should provide a means of obtaining status information from the various system elements concerned with DLC to allow the total link to be serviceable and maintainable.

A generalized structure for DLC

The structure defined here is a general DLC intended to meet the requirements summarized above. Prior to presenting the specific architecture of SDLC, it seems worthwhile to review this structure to demonstrate its inherent flexibility.

All transmissions on a data link will take the following form, defined as a frame:

F [A] [C] [Information] [BC] [F]

where:

[] = optional field
 F = Flag
 A = Station Address
 C = Control Field
 BC = Block Check

The flag F is defined as a unique sequence not to be found (see Note) in a transmission in any way other than as a frame delimiter. F is always required to initiate transmissions to separate information from line noise. (Note: The way in which one insures that this unique sequence is not to be found elsewhere is discussed later when we test the specific SDLC embodiment of the more generalized structure we are now discussing.)

This unique flag allows a DLC structure that is both bit-oriented and positionally significant. The structure is bit-oriented in that the length and content of all fields (A, C, I, BC) of a frame can be independent of any code set or character size (e.g., six-, seven-, or eight-bit codes). This is in contrast to previous DLCs that were character-oriented; that is, the elements of the DLC were required to be of integer character length, on character-bound areas, and represented in a given code such as EBCDIC or ASCII. The structure is positionally significant in that one can take the next k bits following the unique delimiter F (where k need not be on a character boundary) as having particular mean-

ing. In the generalized structure presented here, n bits are taken as an address field and m bits are taken as a control field; thus $k = n + m$. A similar positional significance of j bits preceding the ending flag is taken as a block check (BC) field. The result is a DLC structure that is insensitive to both code set and character size.

The station address A identifies the outlying station (relative to the central station) that is to receive an outgoing transmission or that originated an incoming transmission. If a specific architecture evolving from the generalized structure is to be implemented by a system that never expects to operate on any configuration other than point-to-point dedicated lines (where the stations are known a priori), then the address field should be used. When used, the station address field, A , must be the first n bits following the leading flag sequence, F . The length of the A field, n , once chosen, must remain fixed for a given architecture subset. Other architecture subsets may, however, choose n of a longer length if proper exclusions are made to allow for the addresses of architecture subsets with smaller n .

The command field C in outgoing transmissions (called the response field in incoming transmissions) is required in all specific architectures evolving from this generalized structure when implied controls or implied responses to those controls are inadequate.

The C field contents may have any desired definition, depending on the requirements of the specific architecture. When used, the C field must be immediately preceded by the address field, and it must be the next m bits following the n bits of the A field. The length of the C field, m , once chosen, must remain fixed for a given architecture subset. Other architecture subsets may, however, choose m of a longer length if proper considerations are made for the control field of architecture subsets with smaller m .

The information field I is an arbitrary bit string (both length and pattern are unconstrained). To simplify implementations, specific architectures evolving from this generalized structure may place restrictions on the freedom possible in the I field, e.g., byte size restricted to eight bits is a likely restriction in some specific architectures. The I field is optional, and the contents of the C field may explicitly or implicitly define whether or not the I field exists in a given frame.

The block check field BC permits specific architectures to define varying amounts of redundancy for checking as required for a given set of conditions—environment, facilities, application, etc. The BC field is optional, as is its length, but it occupies the last k bits preceding the end of the frame. Architecture subsets using

different checking algorithms can, without interference, coexist on a common communications link. This is not to say that such differences should be encouraged for that would introduce incompatibilities that prevent interchanges between stations using the differing implementations. The point here is that this generalized structure does permit that degree of flexibility—a point worth consideration if mere line cohabitation should become a prime requisite.

The trailing flag sequence *F* (identical to the leading flag sequence) is required when undefined variable length blocks may be transmitted. Alternate approaches (which would not require a trailing flag) include fixed length transmission blocks or variable length blocks containing a count.

To summarize, this generalized DLC has a bit-oriented, positionally significant structure with:

- A unique flag sequence used to delimit the beginning and end of a transmission block.
- An optional address field in the first *n* bits following the beginning delimiter.
- An optional control/response field in the next *m* bits after the *n*-bit address field following the beginning delimiter. If the control/response field is present, the address field must be present.
- An optional information field immediately following the previously listed fields.
- An optional block check field in the last *k* bits preceding the ending delimiter if used, otherwise serving as the last *k* bits in a transmission of known length.
- An optional ending delimiter.

A significant feature of this structure is that data communications stations that are built to implement a number of DLC architectures following this structure can successfully cohabit a common communications facility.

Development of SDLC

We now turn from the discussion of a generalized structure to the specific development of SDLC. In looking back at the evolution of SDLC, we cannot say, with full objectivity, that what was intended at the outset was a new data link control. There were, however, three primary objectives.

The first primary objective was to integrate into a single DLC the capability for efficient: (1) two-way simultaneous transmission of both data and control, (2) interactive operation for both in-

quiry and conversational systems, (3) efficient batch type operation, (4) operation on communications facilities with significant propagation delays (e.g., facilities involving earth satellites), and (5) operation on half-duplex communications facilities.

While all of these capabilities have existed separately or in various combinations in prior DLCs, they have not all been previously integrated into a single DLC. For example, the combining of batch and interactive operations into a single DLC, above all others, required a departure from the previously used technique (for batch operation) of establishing a logical connection (via polling or selection) between two stations that remained connected for perhaps several exchanges of transmissions until the logical connection was broken by a subsequent DLC operation. This technique is sometimes referred to as "select hold" from the concept of holding an established logical connection until directed otherwise. The overhead imposed by such a technique (including the accompanying line turnarounds on half-duplex facilities) is inefficient for interactive operation.

The second primary objective was to improve certain characteristics that are only too often viewed with apathy and taken for granted. Yet, these characteristics can usually, with enthusiasm, be made specific and realizable in a nonmundane manner: They include improved reliability, availability, serviceability, and flexibility. How these objectives are met in SDLC and what, specifically, was improved will be discussed in a later section.

The third primary objective was to have the individual capabilities of the integrated DLC to be no less attractive than these same capabilities were in previous DLCs. It would do little good toward achieving a unified multi-purpose line control if particular functions and requirements met by previous DLCs were not included or were done in a less desirable (e.g., less efficient, etc.) manner. Particularly, the inclusion of a two-way simultaneous transmission and efficient interactive operation should not detrimentally affect half-duplex batch-type operations.

Scope of SDLC

SDLC is a data link control for serial-by-bit synchronous transmission between buffered stations on a data transmission link using centralized control. The data transmission link may be customer owned, leased, or switched facilities connected in half-duplex (HDX), full-duplex (FDX), HUB, or LOOP configurations. SDLC provides a means by which information can be exchanged between an outlying station (called a secondary station) and one central station (called the primary station). Format requirements for transmission block framing, station addressing, commands/re-

sponses with associated sequence numbers or modifiers are included. Specific commands, responses, and associated modifiers are defined and assigned for those communication channel functions required to support normal operation as well as exception conditions and corresponding recovery procedures. Further, SDLC includes a method of detecting errors in transmissions and a method to correct those detected errors via retransmission.

SDLC does not include a method for delimiting records, messages, or any other units of data other than the information fields within frames. Furthermore, device addressing, device control, device status, and end-to-end control functions are all beyond the scope of SDLC. Also, SDLC architecture does not encompass procedures for the establishment, maintenance, or termination of a switched channel connection between stations or for the exchange of supervisory signals (hand-shaking) between modems.

Concepts of SDLC

SDLC has certain underlying concepts which are particularly significant. These concepts provide the opportunity for SDLC to realize the objectives and requirements previously stated.

Basic to the architecture of SDLC is the concept that a data link control provides a common grammar by which machines may transfer information over a data link. SDLC grammar has an architecture designed so that it is not dependent on the characteristics of the particular input/output devices serving as either source or sink for the information exchanged. Also, SDLC grammar does not place restrictions on the content of the information being exchanged. That is, what has been called transparency in previous DLCs is inherent in the architecture of SDLC. Further, SDLC grammar is divorced from the organization of the information. Stated another way, the grammar applies to the data link only; delineation of information is restricted to blocking for transmission purposes. Other delineation of information such as start or end of message for contextual reasons (analogous to sentence and paragraph of text) is not taken as a DLC function. Still further, SDLC grammar remains basically unchanged (although perhaps supplemented) regardless of the differences in the transmission facility (switched, nonswitched, HDX, FDX, HUB, LOOP, etc.).

A second fundamental concept of SDLC is that once the data link has accepted information for transfer, it should deliver that information to the other end of the link without error and in the form it was accepted. The responsibility of the data link for transfer of the information ends when the information has been accurately transferred to the receiving end. In this context, the

link is a logical entity consisting of much more than a physical communications channel. For example, the link should include all of the detection and recovery mechanisms associated with error control for providing accurate delivery of information from sender to receiver.

A third fundamental concept of SDLC is that each link is controlled by a single station called a primary station. Every link—point-to-point, multipoint, hub, or loop—has one and only one primary station. All transmissions, regardless of the direction of information flow, are initiated or authorized by the primary station by issuing commands to the other stations on the link called secondary stations. Secondary stations originate transmissions in response to, or as a result of, commands from the primary station. These secondary station transmissions include appropriate acknowledgments for the transmissions received from the primary station and may, under proper conditions, contain text for the primary station. The primary station is responsible for traffic management on the link (what station is allowed to send at a given time), recovery from errors that occur within the link, and collection of any required operational statistics. One or more secondary stations may be attached to the link, but transmission is always between a particular secondary station and the designated primary station. Both primary and secondary stations may incorporate the capability for alternate transmit-receive (HDX) and/or simultaneous transmit-receive (FDX).

A fourth fundamental concept of SDLC is that all transmissions on the link have a specific format called a *frame*. A frame is delimited by a unique sequence, called a FLAG, at the beginning and end of each transmission block. A frame contains, between FLAGS, a station address field, a control field, an information field and a block check field. The advantages of this format using bit-oriented positional significance (as opposed to multiformat character significance) was alluded to previously in the presentation of the generalized structure and will be discussed further below.

Specific structure

The specific format for SDLC transmission blocks is derived directly from the generalized structure previously presented. In SDLC, all transmission blocks, or frames, have this specific structure:

bit	0---7	8---15	16---23	i	$j-23$ $j-8$	$j-7$ j
	F	A	C	[Information]	BC	F

All fields, except the information field, are required in all transmissions. The length of the transmission block is at all times equal to $(48 + i)$ bits where i is the length of the information field. The length i may be zero or greater. While there is no theoretical limit to i , there is naturally a practical limit based on the capabilities of the checking algorithm, the error characteristics of the transmission channel, and the buffering capacity of the individual stations.

FLAG Information transfer in SDLC is characterized by isochronous transmission and synchronous detection of all of the bits within the frame; the individual frames may vary in length depending on the length of the I field.

Since variable length frames are transmitted and the start of any frame cannot be predicted, a method must be provided to uniquely signal the start and end of each frame. This is done by using a unique sequence of bits that are prevented from being inadvertently duplicated in the information stream. This sequence is referred to as a FLAG and consists of a binary zero followed by six contiguous binary one bits followed, in turn, by a single binary zero bit (01111110).

All frames will begin with a minimum of one FLAG; however, depending on delays or other causes, a number of FLAG sequences may actually be transmitted prior to the start of the actual frame. All frames will also end with the FLAG sequence. All stations connected to a telecommunication channel will, unless transmitting, continuously monitor the channel for the appearance of the FLAG sequence. This has been called a "continuous hunt system," but the FLAG sequence along with its station address(es) is the only thing for which a station must continuously hunt.

The uniqueness of the FLAG sequence does not occur naturally, but must be provided by special procedures.

When transmitting: A station monitors the sequence of bits transmitted between FLAGS. If a contiguous sequence of five binary one bits is noted, the transmitting station must automatically insert a binary zero bit into the information stream. As a result of this "zero-bit insertion" technique, there will be no more than five contiguous one bits transmitted within the frame. Note that the zero-bit insertion technique applies to the address, control, information, and block check fields; that is, all bits between, but not including, the FLAGS.

When receiving: A station inspects the bit following any contiguous sequence of five binary one bits. If this bit is a binary zero,

the receiving station deletes it from the information stream prior to presenting the information to the receiving sink. If the bit in question is a binary one, the sequence is either a FLAG sequence or an error. When a sixth binary one is received, the receiving station examines the next received bit, and if it is a binary zero, the receiving station accepts the total combination (01111110) as an end FLAG and proceeds with the frame-check algorithm. If the bit is a binary one, the receiving station rejects the frame.

The first eight bits following a beginning FLAG contain the secondary station address and are included in all frames from both primary and secondary stations. In transmissions to secondary stations, the field designates which secondary station (or stations in the case of a group or broadcast address) is to receive the frame. In transmissions from a secondary station, the field designates the secondary station from which the frame originated. The address is handled as an eight-bit entity and may be used to refer to a single station or a group of stations.

**station
address**

A secondary station must recognize its valid address or, in some cases, any one of several valid addresses prior to accepting a frame and taking any subsequent action on the contents of the frame. A secondary station may have more than one address for receiving; that is, it may be expected to receive broadcast transmissions by recognizing an "all stations" address; it may be part of one or more groups and thus required to recognize one or more group addresses; and, it will always have its own unique address. However, for transmitting, any frame initiated by a secondary station will utilize its unique station address. Also, for leased point-to-point or multipoint, the primary station will accept a frame only if it contains a secondary station address that is expected, i.e., the address of a secondary station that has been given prior permission to transmit by the primary station through the use of some particular command. However, in hub-go-aheads and loops, more than one secondary station may respond as a result of a group poll. (See the later discussion on asynchronous response mode.) Further, in the case of a switched environment, the primary station must be prepared to accept a multiplicity of valid addresses (or a "don't-care" address) for initialization purposes.

The second byte following a beginning F is the control field. The primary station uses the control field to indicate to secondary stations: (a) information transfer, (b) supervisory commands, and (c) nonsequenced commands. A secondary station uses the control field to indicate: (a) information transfer, (b) supervisory responses, and (c) nonsequenced responses.

**control
field**

There are three basic formats for the control field:

Information Transfer (I)	FI	N_s	P/F	N_r
Supervisory (S)	FI	S	P/F	N_r
Nonsequenced (NS)	FI	M	P/F	M

where

FI = Format Identifier (1 bit for I format, 2 bits for S and NS formats)

N_s = Transmitting station send sequence count (3 bits)

P/F = Poll bit for primary station transmissions (1 bit),
= Final frame bit for secondary station transmissions (1 bit)

N_r = Transmitting station receive sequence count (3 bits)

S = Supervisory function bits (2 bits)

M = Modifier function bits (5 bits)

The transmitting station send sequence count (N_s), accumulated modulo eight, provides a sequence count for each I-format frame that is transmitted from that station whether primary or secondary. The count is reset only upon receipt of a mode change. (Modes are discussed later.) The receiving station verifies the sequence count of each frame received as part of the error detection and recovery procedures.

The transmitting station receive sequence count (N_r), also modulo eight, is similar to the N_s count except it is incremented for each valid, in-sequence, error-free, I-format frame that is received by that station, whether primary or secondary. This count confirms to the remote station that all frames through number $N_r - 1$ have been received and accepted and that frame number N_r must be received next to satisfy the I format sequence count. The send and receive sequence counts (N_s and N_r) are relative to a send-receive pair of terminals. A secondary station can only send and receive relative to the primary station; hence, a secondary station need maintain only one set of send/receive sequence counts. A primary station, which may send to or receive from all stations on a multipoint link, must maintain a separate and independent set of send/receive sequence counts for each secondary station on the link.

The poll bit is used by the primary station to authorize or initiate secondary station transmission. Only one frame with the poll bit on should be outstanding on a link at any point in time; that is, once a poll frame is sent from the primary station, no additional polling frames should be transmitted until a response to the previous poll frame is received. Once a secondary station has initiated transmission, it may continue to transmit one or more frames to the primary station until one of the following conditions occur:

1. The secondary station has transmitted a maximum of seven frames (the maximum permitted by the modulo eight count) without having received acknowledgments from the primary station. Outstanding frames cannot exceed the modulus minus 1 without leading to ambiguity in acknowledgment. For example, if the secondary station transmits frames with $N_s = 0 - 7$ (eight frames) and the primary station acknowledges with $N_r = 0$, the secondary cannot distinguish between these two cases: (1) first frame transmitted was in error, (2) all frames were correctly received.
2. The secondary station has no more traffic to send.
3. The secondary station is commanded by the primary station to terminate transmission (applicable only to full-duplex secondary stations).

The last frame of a transmission from a secondary station is always indicated by the final bit being on in the control field.

The supervisory function bits (used only in an S-format frame) are used by the primary station to: (a) inhibit secondary stations from sending I frames, (b) request secondary stations to retransmit I frames, and (c) acknowledge the receipt of I frames from secondary stations. A secondary station uses the supervisory function bits to: (a) indicate that it cannot accept additional I frames from the primary station until certain conditions are cleared, (b) request retransmission of I frames from the primary station, and (c) acknowledge receipt of I frames from the primary station.

When the supervisory function bits are used (S format), the frame does not include an information field. Further, the transmission and/or receipt of frames having the supervisory format do not increment the N_s or N_r sequence counts.

The modifier function bits (used only in a nonsequenced format frame) are used to perform various other data link control functions and to provide a means for information transfer without regard to sequence counts. Sequence counts are unchanged by the transmission or receipt of nonsequenced format frames.

Frames using the modifier function bits are acknowledged by the secondary station only if received without error and with the poll bit on.

Control functions performed by the modifier bits include: (a) information transfer without sequence numbering (as might be used for group or broadcast messages), (b) placing the addressed secondary station in one of several operational modes, (c) polling of stations without affecting sequence numbers, (d) exchanging of identification information, and (e) placing the secondary station "on hook" in a switched communications link environment.

Response functions performed by the modifier bits include: (a) information transfer without sequence numbering, (b) acknowledging nonsequenced commands, (c) requesting certain initialization procedures from the primary station (e.g., placed in a "logical on-line" condition), (d) rejecting a command from the primary station resulting from a system programming or hardware problem (When a command is rejected, status information to specify the reason for the reject is included in the information field following the control field.), and (e) diagnostic-related responses.

The remaining bits in the control field, the format identifier bits, define the format of the control field; that is, how those bits (other than the P/F bit whose definition remains constant in all three formats) are to be interpreted.

**information
field**

An information field is not necessarily included in all frames. In fact, a frame may have an I field only if it is of the "I" format or "NS" format. Further, NS format frames may include an I field only if they are for: (a) information transfer without sequence numbering, (b) exchange of identification information, and (c) status information accompanying a rejected command.

When present, the information field immediately follows the control field and continues up to, but does not include, the block check field. The length of the information field is restricted only by buffering constraints of the stations involved in the information transfer and by the usual considerations of transmission block length due to communications channel error characteristics.

The information field may contain any bit sequence configuration (i.e., full transparency is the normal condition) to convey header information, control information, status, text (user data), etc. The content of the information field should be defined by actual or implied information included in the frame.

All SDLC frames include a block check (BC) field for the purpose of detecting errors that may occur during transmission. The checking is based on the transmission of redundant information in the form of a remainder polynomial numerator R derived from a division of the transmitted data by a generator polynomial; that is:

$$\frac{P}{G} = Q + \frac{R}{G}$$

where

P is the transmitted data polynomial

G is the fixed generator polynomial

Q is the whole polynomial quotient

R is the remainder polynomial numerator

block
check
field

The checking accumulation is initiated by the first bit following the beginning FLAG and includes all bits up to, but not including, the ending FLAG *except* those zero bits inserted by the transmitter and deleted by the receiver as a result of the occurrence of five contiguous one bits in the transmitted bit stream (to prevent unwanted FLAGS).

Modes can be viewed as means by which different sets of rules can be invoked. Modes ease the specifying of predetermined ("canned") rules with resulting improvements in efficiency.

modes

The "normal" mode for operation of secondary stations in SDLC is for all inputs from them to be solicited by the primary station. However, there are several operational conditions and link configurations (LOOP and FDX point-to-point), which have a requirement to allow a secondary station to initiate asynchronous inputs to the primary station. Further, there are sometimes situations in which a secondary station may not be able to obey all of the rules of protocol for normal operation and needs help to do so. These requirements are met in SDLC by defining three modes in which a secondary station may exist and/or be commanded to assume.

Normal Response Mode is a secondary station operational mode in which the secondary station initiates transmission *only* as the result of receiving a poll-type frame from the primary station.

Asynchronous Response Mode is a secondary station operational mode in which a secondary station may begin transmission without prior receipt of a poll-type frame from the primary station. (This transmission is authorized by virtue of the secondary station being previously placed into Asynchronous Response

Mode as opposed to a transmission being explicitly initiated as just previously described for Normal Response Mode.) On a loop link, multiple secondary stations may be in Asynchronous Response Mode simultaneously. Thus, an asynchronous transmission on that link may originate with any of those stations in Asynchronous Response Mode but in an orderly sequence as the "go-ahead" signal progresses around the loop. An asynchronous transmission may contain single or multiple frames that may contain pending traffic and/or a report of a status change at the secondary station. For half-duplex secondary stations, the last frame of a transmission in Asynchronous Response Mode must be indicated by the final frame bit in the control field. For full-duplex secondary stations, use of the final frame bit is less restrictive, but the details of that are beyond the scope of this paper.

When a secondary station has sent a complete transmission in Asynchronous Response Mode, that secondary station is responsible for successful transmission; that is, it must continue to retransmit it at periodic intervals until a primary station frame is received that acknowledges receipt of the asynchronous transmission.

Initialization Mode is a secondary station mode for those cases where normal operation within either of the other modes is not possible. Particularly, it allows the primary station to establish basic initialization functions in a secondary station. The initialization may be accomplished either from a local capability at the secondary station, or the initialization data may be transmitted to the secondary station while in Initialization Mode. Due to the special nature of this mode, the normal requirements for response characteristics may vary, i.e., response time, established protocol, etc., during the time of information transfer. This mode should be used only for minimum "boot-strapping" sufficient to restore/initialize a secondary station to a level of capability that permits operation in either of the other modes.

Significant attributes of SDLC

As we examine SDLC retrospectively, there are a few attributes that stand out as being more significant than others. Further, these fundamental attributes have at least one common quality—simplicity. At this time, we review these attributes and examine the benefits derived from them.

SDLC has a unique frame delimiter called a FLAG. This unique frame delimiter also functions as a unique means of synchronization. FLAG, along with the bit-insertion mechanism that provides the uniqueness of FLAG, accomplishes or contributes to several objectives.

FLAG being unique means that a receiving station need not follow the information on a communications line; it need only hunt for the sequence FA, where A is its station address. Previous DLCs required all stations on a link to continuously follow the information on the link in order to be aware of such things as: (a) the link being in control mode or text mode, (b) the appearance of block check fields (in order to not interpret their contents as possible control characters), and (c) when to look for new character synchronization.

The bit-insertion mechanism that is used to provide the uniqueness of FLAG also provides that there will not be long strings of consecutive ones on an active link. Further, by using zero-complement differential encoding, there will not be long strings of zeros within a transmission. Together, these two mechanisms eliminate a requirement for synchronization patterns to be inserted in a data stream to yield transitions for clocking.

The requirement to search for and find synchronization patterns within certain time-interval tolerances in a received data stream is eliminated. Particularly, the "synchronization receive" time-out function is eliminated and no transmissions are aborted from being received because of nonreceipt of synchronization within the time-out interval.

FLAG being unique and serving as the beginning and ending delimiter of a frame permits the information field within a frame to be any length and contain characters of any width.

An abort function can be defined as the reception of eight contiguous one bits; that is, a violation of the zero-bit insertion rule. The abort function is used by a transmitting station to terminate a frame prematurely. A specific example for use of the abort function would be following an equipment malfunction that precludes the validity of the transmission, e.g., buffer parity error on read-out.

Since in the body of a frame there will never be more than five contiguous one bits, if inter-frame intervals on an active link are filled with FLAGS, the transmission or reception of sixteen one bits can be defined as an idle state.

Given that FLAG detection is defined as the reception of a zero followed by six ones followed by a zero, then the special case of zero followed by seven ones can be defined to have special significance such as a "go-ahead" signal on a loop-configured link.

SDLC has one common format for all transmissions, i.e., F, A, C, [Info], BC, F. The advantages of a single common format re-

sult, ultimately, in simpler implementation design. Specifically, the positional significance of the address, control, and block check fields relative to the delimiting FLAGS results in straightforward logical decoding without complications from exception conditions or situations. Further, the information field, when present, starts after the sixteen bits following the beginning FLAG and ends sixteen bits prior to the ending FLAG. This, then, allows higher levels of control beyond data link control (such as network control) to be defined with positional significance relative to the delimiting FLAGS.

SDLC has complete functional separation. That is, the address, control, information, and block check fields are each distinct, separable entities with no interdependence. For example, the contents of the address field is not used as a modifier to the control field as in some DLCs where the address modifies a control to mean either POLL or SELECT, (e.g., prefix ENQ). Further, functional separation is also intended to mean that SDLC is data link control only and does not include other levels of addressing or control such as the addressing of a particular I/O device at a station or the controlling of those I/O devices in any way. Likewise, SDLC addresses and controls do not apply to networks in any way other than to a single link. Device or network addressing or control are additional levels of control in a hierarchical sense that may be contained in the information field of an SDLC frame.

SDLC checks all transmissions. As the format indicates, a block check field is always present irrespective of the contents of the frame. In many instances a frame is used only for control or response purposes and the information field will not be present. This results in a sixteen-bit BC field checking the sixteen bits of the address and control fields. While this, at first, seems extremely wasteful, the simplicity of a single format and a single checking algorithm along with the reliability advantage of all transmissions being fully checked are adequate reasons for justifying such redundancy.

The control discipline of SDLC is centralized. In all instances, the primary station either directly initiates a transmission from a secondary station or otherwise authorizes the secondary station to transmit—perhaps at some later time as determined by the secondary station. When the primary station directly initiates a secondary station transmission, the responsibility for retry or other error recovery procedures rests with the primary station. For the case where the primary station authorizes a secondary station to initiate a transmission at will, the responsibility for retry of that initiated transmission rests with the secondary station. However, once a single frame from the secondary station is acknowledged by the primary station, then retry responsibility

reverts back to the primary station. In either case, responsibility for error recovery procedures beyond initial retry remains with the primary station. Such an assignment of error recovery responsibility places the requirement for intelligence at a single point per link and at the point on the link which can best afford to expend such intelligence.

SDLC employs what is termed "non-select-hold". This means that a station is selected by the A field to receive a transmission only for the duration of the frame containing that station's address. Since each frame contains an address, there is no holding of a selection over several transmissions as is done in most DLCs prior to SDLC. This attribute is significant for at least two reasons.

1. Non-select-hold permits more efficient operation in an interactive environment and shorter response times when multiple stations share a single half-duplex link. The number of line turnarounds is reduced to just that number required for transmission of text (queries and answers); no line turnarounds are incurred solely for control purposes.
2. A multidropped link need not be devoted to a particular station for the duration of what could be lengthy error recovery procedures.

Finally, (and probably the most important attribute) the basic structure of SDLC is viable in a multitude of operational environments such as point-to-point or multipoint, half-duplex or duplex, or hub-go-ahead or loop configurations, and on links with short or long propagation delays.

Summary

SDLC is a data link control for serial-by-bit synchronous transmission between buffered stations on a data transmission link. A number of functional capabilities that have not heretofore been encompassed by any single data link control are provided along with improvements in many of the intangible characteristics.

The functional capabilities provided by SDLC include:

1. Two-way simultaneous transmission of data and/or controls, i.e., duplex operation, as well as half-duplex operation.
2. Operation on loop and hub-go-ahead configuration data transmission links.
3. Efficient operation in environments with interactive communications as well as batch type operations and on links with long propagation delays.
4. Independence of the DLC from the code(s) being transmitted on the link.

Improvements of intangible characteristics include:

1. Reliability is achieved in a direct and simple manner by including checking on all transmissions.
2. Availability is improved by the use of a "non-select-hold" discipline. Specifically, the full resources of a link, including all stations, are available (accessible) to the control station at all times insofar as data link control is involved. ERPs can run concurrently with, and be interleaved with, traffic to other stations.
3. Serviceability is improved by the inclusion of status information from a secondary station when required to inform the primary station of conditions at the secondary station. Further, there are specific means included in the C field of a response transmission to facilitate diagnostic procedures when unusual conditions exist.
4. Flexibility is achieved by having functional separation between delimiters, addresses, controls, information, and checks. Also the use of mode setting commands adds flexibility to the protocol invoked between the primary and secondary stations.

Generally, the requirements which were previously summarized have been met with SDLC.