

## Data Link Control in Data Communication

Alluri Sri Amith Varma<sup>1</sup>, Changanti Veera Sankara Sarma<sup>2</sup>, Pathalapu Sai  
Madan kumar<sup>3</sup>, Thumati Ravi<sup>3</sup>

<sup>1,2,3</sup> Final Year B.Tech, Dept. of ECE, KL University, Vaddeswaram, AP, India

<sup>4</sup> Associate Professor B.Tech, Dept. of ECE, KL University, Vaddeswaram, AP, India

---

**Abstract:** In this paper, Data link control in data communication is discussed. The control done by the Data link layer is Data link control. It mainly does three controls that are its functions and their advantages.

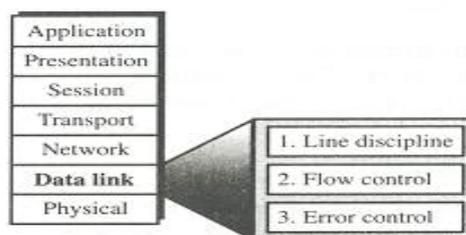
---

### I. Introduction

DLC is the service provided by the Data Link layer of function defined in the Open Systems Interconnection OSI model for network communication. The Data Link layer is responsible for providing reliable data transfer across one physical link (or telecommunications path) within the network. Technical communication systems are developed to transmit messages from a sender to a receiver beside the task of sending and receiving information over a channel, there are many other tasks a communication system has to do. One important task is to guarantee the correctness of the received information. No existing channel is completely error-free. In order to receive correct information, redundancy is often sent. On the receiver side the redundancy is needed to correct errors which have occurred while transmission Data link layer receives data from next higher layer adds some control bits i.e. frames are formed at this layer frame is handed over to the physical layer Data link layer removes the control bits and check for errors If there is no error, it hands over the received data to the physical layer The specified set of rules and procedures for carrying out data link control functions is called as Data Link protocol. Data-Link layer ensures that an initial connection has been set up, divides output data into data frames, and handles the acknowledgments from a receiver that the data arrived successfully. It also ensures that incoming data has been received successfully by analyzing bit patterns at special places in the frames.

### II. Functions Of Data Link Control

Data link layer due to possibility of transmission errors receiver may need to control the rate at which data arrive so. Synchronization and interfacing techniques are not sufficient .It is necessary to have a layer of control in each communicating device that provides certain functions The most important responsibilities of the data link layer are line discipline, flow control and error control. Collectively, these functions are known as data link control.



- **Line discipline** coordinates the link systems. It determines which device can and when it can send.
- **Flow control** coordinates the amount of data that can be sent before receiving acknowledgment. It also provides the receiver's acknowledgment of frame received intact and so is linked to error control.
- **Error control** means error detection and correction. It allows the receiver to inform the sender of any frames lost or damaged in transmission and coordinates the retransmission of those frames by the sender.

### III. Line Discipline

Whatever the system, no device in it should be allowed to transmit until that device has evidence that the intended receiver is able to receive and is prepared to accept the transmission. if the receiving device does not expect a transmission, it is busy or is out of commission With no way to determine the status of the intended receiver the transmitting device may waste its time sending data to a nonfunctioning receiver or may interfere with signals already on the link. The line discipline functions of the data link layer over see the establishment of links and the right of a particular device to transmit at a given time.

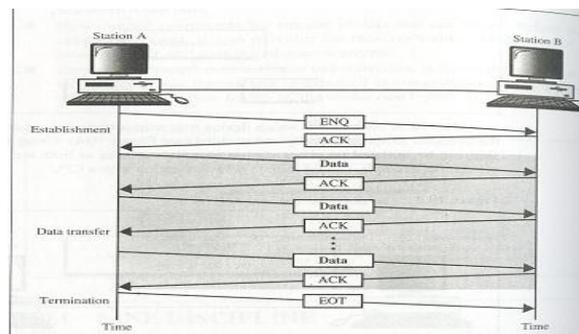
Line discipline can be done in two ways enquiry/acknowledgment. (ENQ/ACK) and poll/select. The first method is used in peer-to-peer communication: the second method is used in primary-secondary communication.

### ENQ/ACK

Enquiry/**acknowledgment (ENQ/ACK)** is used in systems where there is no question of the wrong receiver getting the transmission, that is, when there is a dedicated link between two devices so that the only device capable of receiving the transmission is the intended one.

In both half-duplex and full-duplex transmission, the initiating device establishes the session. In half-duplex, the initiator then sends its data while the responder waits. The responder may take over the link when the initiator is finished or has requested a response. In full-duplex, both devices can transmit simultaneously once the session has been established.

**WORKING:** The initiator first transmits a frame called an enquiry (ENQ) asking if the receiver is available to receive data. The receiver must answer either with an acknowledgement (ACK) frame if it is ready to receive or with a negative acknowledgement (NAK) frame if it is not. By requiring a response even if the answer is negative, the initiator knows that its enquiry was in fact received even if the receiver is currently unable to accept a transmission. If neither an ACK nor a NAK is received within a specified time limit, the initiator assumes that the ENQ frame was lost in transit, disconnects, and sends a replacement. An initiating system ordinarily makes three such attempts to establish a link before giving up.



If the response to the ENQ is negative for the three attempts, the initiator disconnects and begins the process again at another time. If the response is positive, the initiator is free to send its data. Once all of its data have been transmitted, the sending system finishes with an **end of transmission (EOT)** frame.

### Poll/Select

The **poll/select** method of line discipline works with topologies where one device is designated as a **primary station** and the other devices are **secondary stations**

### WORKING:

Whenever a multipoint link consists of a primary device and must be secondary devices using a single transmission line, all exchanges must be such through the primary device even when the ultimate destination is a secondary device. The primary device controls the link; the secondary devices follow its instructions. It is up to the primary to determine which devices are allowed to use the channel at a given time. The primary, therefore, always is the initiator of a session. If the primary wants to receive data, it asks the secondary's if they have anything to send; this function is called polling. If the primary device to send data, it tells the target secondary to get ready to receive.

### Addresses

For point-to-point configuration, there is no need for addressing; any transmission put onto the link by one device can be intended only for the other. For the primary device in a multipoint topology to be able to identify and communicate with a specific secondary device, however, there must be an addressing convention. For this reason, every device on a link has an address that can be used for identification.

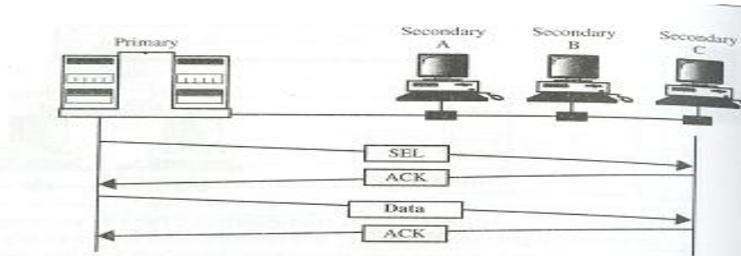
Poll/select protocol identifies each frame as being either to or from a specific device on the link. Each secondary device has an address that differentiates it from the others. In any transmission, that address will appear in a specified portion of each frame called an address field or header depending on the protocol. If the

transmission comes from the primary device, the address indicates the recipient of the data. If the transmission comes from a secondary device, the address indicates the originator of the data.

**Select:**

The select mode is used whenever the primary device has something to send. Remember that the primary controls the link. If the primary is not either sending or receiving data, it knows the link is available. If it has something to send, it sends it. What it does not know, however, is whether the target device is prepared to receive (usually, prepared to receive means on). So the primary must alert the secondary to the upcoming transmission and wait for an acknowledgment of the secondary's ready status. Before sending data, the primary creates and transmits select.

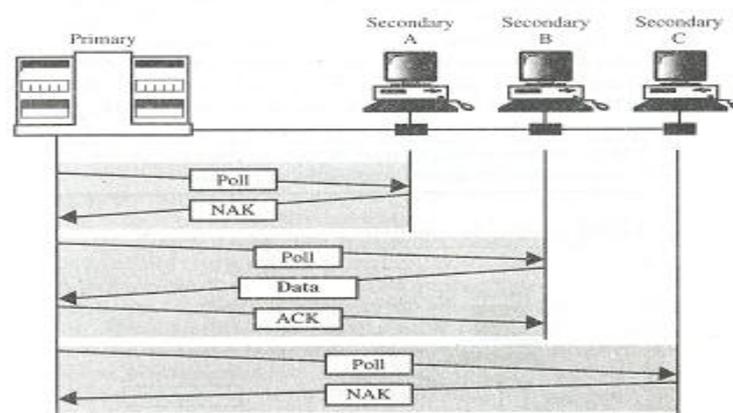
If the secondary is awake and running, it returns an ACK frame to the primary. The primary then sends one or more data frames, each addressed to the intended secondary.



**Poll :** The polling function is used by the primary device to solicit transmissions from the secondary devices. the secondary's are not allowed to transmit data unless asked. By keeping all control with the primary, the multipoint system guarantees that only one transmission can occur at a time thereby ensuring against signal collisions without requiring elaborate precedence protocols. When the primary is ready to receive data, it must ask (poll) each device in turn if it has anything to send. When the first secondary is approached, it responds either with a NAK frame if it has nothing to send or with data (in the form of a data frame) it does.

If the response is negative (a NAK frame), the primary then polls the next secondary in the same way until it finds one with data to send. When the response is positive a data frame, the primary reads the frame and returns an acknowledgment (ACK frame) verifying its receipt. The secondary may send several data frames one after the other, or it may be required to wait for an ACK before sending each one, depending on the protocol being used.

There are two possibilities for terminating the exchange: either the secondary sends all its data, finishing with an end of transmission (EOT) frame, or the primary says, "Time's up." Which of these occurs



depends on the protocol and the length of the message. Once a secondary has finished transmitting, the primary can poll the remaining devices.

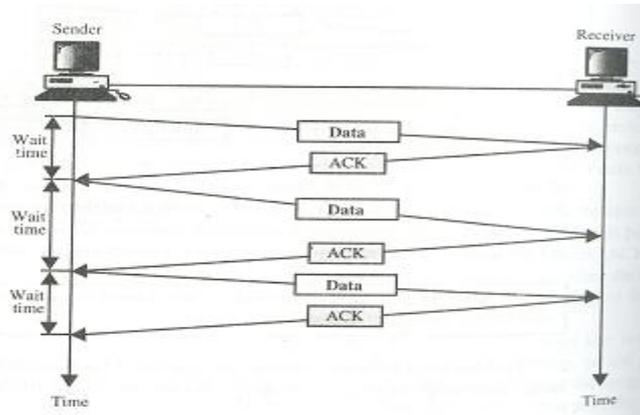
**IV. Flow Control**

The second aspect of data link control is flow control. In most protocols, flow control is a set of procedures that tells the sender how much data it can transmit before it must wait for an acknowledgment from the receiver. The flow of data must not be allowed to overwhelm the receiver. Any receiving devices has a limited speed at which it can process incoming data and a limited amount of memory in which to store incoming data. The receiving device must be able to inform the sending device before those limits are reached and to request that the transmitting device send fewer frames or stop temporarily. Incoming data must be checked and processes before they can be used.

Two methods have been developed to control flow of data across communications links: stop-and-wait and sliding window.

**Stop-and-wait:**

In a stop-and-wait method of flow control, the sender waits for an acknowledgement after every frame it sends. Only when an acknowledgement has been received is the next frame sent. This process of alternately sending and waiting repeats until the sender transmits an end of transmission (EOT) frame.



The advantage of stop-and-wait is simplicity: each frame is checked and acknowledged before the next frame is sent. The disadvantage is inefficiency: stop-and-wait slow. Each frame must travel all the way to the receiver and an acknowledgment must travel all the way back before the next frame can be sent.

**Sliding window:**

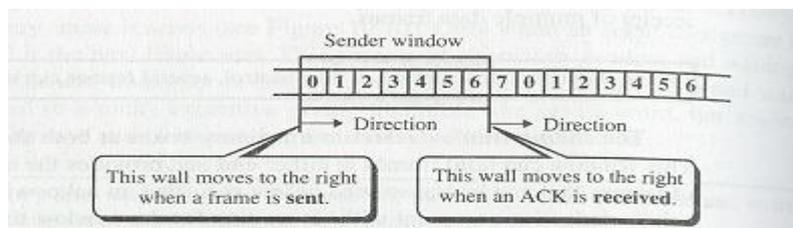
In the sliding window method of flow control, the sender can transmit several frames before needing an acknowledgment. Frames can be sent one right after another, meaning that the link can carry several frames at once and its capacity can be used efficiently. The receiver acknowledges only some of the frames, using a single ACK to confirm the receipt of multiple data frames.

When the receiver sends an ACK, it includes the number of the next frame it expects to receive. In other words, to acknowledge the receipt of a string of frames ending in frame 4, the receiver sends an ACK containing the number 5. When the sender sees an ACK with number 5, it knows that all frames up through number 4 have been received.

The window can hold  $n - 1$  frames at either end; therefore, a maximum of  $n - 1$  frames may be sent before an acknowledgment is required.

**Sender window:**

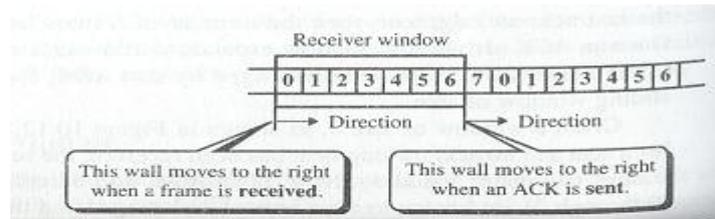
At the beginning of a transmission, the sender's window contains  $n - 1$  frame. As frames are sent out, the left boundary of the window moves inward, shrinking the size of the window. Given a window of size  $w$ , if three frames have been transmitted since the last acknowledgement, then the number of frames left in the window is  $w - 3$ . Once an ACK arrives, the window expands to allow in a number of new frames equal to the number of frames acknowledged by that ACK.



Given a window of size 7., if frames 0 through 4 have been sent and no acknowledgment has been received, the sender's window contains two frames (numbers 5 and 6). Now, if an ACK numbered 4 is received, four frames (0 through 3) are known to have arrived undamaged and the sender's window expands to include the next four frames in its buffer. At this point, the sender's window contains six frames (numbers 5, 6, 7, 0, 1, 2). If the received ACK had been numbered 2, the sender's window would have expanded by only two frames, to contain a total of four.

**Receiver Window:**

At the beginning of transmission, the receiver window contains not  $n - 1$  frames but  $n - 1$  spaces for frames. As new frames come in, the size of the receiver window shrinks. The receiver window therefore represents not the number of frames received but the number of frames that may still be received before an ACK must be sent. Given a window of size  $w$ , if three frames are received without an acknowledgment being returned, the number of spaces in the window is  $w - 3$ . As soon as an acknowledgment is sent, the window expands to include places for a number of frames equal to the number of frames acknowledged. In the figure, the window contains spaces for seven frames, meaning that seven frames may be received before an ACK must be sent. With the arrival of the first frame, the receiving window shrinks, moving the boundary from space 0 to 1. The window has shrunk by one, so the receiver may now accept six frames before it is required to send an ACK. If frames 0 through 3 have arrived but have not been acknowledged, the window with contains three frame spaces.



As each ACK  $a$  is sent out, the receiving window expands to include as many new placeholders as newly acknowledged frames. The window expands to include a number of new frame spaces equal to the number of the most recently acknowledged frame minus the number of the previously acknowledged frame. In a seven-frame window, if the prior ACK was for frame 2 and the current ACK is for frame 5, the window expands by three ( $5 - 2$ ). If the prior ACK was for frame 3 and the current ACK is for frame 1, the window expands by six ( $1 + 8 - 3$ )

**V. Error Control**

In the data link layer, the term error control refers primarily to methods of error detection and retransmission.

**Automatic Repeat Request (ARQ):**

Error correction in the data link layer is implemented simply: anytime an error detected in an exchange, a negative acknowledgement (NAK) is returned and the specified frames are retransmitted. This process is called automatic repeat request (ARQ).

ARQ error control is implemented in the data link layer as an adjunct to flow control. In fact, stop-and-wait flow control is usually implemented as stop-and-wait ARQ and sliding window is usually implemented as one of two variants of sliding window ARQ, called go-back- $n$  or selective-reject.

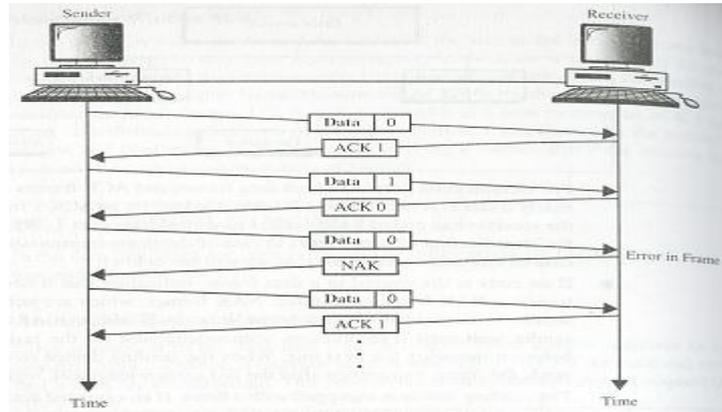
**Stop-and-Wait ARQ:**

**Stop-and-wait ARQ** is a form of stop-and-wait flow control extended to include retransmission of data in case of lost or damaged frames. For retransmission to work four features are added to the basic flow control mechanism:

- The sending device keeps a copy of the last frame transmitted until it receives an acknowledgment for that frame. Keeping a copy allows the sender to retransmit lost or damaged frames until they are received correctly.
- A data 0 frame is acknowledged by an ACK 1 frame, indicating that the receiver has gotten data 0 and is now expecting data 1. This numbering allows of identification of data frames in case of duplicate transmission
- If an error is discovered in a data frame a NAK frame is returned. NAK frames, which are not numbered, tell the sender to retransmit the last frame sent. Stop-and-wait ARQ requires that the sender wait until it receives an acknowledgment for the last frame transmitted before it transmits the next one.
- The sending device is equipped with a timer. If an expected acknowledgment is not received within an allotted time period, the sender assumes that the last data frame was lost in transit and sends it again.

**Damaged Frames** When a frame is discovered by the receiver to contain an error, it returns a NAK frame and the sender retransmits the last frame. For example, the sender transmits a data frame: data 0. The receiver returns an ACK 1, indicating that data 0 arrived undamaged and it is now expecting data 1. The sender transmits its next frame: data 1. It arrives undamaged, and the receiver returns ACK 0. The sender transmits its next

frame: data 0. The receiver discovers an error in data 0 and returns a NAK. The sender retransmits data 0. This time data 0 arrives intact, and the receiver returns ACK 1.

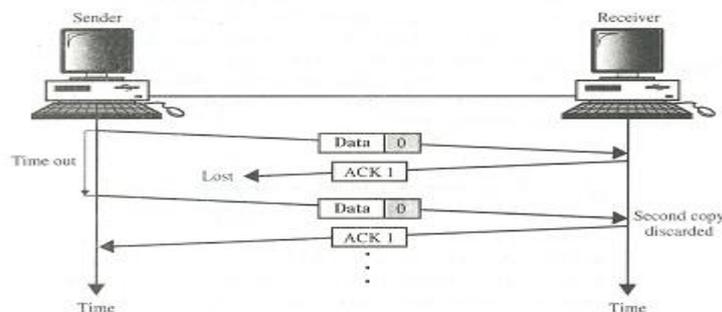


**Lost Frame** Any of the three frame types can be lost in transit.

**Lost Data Frame** the sender is equipped with a timer that starts every time a data frame is transmitted. If the frame never makes it to the receiver, the receiver can never acknowledge it, positively or negatively. The sending device waits for an ACK or NAK frame until its timer goes off, at which point it tries again. It retransmits the last data frame, restarts its timer, and waits for an acknowledgment.

**Lost Acknowledgment** In this case, the data frame has made it to the receiver and has been found to be either acceptable or not acceptable. But the ACK or NAK frame returned by the receiver is lost in transit. The sending device waits until its timer goes off, and then retransmits the data frame. The receiver checks the number of the new data frame. If the lost frame was a NAK, the receiver accepts the new copy and returns the appropriate ACK.

**Sliding Window ARQ:**



Among the several popular mechanisms for continuous transmission error control, two protocols are the most popular: go-back-n ARQ and selective-reject ARQ, both based on sliding window flow control. To extend sliding window to cover retransmission of lost or damaged frames, three features are added to the basic flow control mechanism:

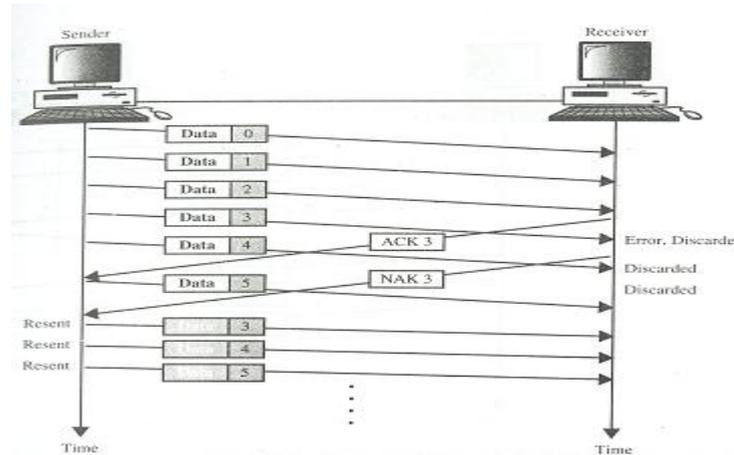
- The sending device keeps copies of all transmitted frames until they have been acknowledged. If frames 0 through 6 have been transmitted and the last acknowledgment was for frame 2 the sender keeps copies of frames 3 through 6 until it knows that they have been received undamaged.
- In addition to ACK frames the receiver has the option of returning a NAK frame if the data have been received damaged. The NAK frame tells the sender to retransmit a damaged frame. Because sliding window is a continuous transmission mechanism both ACK and NAK frames must be numbered for identification. Note that data frames that are received without errors do not have to be acknowledged individually
- . Like stop-and-wait ARQ, the sending device in sliding window ARQ is equipped with a timer to enable it to handle lost acknowledgment. In sliding window ARQ,  $n - 1$  frame (the size of the window) may be sent before an acknowledgment must be received. By retransmitting the data frames, two possibilities are covered:; lost data and lost NAK. If the lost frame was an ACK frame, the receiver can recognize the redundancy by the number on the frame and discard the redundant data.

**Go-Back-n ARQ:**

In this sliding window go-back-n method. If one frame is lost or damaged, all frames sent since the last frame acknowledged are retransmitted.

**Damaged Frame**

What if frames 0, 1, 2, and 3 have been transmitted, but the first acknowledgment received is a NAK 3? Remember that a NAK means two things: (1) a positive acknowledgment of all frames received prior to the damaged frame and (2) a negative acknowledgment of the frame indicated. If the first acknowledgment is a NAK 3, it means that data frames 0, 1, and 2, were all received in good shape. Only frame 3 must be resent.

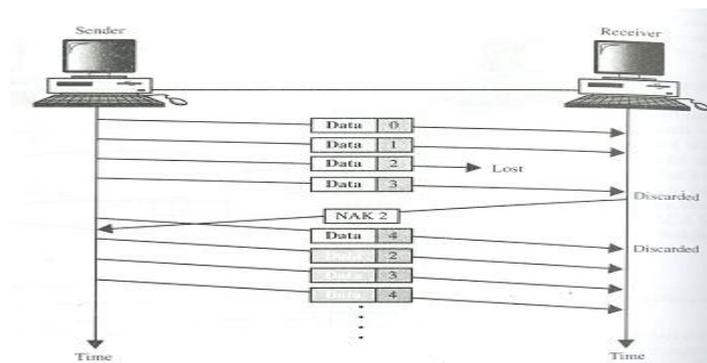


What if frames 0 through 4 have been transmitted before a NAK is received for frame 2? As soon as the receiver discovers an error, it stops accepting subsequent frames until the damaged frame has been replaced correctly.

**Lost Data frame:**

Sliding window protocols require that data frames be transmitted sequentially. If one or more frames are so noisy corrupted that they become lost in transit, the next frame to arrive at the receiver will be out of sequence. The receiver checks the identifying number on each frame, discovers that one or more have been skipped, and returns a NAK for the first missing number frame. A NAK frame does not indicate whether the frame has been lost or damaged, just that it needs to be resent. The sending device then retransmits the frame indicated by the NAK as well as any frames that it had transmitted after the lost one.

Data 0 and data 1 arrive intact but data 2 is lost. The next frame to arrive at the receiver is data 3. The receiver is



expecting data 2 and so considers data 3 to be an error, discards it and returns a NAK 2, indicating the 0 and 1 have been accepted but 2 is in error (in this case lost). **Lost Acknowledgement:**

The sender is not expecting to receive an ACK frame for every data frame it sends. It cannot use the absence of sequential ACK numbers to identify lost ACK or NAK frames. Instead, it uses a timer. The sending device can send as many frames as the window allows before waiting for an acknowledgment. Once that limit has been reached or the sender has no more frames to send, it must wait.

If the ACK (or, especially, if the NAK) sent by the receiver has been lost, the sender could wait forever. To avoid tying up both device, the sender is equipped with a timer that begins counting whenever the window capacity is reached. If an acknowledgment has not been received within time limit, the sender retransmits every frame transmitted.

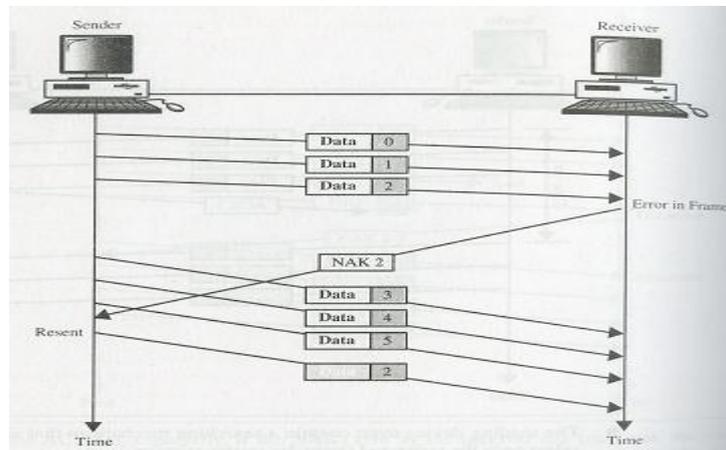
**Selective-Reject ARQ:**

In **selective-reject ARQ**, only the specific damaged or lost frame is retransmitted. If a frame is corrupted in transit, a NAK is returned and the frame is resent out of sequence. The receiving device must be able to sort the frames it has and insert the retransmitted frame into its proper place in the sequence. To make such selectivity possible, a selective reject ARQ system differs from a go-back-n ARQ system in the following ways

- The receiving device must contain sorting logic to enable it to reorder frames received out of sequence..
- The sending device must contain a searching mechanism that allows it to find and select only the requested frame for retransmission.
- A buffer in the receiver must keep all previously received frames on hold until all retransmissions have been sorted and any duplicate frames have been identified and discarded. To aid selectivity, ACK numbers, like NAK numbers, must refer to the frame received (or lost) instead of the next frame expected. This complexity requires a smaller window size than is needed by the go-back-n method if it is to work efficiently.

**Damaged Frames:**

A situation in which a damaged frame is received. As you can see, frames 0 and 1 are received but not acknowledged. Data 2 arrives and is found to contain an error. A NAK is returned. Like NAK frames in go-back-n error correction, a NAK here both acknowledges the intact receipt of any previously unacknowledged data frames and indicates an error in the current frame.

**Lost Frames:**

Although frames can be accepted out of sequence, they cannot be acknowledged out of sequence. If a frame is lost, the next frame will arrive out of sequence. When the receiver tries to reorder the existing frames to include it, it discovers the discrepancy and returns a NAK.

**Lost Acknowledgement:**

Lost ACK and NAK frames are treated by selective ARQ just as they are by go-back-n ARQ. When the sending device reaches either capacity of its window or the end of its transmission, it sets a timer. If no acknowledgment arrives in the time allotted, the sender retransmits all of the frames that remain unacknowledged. In most cases, the receiver will recognize any duplication and discard them.

**Comparison between Go-Back-n and Selective-Reject:**

Although retransmitting only specific damaged or lost frames may seem more efficient than resending undamaged frames as well, it is in fact less so. Because of the complexity of the sorting and storage required by the receiver, and the extra logic needed by the sender to select specific frames for retransmission, selective-reject ARQ is expensive and not often used. In other words, selective-reject gives better performance but in practice it is usually discarded in favor of go-back-n for simplicity of implementation.

**VI. Conclusion:**

From all the techniques of DLC in Data link control we can conclude frames received from the physical layer are corrected and make sure that frames are error-free flow control and not corrupted during transferring of data from physical layer to network layer.

**References:**

- [1] A.Harish Babu, P.Ashesh Babu, E.Vamsi Krishna IETE Members and Research students in kl-university.
- [2] C. Balanis, Antenna Theory, Analysis and Design, 3rd edition, New York: Wiley, 2005. 2.
- [3] 4S Telecom (P) LTD. BANGALORE, User Manual

**Biography:**



**Alluri Sri Amith Varma** was born in 1992 in, West Godavari District. He is pursuing his B.tech from K L University. He is interested in Communications and Wireless Networks.



**Changanti Veera Sankara Sarma** is born in Rajahmundry, East Godavari District, Andhra Pradesh ,India on 4<sup>th</sup> June1992 and currently pursuing B.TECH 4<sup>th</sup>year in Electronics and Communication Engineering in K.L.University with specialization in Communications. Areas of interests are Linear integrated circuit applications , Digital Logic Design and Antennas.



**P. Sai Madan Kumar** was born in 1992 at prakasam district. He is currently pursuing B.Tech from K.L.University. He is interested in Wireless systems and Telecommunication.



**Thumati Ravi** is working as Associate Proffesor in KL University. He is interested in Image Processing.