Business Data Communications & Networking:

12th Edition

Chapter 4

Sadiq M. Sait, PhD
Professor
College of Computer Sciences and Engineering
Director, Industry Collaboration

October 2023

DATA LINK LAYER

- The data link layer (also called layer 2) is responsible for moving a message from one computer or network device to the next computer or network device in the overall path from sender or receiver.
- It controls the way messages are sent on the physical media.
- Both the sender and receiver have to agree on the rules, or protocols, that govern how they will communicate with each other.
- A data link protocol determines:
  - '*who can transmit at what time (media access)*'
  - '*where a message begins and ends*, and
  - '*how a receiver recognizes and corrects a transmission error*'

OBJECTIVES

- Understand the role of the data link layer
- Become familiar with ***two*** basic approaches to controlling access to the media
- Become familiar with common sources of error and their prevention
- Understand three common error detection and correction methods
- Become familiar with several commonly used data link protocols

4.1 INTRODUCTION

- The data link layer is responsible for sending and receiving messages to and from other computers.
- Its job is to ***reliably*** move a message from one computer over one circuit to the next computer where the message needs to go.
- The data link layer performs two main functions and therefore is often divided into two sublayers.
  - The first sublayer (called the **logical link control** [LLC] sublayer) is the data link layer's connection to the network layer above it.
    - At the sending computer, the LLC sublayer software is responsible for communicating with the network layer software (e.g., IP) and for taking the network layer Protocol Data Unit (PDU)—usually an IP packet—and surrounding it with a data link layer PDU—often an Ethernet frame.
    - At the receiving computer, the LLC sublayer software removes the data link layer PDU and passes the message it contains (usually an IP packet) to the network layer software

4.1 INTRODUCTION (contd).

- The second sublayer (called the media access control [MAC] sublayer) controls the physical hardware.
- The MAC sublayer software at the sending computer controls ***how and when the physical layer converts bits into the physical symbols*** that are sent down the circuit.
  - At the ~~receiving~~ ***sending*** computer, the MAC sublayer software takes the data link layer PDU from the LLC sublayer, converts it into a stream of bits, and controls when the physical layer actually transmits the bits over the circuit.
  - At the receiving computer, the MAC sublayer receives a stream of bits from the physical layer and translates it into a coherent PDU, ensures that no errors have occurred in transmission, and passes the data link layer PDU to the LLC sublayer.
- Both the sender and receiver have to agree on the rules or protocols that govern how their data link layers will communicate with each other. A data link protocol performs three functions:
  - ***Controls when computers transmit (media access control)***
  - ***Detects and corrects transmission errors (error control)***
  - ***Identifies the start and end of a message by using a PDU (message delineation).***

4.2 MEDIA ACCESS CONTROL

- Media access control refers to the need to control ***when*** computers transmit.
- With point-to-point full-duplex configurations, MAC is unnecessary
- MAC is important when several computers share the same communication circuit
  - such as a point-to-point configuration with a half-duplex configuration that requires computers to take turns or
  - a multipoint configuration in which several computers share the same circuit.
- It is critical to ensure that no two computers attempt to transmit data at the same time—but if they do, there must be a way to recover from the problem.
- There are two fundamental approaches to media access control:
  - contention and
  - controlled access

4.2.1 Contention

- With contention, computers wait until the circuit is free (i.e., no other computers are transmitting) and then transmit whenever they have data to send.
- Contention is commonly used in Ethernet LANs.
  - As an analogy, suppose that you are talking with some friends. People listen, and if no one is talking, they can talk. If you want to say something, you wait until the speaker is done and then you try to talk.
  - Usually, people yield to the first person who jumps in at the precise moment the previous speaker stops.
- Sometimes two people attempt to talk at the same time, so there must be some technique to continue the conversation after such a verbal collision occurs.

4.2.2 Controlled Access

- With controlled access controls the **circuit** and determines which clients can transmit at what time.
- There are two commonly used controlled access techniques:
  - access requests
  - and polling.

4.2.2 Controlled Access (contd)

- With the **access request technique**, client computers that want to transmit send a request to transmit to the device that is controlling the circuit (e.g., the wireless access point).
- The **controlling device grants permission** for one computer at a time to transmit.
- When one computer has permission to transmit, all other computers wait until that computer has finished, and then, if they have something to transmit, they use a **contention technique** to send an access request.
- The access request technique is like a classroom situation in which the instructor calls on the students who raise their hands.

4.2.2 Controlled Access (contd)

- ***Polling*** is the process of sending a signal to a client computer that gives it permission to transmit.
- With polling, the clients store all messages that need to be transmitted.
- ***Periodically***, the controlling device (e.g., a wireless access point) polls the client to see if it has data to send.
  - If the client has data to send, it does so.
  - If the client has no data to send, it responds negatively, and the controller asks another client if it has data to send.
- Types of Polling
  - Roll-call polling, the controller works consecutively through a list of clients, first polling client 1, then client 2, and so on, until all are polled.
  - Roll-call polling can be modified to select clients in ***priority*** so that some get polled more often than others. For example, one could increase the priority of client 1 by using a polling sequence such as 1, 2, 3, 1, 4, 5, 1, 6, 7, 1, 8, 9.

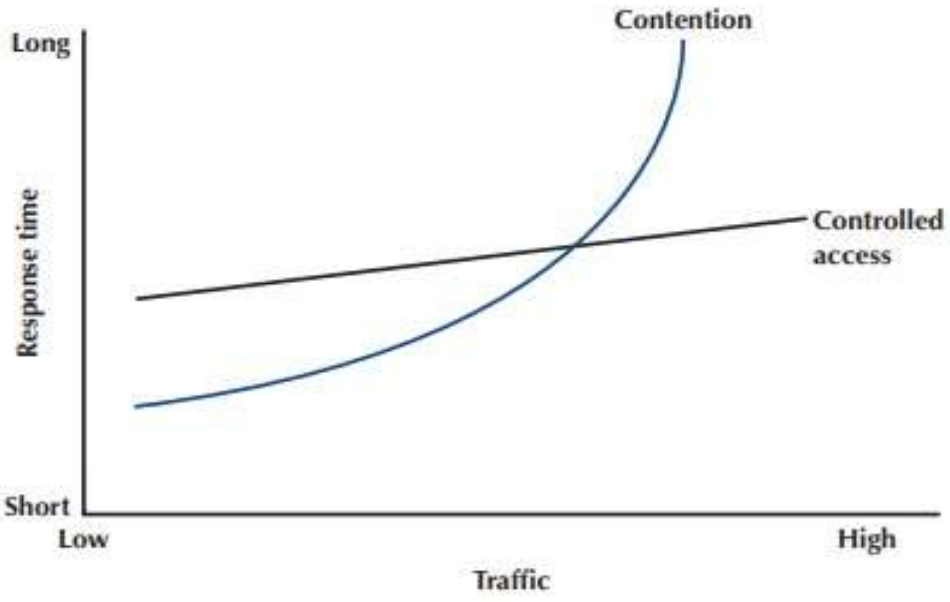
4.2.2 Controlled Access (contd)

- Typically, roll-call polling involves some ***waiting*** because the controller has to poll a client and then wait for a response.
  - The response might be
    - an incoming message that was waiting to be sent,
    - a negative response indicating nothing is to be sent, or
    - the full "time-out period" may expire because the client is temporarily out of service (e.g., it is malfunctioning or the user has turned it off).
  - Usually, a ***timer*** "times out" the client after waiting several seconds without getting a response.
  - If some sort of fail-safe time-out is **not used**, the circuit poll might lock up indefinitely on an out-of-service client.
- With ***hub polling*** (often called ***token passing***), one device starts the poll and passes it to the next computer on the multipoint circuit, which sends its message and passes the poll to the next. That computer then passes the poll to the next, and so on, until it reaches the first computer, which restarts the process again.

4.2.3 Relative Performance

- Which is better? Controlled Access or Contention?
  - ***Whichever*** will permit the most amount of data to be transmitted!
  - In general:
    - Contention is better for small networks with low usage.
    - Controlled Access is better for large networks with high usage.
    - The crossover point is around 20 computers.

FIGURE 4-1
Relative response times



4.3 ERROR CONTROL

- Types of Errors
  - Human Errors (e.g., Typing, controlled by Application Programs)
  - Network Errors (during transmission, controlled by Network HW/SW)
    - ***Categories*** of NW Errors:
      - Corrupted Data
      - Lost Data
  - We will look into how to:
    - Prevent
    - Detect
    - Correct
  - Source:
    - Noise on the line
    - Rate could be something like 1-bit error in 500K transmitted (this is also known as ***error rate***)
- Burst Errors, not uniformly distributed (more than one data bit changed).
- 1 in 500K also means 100 bits in 50,000K
- Sometimes, 2-bit errors in a character may become another valid character

4.3.1 Sources of Errors

- Line noise and distortion can cause data communication errors.
  - May add additional bits or flip bits.
- **Types:**
  1. **White or Gaussian Noise:** (Due to thermal agitation of electrons), S/N can be increased to address this.
  2. **Impulse Noise (or Spikes):** Heard as a crackling noise or a click (1/100 of a Second). On a 1.5 Mbps line, 15,000 bits could change.
     - Source: Voltage Fluctuations in adjacent lines, lightning, poor connections, fluorescent lights, etc.
  3. **Cross-Talk:** One circuit picks up a signal from another. Telephone lines. (Increases with increased distance, proximity between lines, increased signal strength, and higher frequency signals.) Generally, the corrupting signal has low strength.
  4. **Echoes:** Poor connections reflect the signal back to the transmitter.
  5. **Attenuation:** Loss of signal as it travels.
     - Some power is absorbed by the medium. High-frequency signals lose more power.
     - Noise occurs due to unequal loss of component frequencies.
  6. **Intermodulation Noise:** Like cross talk. Signals from 2 circuits combine to form a new signal that falls into a band reserved for another signal.

4.3.2 Error Prevention

- There are many techniques to prevent errors (or at least reduce them), depending on the situation.
  - **Shielding** (protecting wires by covering them with an insulating coating) is one of the best ways to prevent **impulse noise**, **cross-talk**, and **intermodulation noise**.
    - **Cost?** The greater the shielding, the more expensive the cable and the more difficult it is to install.
  - Moving cables away from sources of noise (especially power sources) can also reduce impulse noise, cross-talk, and intermodulation noise.
    - **For impulse noise, this means avoiding lights and heavy machinery.**
    - Locating communication cables away from power cables is always a good idea.
  - For cross-talk, this means **physically separating the cables from other communication cables**.
    - Cross-talk and intermodulation noise are often caused by **improper multiplexing.**
      - Changing multiplexing techniques (e.g., from FDM to TDM or changing the frequencies, or size of the guardbands in FDM can help.

4.3.2 Error Prevention (contd)

- Many types of noise (e.g., echoes, white noise) can be caused by poorly maintained equipment or poor connections and splices among cables.
- This is particularly true for echo in fiber-optic cables, which is almost always caused by poor connections.
  - Solution? Tune the transmission equipment and redo the connections.
- To avoid **attenuation**, telephone circuits have **repeaters** or **amplifiers.**
- An amplifier (for **analog** signals, voice on telephone circuit) takes the incoming signal, increases its strength, and retransmits it to the next section of the circuit.
  - The distance between amplifiers depends on the amount of attenuation, although 1- to 10-mile intervals are common.
  - On analog circuits, it is important to recognize that the noise and distortion are also amplified.
- **Repeaters** are used on **digital** circuits.
  - A repeater receives the incoming signal, translates it into a digital message, and retransmits the message.
  - Because the message is recreated at each repeater, noise and distortion from the previous circuit are not amplified.
  - This provides a much cleaner signal and results in a lower error rate for digital circuits

4.3.3 Error Detection

- The **only** way to do error detection is to send **extra** data with each message.
- These extra bits are added to each message by the data link layer of the sender on the basis of some mathematical calculations performed on the message.
- The receiver performs the same mathematical calculations on the message it receives and matches its results against the error-detection data that was transmitted with the message.
- If the two match, the message is assumed to be correct.
- In general, the larger the amount of error-detection data sent, the greater the ability to detect an error.
- However, as the amount of error-detection data is increased, the **throughput** of useful data is reduced, because more of the available capacity is used to transmit these error-detection data and less is used to transmit the actual message itself.
- Therefore, the efficiency of data throughput varies inversely as the desired amount of error detection is increased.
- Three well-known error-detection methods are: (i) parity checking, (ii) checksum, and, (iii) cyclic redundancy checking.

4.3.3 Parity Checking

- Oldest and simplest error-detection methods.
- One additional bit is added to each byte in the message.
- The value of this additional parity bit is based on the number of 1s in each byte transmitted.
- This parity bit is set to make the total number of 1s in the byte (*including the parity bit*) either an even number or an odd number.
- Any single error will be detected by parity, but it cannot determine which bit was in error.
- If two bits are switched, the parity check will not detect any error. It is easy to see that parity can detect errors only when an odd number of bits have been switched; any even number of errors cancel one another out.
- Therefore, the probability of detecting an error, given that one has occurred, is only about 50%.
- Many networks today do not use parity because of its low error-detection rate.

4.3.3 Checksum

- With the checksum technique, a checksum (typically 1 byte) is added to the end of the message.
- The checksum is calculated by adding the decimal value of each character in the message, dividing the sum by 255, and using the remainder as the checksum.
- The receiver calculates its own checksum in the same way and compares it with the transmitted checksum.
- If the two values are equal, the message is presumed to contain no errors.
- Use of checksum detects close to 95% of the errors for multiple-bit burst errors.

4.3.3 Cyclic Redundancy Check

- Most popular error-checking schemes, Adds 8, 16, 24, or 32 bits to msg.
- A message is treated as one long binary number, P.
- Before transmission, the data link layer (or hardware device) divides P by a fixed binary number, G, resulting in a whole number, Q, and a remainder, R.
- So, P⁄G = Q + R⁄G. For example, if P=58 and G=8, then Q=7 and R=2.
- G is chosen so that the remainder, R, will be either 8 bits, 16 bits, 24 bits, or 32 bits.
- The remainder, R, is appended to the message as the error-checking characters before transmission.
- The receiving hardware divides the received message by the same G, which generates an R. The receiving hardware checks to ascertain whether the received R agrees with the locally generated R.
- If it does not, the message is assumed to be in error.

4.3.3 Cyclic Redundancy Check

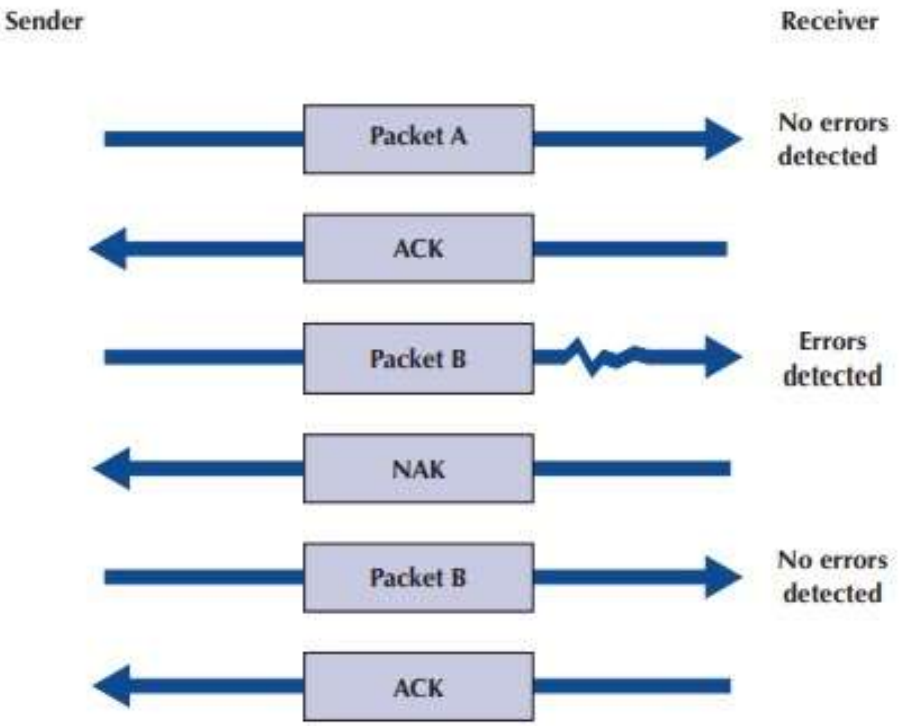
- Cyclic redundancy check performs quite well.
- The most commonly used CRC codes are CRC-16 (a 16-bit version), CRC-CCITT (another 16-bit version), and CRC-32 (a 32-bit version).
- The probability of detecting an error is 100% for all errors of the same length as the CRC or less.
- For example, CRC-16 is guaranteed to detect errors if 16 or fewer bits are affected.
- If the burst error is longer than the CRC, then CRC is not perfect but is close to it.
  - CRC-16 will detect about 99.998% of all burst errors longer than 16 bits
  - CRC-32 will detect about 99.99999998% of all burst errors longer than 32 bits

4.3.4 Error Correction via Retransmission

- Once an error has been detected, it must be corrected. The simplest, most effective, least expensive, and most commonly used method for error correction is **retransmission.**
- With retransmission, a receiver that detects an error simply asks the sender to retransmit the message until it is received without error.
- This is often called Automatic Repeat reQuest (**ARQ**). There are two types of ARQ: stop-and-wait ARQ and continuous ARQ.
  - With stop-and-wait ARQ, the sender stops and waits for a response from the receiver after each data packet. After receiving a packet, the receiver sends either an ACK or a NAK.
  - If it is a NAK, the sender resends the previous message. If it is an ACK, the sender continues with the next message.
  - Stop-and-wait ARQ is by definition a half-duplex transmission technique (Figure 4-4).
  - With stop-and-wait ARQ, the receiver does not send an ACK until it is ready to receive more packets.

4.3.4 Error Correction via Retransmission (contd)

FIGURE 4-4
Stop-and-wait ARQ (Automatic Repeat reQuest).
ACK = acknowledgment; NAK = negative acknowledgment



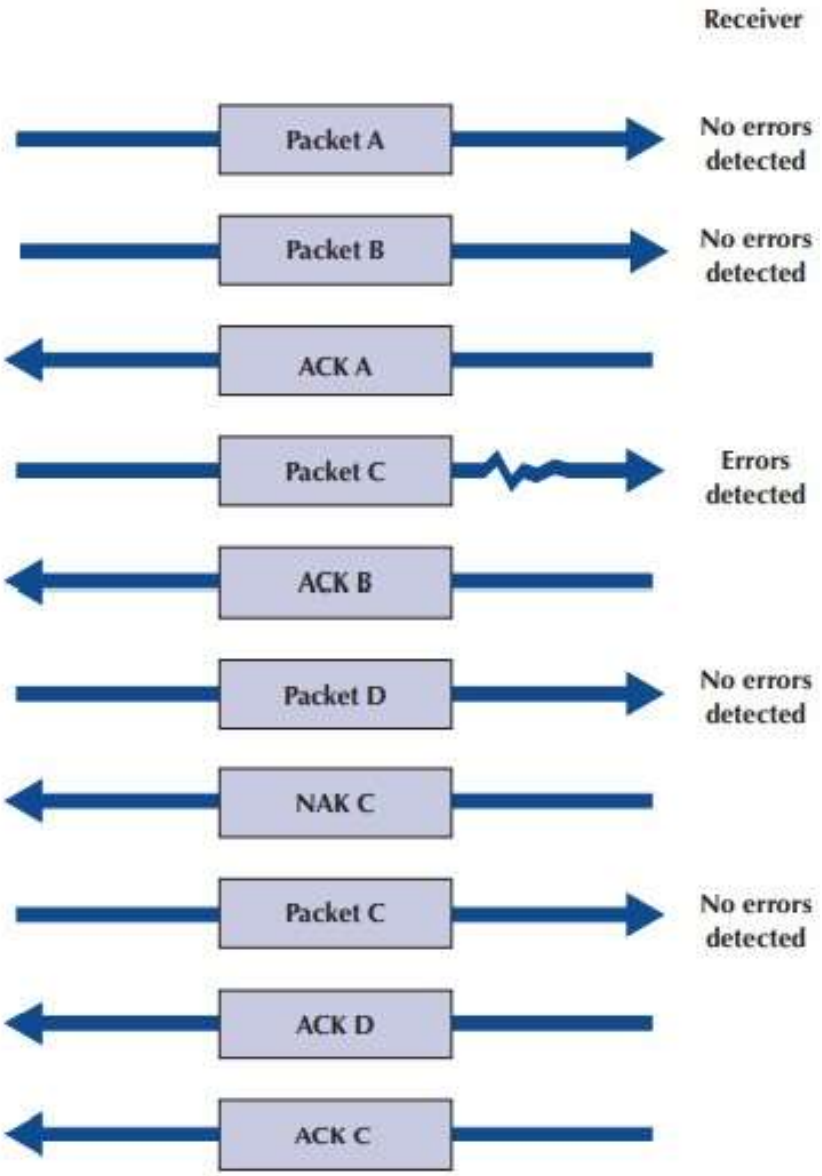
4.3.4 Error Correction via Retransmission (contd)

- With continuous ARQ, the sender does not wait for an acknowledgment after sending a message; it immediately sends the next one.
- Although the messages are being transmitted, the sender examines the stream of returning acknowledgments.
- If it receives an NAK, the sender retransmits the needed messages.
  - The packets that are retransmitted
    - may be only those containing an error (called Link Access Protocol for Modems [LAP-M]) or
    - may be the first packet with an error and all those that followed it (called Go-Back-N ARQ).
- LAP-M is better because it is more efficient.

4.3.4 Error Correction via Retransmission (contd)

- Continuous ARQ is by definition a full-duplex technique,
- Continuous ARQ is sometimes called ***sliding window*** because of the visual imagery the early network designers used to think about continuous ARQ.
- In continuous ARQ, the sender and receiver usually agree on the ***size*** of the sliding window.
- Once the sender has transmitted the maximum number of packets permitted in the sliding window, it cannot send any more packets until the receiver sends an ACK.
- Continuous ARQ is also important in providing ***flow control***, which means ensuring that the computer sending the message is not transmitting too quickly to the receiver.
- For example, if a client computer was sending information too quickly for a server computer to store a file being uploaded, the server might run out of memory to store the file.
- By using ACKs and NAKs, the receiver can control the ***rate*** at which it receives information.

FIGURE 4-5
Continuous ARQ (Automatic Repeat reQuest).
ACK = acknowledgment;
NAK = negative acknowledgment



TF 4-1: How Does Forward Correction Work?

- **Hamming Code**
- Associates even parity bits with a unique combination of data bits
- As an example, with 4-bit code (1010 for example), 3 bits are used as parity.
- Let the data bits be D_3, D_5, D_6, D_7
- And Parity bits be P_1, P_2, P_4
- As depicted in the upper half of Figure 4-6, parity bit
  - P_1 applies to data bits D_3, D_5, and D_7
  - P_2 applies to data bits D_3, D_6, and D_7
  - P_4 applies to data bits D_5, D_6, and D_7
- For the example, in which D_3, D_5, D_6, and D_7 = 1010
  - P_1 must equal 1 because there is only a single 1 among D_3, D_5, and D_7 and parity must be even
  - P_2 must be 0 because D_3 and D_6 are 1s
  - P_4 is 1 because D6 is the only 1 among D_5, D_6, and D_7
- Now if D_7 is corrupted then all Ps will be in error. And D_7 is corrected by complementing it. No need for retransmission.

4.3.5 Forward Error Correction (FEC)

- FEC uses codes containing sufficient redundancy.
- The redundancy, or extra bits required, varies with different schemes.
  - It ranges from a small percentage of extra bits to 100% redundancy, with the number of error-detecting bits roughly equalling the number of data bits.
  - One of the characteristics of many error-correcting codes is that there must be a minimum number of error-free bits between bursts of errors.
- Forward error correction is commonly used in satellite transmission.
  - A round trip from the earth station to the satellite and back includes a significant **delay**.
  - Error rates can fluctuate depending on the condition of equipment, sunspots, or the weather.
  - Some weather conditions make it impossible to transmit without errors, **making forward error correction essential**.
  - **Compared with satellite equipment costs, the additional cost of FEC is insignificant.**

4.3.6 Error Control in Practice

- In the OSI model (see Chapter 1), error control is defined to be a layer-2 function—it is the responsibility of the data link layer.
- However, in practice, since most network cables—especially LAN cables—are very reliable, and errors are far less common than they were in the 1980s, most data link layer software used in LANs (i.e., Ethernet) is configured to **detect** errors, but **not correct them**.
- Any time a packet with an error is discovered, it is simply discarded.
- Wireless LANs and some WANs, where errors are more likely, still perform both error detection and error correction.
  - The implication of this is that error correction must be performed by software at **higher** layers.
  - This software must be able to detect lost packets (i.e., those that have been discarded) and request the sender to retransmit them.
  - This is commonly done by the **transport layer** using continuous ARQ, as we shall see in the next chapter.

4.4 DATA LINK PROTOCOLS

- We will now talk about commonly used data link layer protocols (summarized in Figure 4-7).
- We focus on message delineation, which indicates where a message starts and stops, and the various parts or ***fields*** within the message.
  - For example, **you must clearly indicate which part of a message or packet of data is the error-control portion**; otherwise, the receiver cannot use it properly to determine if an error has occurred.
  - The data link layer performs this function by adding a PDU to the packet it receives from the network layer.
  - This PDU is called a frame.

Protocol	Size	Error Detection	Retransmission	Media Access
Asynchronous transmission	1	Parity	Continuous ARQ	Full Duplex
Synchronous protocols				
SDLC	*	16-bit CRC	Continuous ARQ	Controlled Access
HDLC	*	16-bit CRC	Continuous ARQ	Controlled Access
Ethernet	*	32-bit CRC	Stop-and-wait ARQ	Contention
PPP	*	16-bit CRC	Continuous ARQ	Full Duplex

*Varies depending on the message length.

ARQ = Automatic Repeat reQuest; CRC = cyclical redundancy check; HDLC = high-level data link control; PPP = Point-to-Point Protocol; SDLC = synchronous data link control.

FIGURE 4-7 Protocol summary

4.4.1 Asynchronous Transmission

- Also known as start–stop transmission because the transmitting computer can transmit a character whenever it is convenient
- The receiving computer will accept that character.
- Typically used on point-to-point full-duplex circuits (i.e., circuits that have only two computers on them), so media access control is not a concern.
- With asynchronous transmission, each character is transmitted independently of all other characters.
- To **separate the characters and synchronize transmission**, a start bit and a stop bit are put on the front and back of each individual character.
- Typically, the start bit is a 0 and the stop bit is a 1.
- There is no fixed distance between characters because the terminal transmits the character as soon as it is typed, which varies with the speed of the typist.

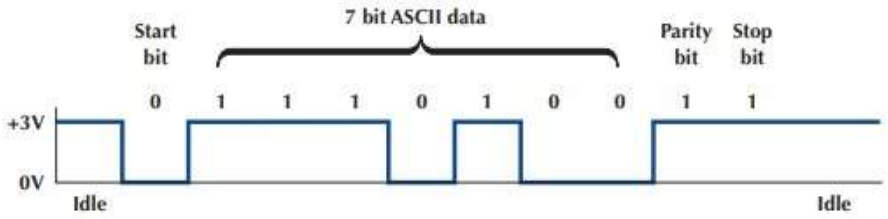


FIGURE 4-8
Asynchronous transmission. ASCII = United States of America Standard Code for Information Interchange

4.4.2 Synchronous Transmission

- With synchronous transmission, all the letters or data in one group of data are transmitted at one time as a block of data. This block of data is called a **FRAME**.
- For example, a terminal or personal computer will save all the keystrokes typed by the user and transmit them only when the user presses a special “transmit” key.
- **In this case, the start and end of the entire frame must be marked, not the start and end of each letter.**
- Synchronous transmission is often used on both point-to-point and multipoint circuits.
- For multipoint circuits, each packet must include a destination address and a source address, and media access control is important.
- The start and end of each frame (synchronization) sometimes is established by adding synchronization characters (SYN) to the start of the frame.

4.4.2 Synchronous Transmission (contd)

- Depending on the protocol, there may be anywhere from one to eight SYN characters.
- After the SYN characters, the transmitting computer sends a long stream of data that may contain thousands of bits.
- In summary, asynchronous data transmission means each character is transmitted as a totally **independent** entity with its own start and stop bits to inform the receiving computer that the character is beginning and ending.
- Synchronous transmission means **whole blocks of data are transmitted as frames** after the sender and the receiver have been synchronized.
- There are many protocols for synchronous transmission. We discuss four commonly used synchronous data link protocols.

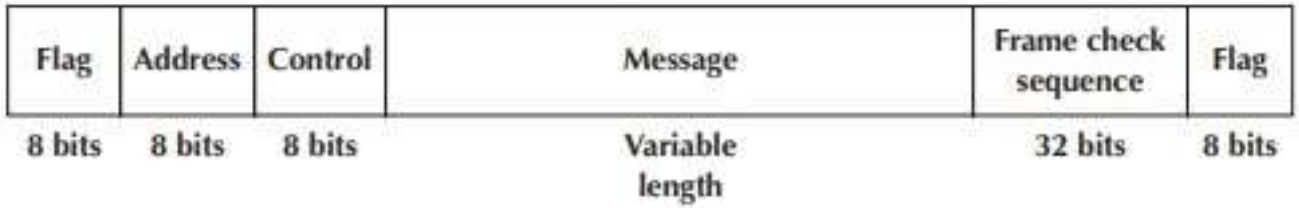
4.4.2 Synchronous Data Link Control (SDLC)

- SDLC is a mainframe protocol developed by IBM in 1972 that is still in use today.
- It uses a controlled-access media access protocol.
- Each SDLC frame **begins and ends** with a special bit pattern (01111110), known as the **flag**.
- The **address** field identifies the destination.
- The length of the address field is usually 8 bits but can be set at 16 bits; all computers on the same network must use the same length.
- The **control** field identifies the kind of frame that is being transmitted, either **information or supervisory**.
  - An information frame is used for the transfer and reception of messages, frame numbering of contiguous frames, and the like.
  - A supervisory frame is used to transmit acknowledgments (ACKs and NAKs).

4.4.2 Synchronous Data Link Control (SDLC)

- The **message** field is of variable length and is the user's message.
- The **frame check sequence** field is a 32-bit CRC code (some older versions use a 16-bit CRC).

FIGURE 4-9
SDLC (synchronous data link control) frame layout



4.4.2 High-Level Data Link Control (HDLC)

- HDLC is a formal standard developed by the ISO used in WANs.
- HDLC is essentially the same as SDLC, except that the address and control fields can be **longer**.
- HDLC also has several additional benefits, such as a larger sliding window for continuous ARQ.
- It uses a controlled-access media access (CAMA) protocol.
- One variant, Link Access Protocol–Balanced (LAP-B), uses the same structure as HDLC but is a scaled-down version of HDLC (i.e., provides fewer of those benefits).
- A version of HDLC called Cisco HDLC (cHDLC) includes a network protocol field.
- cHDLC and HDLC have gradually replaced SDLC.

4.4.2 Ethernet

- Ethernet is a very popular LAN protocol, conceived by Bob Metcalfe in 1973 and developed jointly by Digital, Intel, and Xerox in the 1970s.
- Since then, Ethernet has been further refined and developed into a formal standard called IEEE 802.3ac. There are several versions of Ethernet in use today.
- Ethernet uses a **contention** media access protocol.
- There are several standard versions of Ethernet.
- Figure 4.10a shows an Ethernet 803.3ac frame.
- The frame starts with a 7-byte preamble, (10101010).
- This is followed by a start of frame delimiter, which marks the start of the frame.
- The destination address specifies the receiver, whereas the source address specifies the sender.
- The length indicates the length in 8-bit bytes of the message portion of the frame.
- The VLAN tag field is an optional 4-byte address field used by virtual LANs (VLANs), which are discussed in Chapter 7.

4.4.2 Ethernet (contd)

- The Ethernet frame uses this field only when VLANs are in use; otherwise the field is omitted, and the length field immediately follows the source address field.
- When the VLAN tag field is in use, the first 2 bytes are set to the number 24,832 (hexadecimal 81-00), which is obviously an impossible packet length. When Ethernet sees this length, it knows that the VLAN tag field is in use.
- The DSAP and SSAP are used to pass control information between the sender and receiver.
- The control field is used to hold the frame sequence numbers and ACKs and NAKs used for error control,
- The last 2 bits in the first byte are used to indicate the type of control information being passed and whether the control field is 1 or 2 bytes
- In most cases, the control field is 1-byte long.
- The maximum length of the message is about 1,500 bytes.
- The frame ends with a CRC-32 frame check sequence used for error detection.

Preamble	Start of Frame	Destination Address	Source Address	VLAN Tag	Length	DSAP	SSAP	Control	Data	Frame Check Sequence
7 bytes	1 byte	6 bytes	6 bytes	4 bytes	2 bytes	1 byte	1 byte	1-2 bytes	46-1,500 bytes	4 bytes

FIGURE 4.10a Ethernet 802.3ac frame layout

FIGURE 4.10b
Ethernet II frame layout

Preamble	Start of Frame	Destination Address	Source Address	Type	Data	Frame Check Sequence
7 bytes	1 byte	6 bytes	6 bytes	2 bytes	46-1,500 bytes	4 bytes

4.4.2 Point-to-Point Protocol (PPP)

- PPP was developed in the early 1990s and is often used in WANs.
- It is designed to transfer data over a point-to-point circuit but provides an address so that it can be used on multipoint circuits.
- Figure 4-11 shows the basic layout of a PPP frame, which is very similar to an SDLC or HDLC frame. The frame starts with a flag and has a 1-byte address (which is not used on point-to-point circuits).
- **The control field is typically not used.**
- The protocol field indicates what type of data packet the frame contains (e.g., an IP packet).
- The data field is variable in length and may be up to 1,500 bytes.
- The frame check sequence is usually a CRC-16 but can be a CRC-32.
- The frame ends with a flag.

FIGURE 4-11
PPP frame layout