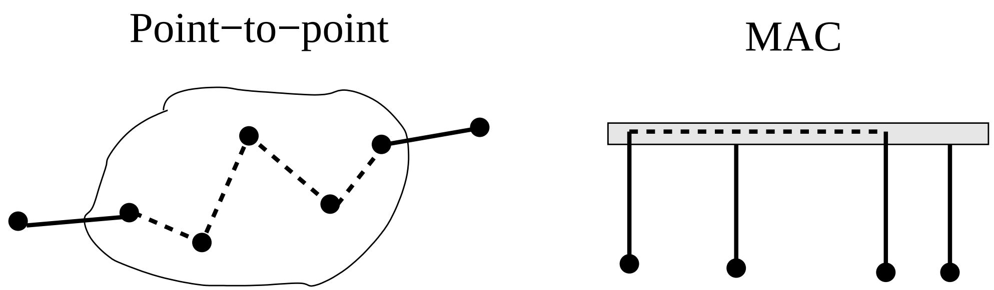
Data Link Layer Overview

- Date link layer deals with two basic issues: **Part I** How data frames can be reliably transmitted, and **Part II** How a shared communication medium can be accessed

- In many networks, such as an LAN, communication medium is shared. So you first have to gain access (II). When you have access, it is equivalent to a "point-to-point" connection (I)

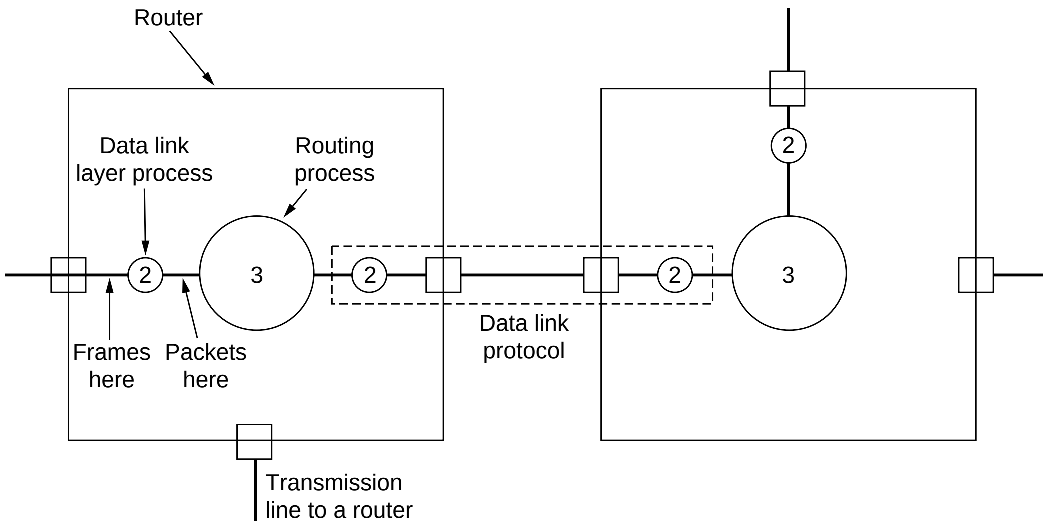


- In 1st part, data link layer has a number of specific functions
  - Provide well-defined service interface to network layer
  - Handle transmission errors (error control) → noisy links becomes error free as far as network layer is concerned
  - Regulate data flow (flow control) → slow receivers not swamped by fast senders

- We will exam how various data link layer protocols deal with these issues

Data Link Layer Overview (continue)

- To see where the part of data link protocol lies, let us recall our 5-layer reference model
  - Physical: describes the transmission of raw bits in terms of mechanical and electrical issues
  - **Data link**: describes how a shared communication channel can be accessed, and **how a data frame can be reliably transmitted**
  - Network: describes how routing is to be done, mostly needed in subnets
  - Transport: this layer provides the actual network interface to applications, does hardest jobs
  - Application: contains the stuff users see, e-mail, remote login, Web's exchange protocol, etc
- **Data link layer protocol** is presented at each link section of point-to-point connection
  - On one hand, physical link may be unreliable and introduce errors
  - One the other hand, router usually does not want to be bothered too much with this, as it has some other "more important things" to worry



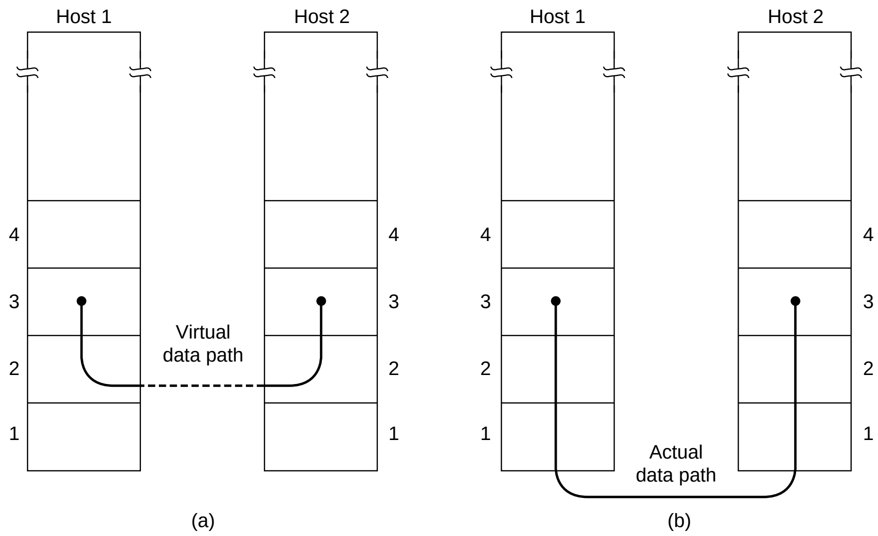
  - It is often up to the data link protocol to **make unreliable communication links look perfect**

Services Provided to Network Layer

- Data link layer is responsible for transmitting data from source network layer to destination network layer

  Source network layer passes a number of bits to data link layer, who then packs them into frames and relies on physical layer to do actual communication

  Receiving data link layer unpacks frames and passes bits to its network layer



- The basic **services** commonly provided are
  - Unacknowledged connectionless: no attempt to recover erroneous or lost frame in layer 2, having least overhead, appropriate when error rate is very low (LANs) so recovery is left to higher layers
  - Acknowledged connectionless: sender knows a frame has arrived safely or not, useful over unreliable channels (wireless systems)
  - Acknowledged connection-oriented: highly reliable but overhead is very high, usually for WANs

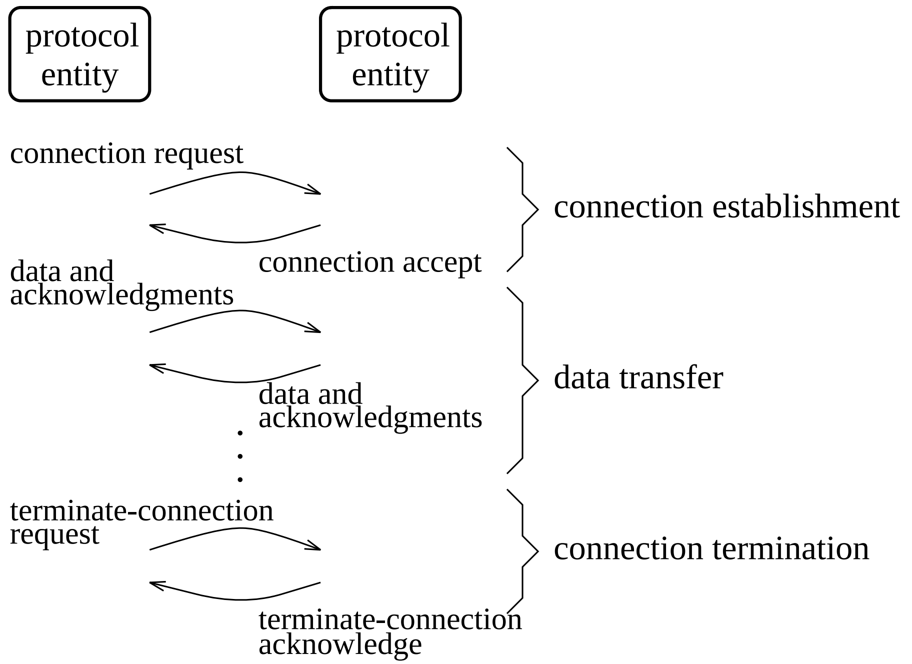
Connection-Oriented / Connectionless

- There are two basic classes of data transfer: **connection-oriented** and **connectionless**

- In connection-oriented data transfer, three phases occur:

  Both sides engaging in communication number PDUs (protocol data units) and keep track of incoming/outgoing numbers. This **sequencing** enables ordered deliver, flow control and error control

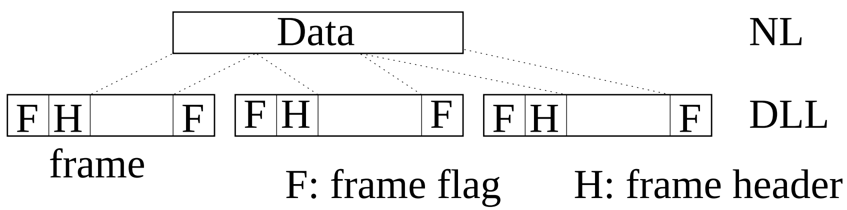
  An example is the virtual circuit



- In connectionless data transfer, data are transferred from one entity to another without the prior mutual construction of a connection. An example is the datagram

Framing

- Data link layer is responsible for making physical link reliable and, to do so, it breaks up network layer data stream into small blocks, a process called segmentation, and adds header and frame flag to each block to form a frame, a process called encapsulation



- Header generally contains three parts or fields
  - **Address**: address of sender and/or receiver
  - **Error detecting code**: a checksum of the frame for error detection
  - **Control**: additional information to implement protocol functions
- The receiving data link layer must know the start and end of a frame → use of frame flag
- **Frame flag**: a unique bit pattern that never appears any other place inside a frame
- Segmentation (at transmitting side)/reassembly (at receiving side), and encapsulation are general operations of a protocol, not just restricted to date link layer
- Note that some physical layer protocols may impose layer-1 framing, simply for physical link synchronisation, which is very different from layer-2 framing, which has higher purposes
- Overhead introduced by a full data link layer protocol can be very high, particularly, in view that data link layer protocol has to be presented at each link section of an end-to-end connection

Frame Flag

- Frame flag must be a special bit pattern that never appears any other place inside a frame, and three general ways of making frame flag are

- **Bit stuffing**: Use 01111110 as frame flag and this bit pattern must be made special

  So sender adds a 0 bit whenever it encounters five consecutive 1 bits in data, and receiver deletes the 0 bit that follows five consecutive 1 bits in the received data

  Examples include high-level data link control, link access procedure balanced for X.25, link access procedure for D-channel for ISDN, and logic link control for FDDI

- **Coding**: Use mb/nb encoding $(n > m) \rightarrow$ data bit patterns 2^m but code bit patterns 2^n, by coding design some of the code bit patterns will never appear in (coded) data stream

  An example is FDDI 4b/5b encoding: two code bit patterns 11000-10001, which never appears in data stream, are used as frame starting flag

- **Physical layer line coding violation**: Recall line coding defines how 0 and 1 bits are transmitted in voltage pulses, and deliberately violating the rule can be used to signify something special

  An example is the ISDN basic rate interface (BRI) frame flag:

  The first bit of the frame, framing bit, is a positive-pulse zero bit (The normal coding rule would require a negative-pulse). The start of a new frame is signalled by this code violation

Error Correction and Detection

- Error control in computer networks is typically achieved by error detecting using **cyclic redundancy check** (CRC), which is the checksum inside a frame, combined with retransmission
- For this purpose, systematic cyclic or polynomial generated codes are often used, which are special linear block codes
- Some basic concepts in error control coding are first reviewed
  - An n-bit frame, which consists of m-bit data and r check bits, is called a **codeword**. All codewords form the **codebook**
  - **Hamming distance** between two codewords is number of bits in which two codewords differ
  - For example, 10001001 and 10110001 have a Hamming distance of 3, where $\oplus$ is bit-wise exclusive OR (modulo 2)

$$\begin{array}{cccccccc} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \quad \oplus \\ \hline 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \end{array}$$

  - The **minimum Hamming distance** $d_{\min}$ of a codebook is the smallest Hamming distance between any pair of codewords in the codebook
  - A codebook with $d_{\min}$ can detect up to $d_{\min} - 1$ errors and correct up to $(d_{\min} - 1)/2$ errors in each codeword

Cyclic Codes

- **Cyclic**: if $< b_{n-1}, b_{n-2}, \cdots, b_1, b_0 >$ is a codeword in the code, $< b_{n-2}, \cdots, b_1, b_0, b_{n-1} >$ is also a codeword in the same code

- Represent m-bit message $\mathbf{a} =< a_{m-1}, \cdots, a_1, a_0 >$ by the polynomial $a(x)$, where a_i are binary-valued (0 or 1)

  $$a(x) = a_{m-1}x^{m-1} + \cdots + a_1x^1 + a_0$$

  A cyclic code is defined by its **generating polynomial**

  $$g(x) = g_r x^r + \cdots + g_1 x^1 + g_0, \;\; g_r = 1, g_0 = 1$$

  That is, n-bit codeword $\mathbf{b} =< b_{n-1}, \cdots, b_1, b_0 >$ of $\mathbf{a}$ is given by

  $$b(x) = x^r \cdot a(x) + \mathrm{Rem}\left(\frac{x^r \cdot a(x)}{g(x)}\right)$$

  The remainder of $x^r \cdot a(x)/g(x)$ is a polynomial up to order x^{r-1}, i.e. providing the r **check bits**. All the calculations use modulo 2 arithmetic

- Example: (7,4) cyclic code with $g(x) = x^3 + x^2 + 1$ and message $\mathbf{a} = 1010$ or $a(x) = x^3 + x^1$

- $x^3 \cdot a(x) = x^6 + x^4$ and $\mathrm{Rem}(x^3 \cdot a(x)/g(x)) = 1 \rightarrow b(x) = x^6 + x^4 + 1$ or $\mathbf{b} = 1010\,001$ (message bits 1010 and check bits 001)

Cyclic Codes (continue)

- In decoding, let received $\mathbf{b}$ be $\hat{b}(x) = b(x) + e(x)$, where nonzero terms in $e(x)$ indicate errors
- Calculate the remainder of $\hat{b}(x)/g(x)$

$$\mathrm{Rem}\left(\frac{b(x)+e(x)}{g(x)}\right) = \mathrm{Rem}\left(\frac{e(x)}{g(x)}\right)$$

  - If zero remainder: no error occurs or errors are undetectable ($e(x)$ is a polynomial containing factor $g(x)$)
  - If nonzero remainder: errors are detected. For example, $\hat{\mathbf{b}} = 1011000 \rightarrow \mathrm{Rem}(\hat{b}(x)/g(x)) = x^2$ is nonzero

- Note that the hardware implementation of CRC is actually very simple: shift register circuit
- Carefully designed generators have become standards, and IEEE 802 uses

$$g(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^{8} + x^{7} + x^{5} + x^{4} + x^{2} + x^{1} + 1$$

- This gives 32 bits of checksum $\longrightarrow$ It has many desired properties, including able to detect all burst errors of length 32 or less and all burst errors affecting an odd number of bits

Flow / Error Control Overview

- If sender transmits frames faster than receiver can accept them, communication will break down
- Flow control is used to prevent this situation happening by limiting the sender into sending no faster than the receiver can handle
  - Basic flow control principle is using some well-defined rules about when sender can transmit next frame
  - These rule often prohibit frames from being sent until receiver has granted permission
  - Data link layer typically uses this type of **feedback-based** flow control
- Frames may be corrupted by noise or get lost, and error control is used to recover from such erroneous situations
  - CRC provides receiver means of detecting errors and typically, protocol calls receiver to send back special control frames bearing positive or negative acknowledgements
  - If sender receives ACK, it knows the frame has arrive safely and if sender receives NACK, it means something has gone wrong and calls a retransmission
- Error control will influence flow control, and in reality, both are closely working together
- We will discuss flow control and error control in more details next lecture

Summary

- Data link layer deals with making physical link reliable and how to access shared communication medium

  In dealing with point-to-point connection, specific functions include providing well-defined service to network layer, flow and error control

- Two basic classes of services: connection-oriented and connectionless

- Framing:

  frame flag — bit stuffing (and similar byte stuffing), coding, line coding violation

  frame header — frame check sequence or checksum, address and control

- Error control coding and systematic cyclic codes