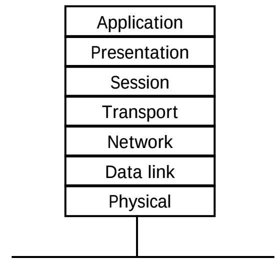
CSCE 515:
Computer Network Programming
OSI Models & Data link layer

Wenyuan Xu

Department of Computer Science and Engineering
University of South Carolina

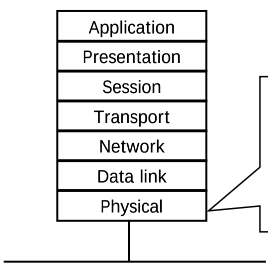
Some slides are made by Dave Hollinger and Badri Nath

Protocol Stack: ISO OSI Model

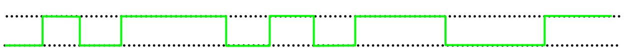


ISO: the International Standards Organization
OSI: Open Systems Interconnection Reference Model (1984)

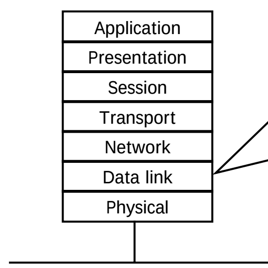
Layer 1: Physical Layer



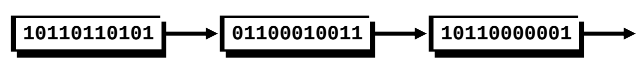
- Responsibilities:
  - Transmission of a raw bit stream
  - Forms the physical interface between devices
- Issues:
  - mechanical and electrical interfaces
  - time per bit
  - distances



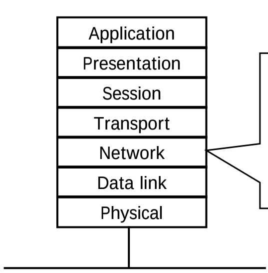
Layer 2: Data Link Layer



- Responsibilities:
  - Provides reliable transfer of information between two adjacent nodes
  - Provides frame-level error control
  - Provides flow control
- Issues:
  - *framing* (dividing data into chunks)
    - header & trailer bits
  - addressing

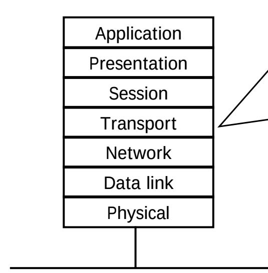


Layer 3: Network Layer



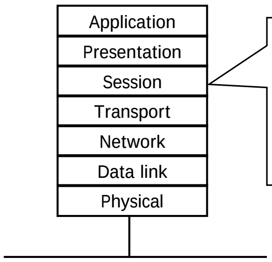
- Responsibilities:
  - path selection between end-systems (routing).
    - Dynamic routing
    - Fixed routing
  - fragmentation & reassembly
  - translation between different network types

Layer 4: Transport Layer



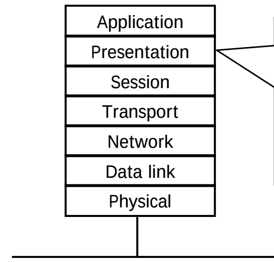
- Responsibilities:
  - provides virtual end-to-end links between peer processes.
  - end-to-end flow control
- Issues:
  - headers
  - error detection
  - reliable communication

Layer 5: Session Layer



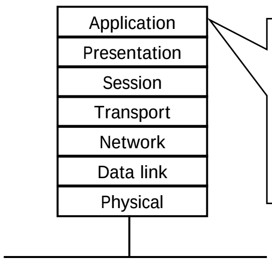
- Responsibilities:
  - Establishes, manages, and terminates a communication session with remote systems
  - Groups several user-level connections into a single “session”
- Many protocol suites do not include a Session Layer.

Layer 6: Presentation Layer



- Responsibilities:
  - Represents data properly
  - Data encryption
  - Data compression
  - Data conversion
- Many protocol suites do not include a Presentation Layer.

Layer 7: Application Layer

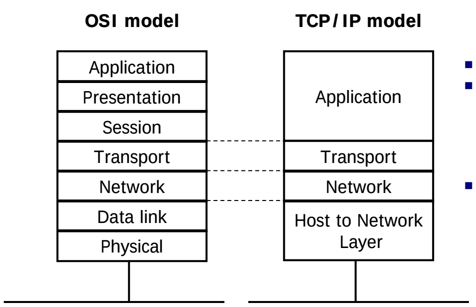


- Responsibilities:
  - Anything not provided by any of the other layers
  - Implements communication between two applications of the same type
    - Examples:
      - FTP
      - HTTP
      - SMTP/POP3/IMAP (email)

Problems

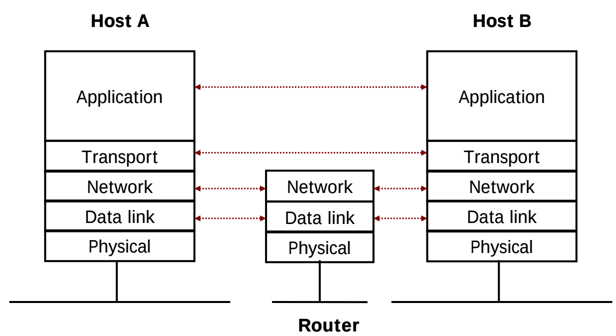
- Seven layers not widely accepted
- Standardized before implemented
- Top three layers fuzzy
- Internet or TCP/IP layering widespread

TCP/IP Layering Architecture

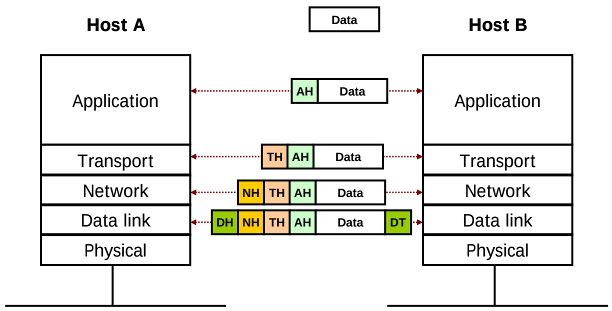


- A simplified model
- The network layer
  - Hosts drop packets into this layer, layer routes towards destination- only promise- try my best
- The transport layer
  - Reliable/unreliable byte oriented stream

Hybrid Reference Model



Header encapsulation and stripping



Layering & Headers

- Each layer needs to add some control information to the data in order to do it's job.
- This information is typically pre-appended to the data before being given to the lower layer.
- Once the lower layers deliver the data and control information - the peer layer uses the control information.

What are the headers?

<u>Physical</u>:

no header - just a bunch of bits.

<u>Data Link:</u>

- address of the receiving endpoints
- address of the sending endpoint
- length of the data
- checksum.

Network layer header - examples

- protocol suite version
- type of service
- length of the data
- packet identifier
- fragment number
- time to live
- protocol
- header checksum
- source network address
- destination network address

Important Summary

- Data-Link:
  - communication between machines on the same network.
- Network:
  - communication between machines on possibly different networks.
- Transport:
  - communication between processes (running on machines on possibly different networks).

Addresses

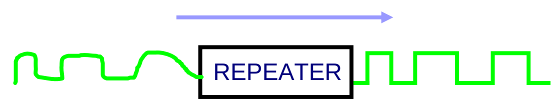
- Each communication endpoint must have an address.
- Consider 2 processes communicating over an internet:
  - the network must be specified
  - the host (end-system) must be specified
  - the process must be specified.

Addresses at Layers

- Physical Layer
  - no address necessary
- Data Link Layer
  - address must be able to select any host on the network.
- Network Layer
  - address must be able to provide information to enable routing.
- Transport Layer
  - address must identify the destination process.

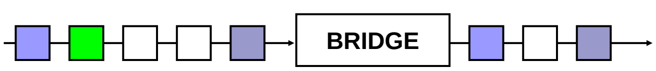
Repeater

- Copies bits from one network to another
- Does not look at any bits
- Allows the extension of a network beyond physical length limitations



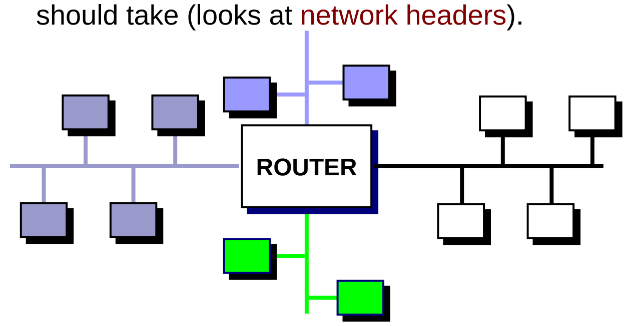
Bridge

- Copies frames from one network to another
- Can operate selectively - does not copy all frames (must look at data-link headers).
- Extends the network beyond physical length limitations.



Router

- Copies packets from one network to another.
- Makes decisions about what *route* a packet should take (looks at network headers).



Gateway

- Operates as a router
- Data conversions above the network layer.
- Conversions:

  encapsulation - use an intermediate network

  translation - connect different application protocols

  encryption - could be done by a gateway

Which layer?

- Repeater & Hub
  - physical layer
- Bridge & Switch
  - data link layer
- Router
  - network layer
- Gateway
  - network layer and above.

Hardware vs. Software

- Repeaters are typically hardware devices.
- Bridges can be implemented in hardware or software.
- Routers & Gateways are typically implemented in software so that they can be extended to handle new protocols.
- Many workstations can operate as routers or gateways.

Data Link Layer Protocol

Date Link Layer Functionality

- Convert bits to signals and recover bits from received signals
  - Encoding
- Decide on a minimum unit for sending bits
  - Frame creation
- Error detection and /or correction of frames
  - Parity, CRC
- Flow control
  - ARQ, Sliding WINDOW

Encoding

- Signals propagate over a physical medium
  - Modulate electromagnetic waves
  - e.g. vary voltage
- Encode binary date onto signals
  - e.g. 0 as low signal and 1 as high signal
  - Known as non-return to zero (NRZ)
  - Non-return to zero inverted (NRZI)
    - Make a transition from current signal to encode a 1; stay at current signal to encode a 0
  - Manchester
    - Transmit xor of the NRZ encoded data and the clock
    - Only 50% efficient

Framing

- The date unit at the date link layer is called a “frame”
- A frame is a group of bits, typically in sequence
- Issues:
  - Frame creation
  - Frame delineation
- Use starting and ending characters (tags) to mark boundaries of frame
  - Problem: what if tag characters occur in the date or control portions of the frame
    - Insert extra escape character when a tag appears in date field

Error Control

- No physical link is perfect
- Bits will be corrupted
- We can either:
  - Detect errors and request retransmission
  - Or correct errors without retransmission

- Error Detection
    - Parity bits
    - Polynomial codes or checksums

Parity bits

- Append a single parity bit to a sequence of bits
- If using 'odd' parity, the parity bit is chosen to make the total number of 1's in the bit sequence odd
- If 'even' parity, the parity bit makes the total number of 1's in the bit sequence even
  - Q: for even parity, what's the parity bit for 00010101?
- Problem: Only detects when there are an odd number of bit errors

Polynomial codes

- Can detect errors on large chunks of data
- Has low overhead
- More robust than parity bit
- Requires the use of a "code polynomial"
  - Example x^2+1
  - Message 1011 -> $1 * x^3 + 0 * x^2 + 1 * x + 1$
    $= x^3 + x + 1$

Cyclic redundancy check

- CRC: Example of a polynomial code
- Procedure:
  - 1. Let *r* be the degree of the code polynomial. Append *r* zero bits to the end of the transmitted bit string. Call the entire bit string *S(x)*
  - 2. Divide *S(x)* by the code polynomial using modulo 2 division.
  - 3. Subtract the remainder from *S(x)* using modulo 2 subtraction.
- The result is the checksummed message

Decoding a CRC

- Procedure
  - 1. Let *n* be the length of the checksummed message in bits
  - 2. Divide the checksummed message by the code polynomial using modulo 2 division. If the remainder is zero, there is no error detected.

Choosing a CRC polynomial

- The longer the polynomial, the smaller the probability of undetected error
- Common standard polynomials:
  - (1) CRC-12: $x^{12} + x^{11} + x^3 + x^2 + x^1 + 1$
  - (2) CRC-16: $x^{16} + x^{15} + x^2 + 1$
  - (3) CRC-CCITT: $x^{16} + x^{12} + x^5 + 1$

Ethernet

Ethernet - A Real Data-Link Layer

- It will be useful to discuss a real data-link layer.
- History
  - developed by Xerox PARC in mid-1970s
  - roots in Aloha packet-radio network
  - standardized by Xerox, DEC, and Intel in 1978
  - similar to IEEE 802.3 standard
- CSMA/CD
  - Multi-access (shared medium)
    - many hosts on 1 wire
  - Carrier sense:
    - can tell when another host is transmitting
  - Collision detection:
    - can tell when another host transmits at the same time

Ethernet

- Addresses
  - unique, 48-bit unicast address assigned to each adapter
  - example: **08:00:e4:b1:20**
  - broadcast: all **1**s
  - multicast: first bit is **1**
- Addresses are assigned to vendors by a central authority
- Bandwidth: 10Mbps, 100Mbps, 1Gbps
- Length: 2500m (500m segments with 4 repeaters)
- Problem: Distributed algorithm that provides fair access

An Ethernet Frame

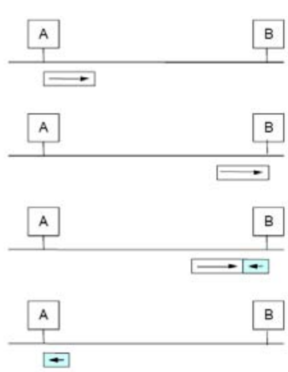
Preamble	Destination Address	Source Address	Len	DATA	CRC
8 bytes	6	6	2	0-1500	4

- The preamble is a sequence of alternating 1s and 0s used for synchronization.
- CRC is Cyclic Redundancy Check

Transmit Algorithm

- If line is idle…
  - send immediately
  - upper bound message size of 1500 bytes
  - must wait 9.6us between back-to-back frames
- If line is busy…
  - wait until idle and transmit immediately

Collisions



Ethernet Backoff Algorithm

- If collision,
  - How to detect collision?
  - How to handle collision?
    - jam for 48 bits, then stop transmitting frame
    - minimum frame is 64 bytes
      - (header + 46 bytes of data) *WHY?*
      - If data portion is less than 46 bytes, *pad* is used.
  - Back off
    - Choose one slot randomly from 2^k slots, where k is the number of collisions the frame has suffered. ([0, 2^k -1])
    - One contention slot length = 2 x end-to-end propagation delay
    - If 16 backoffs occur, the transmission of the frame is considered a failure.

Ethernet Addressing

- Each interface looks at every *frame* and inspects the destination address. If the address does not match the hardware address of the interface (or the broadcast address), the frame is discarded.

- Some interfaces can also be programmed to recognize *multicast* addresses.

Assignment & Next time

- Reading:
  - TI 2.1, 2.2, 2.7, 2.8 **
  - IEEE 802.3 Overview
  - IEEE 802.3 Standard

- Next Lecture:
  - TCP/IP