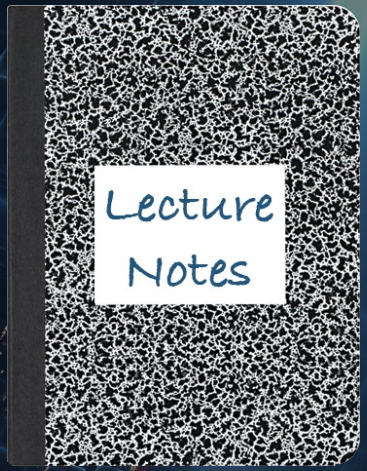


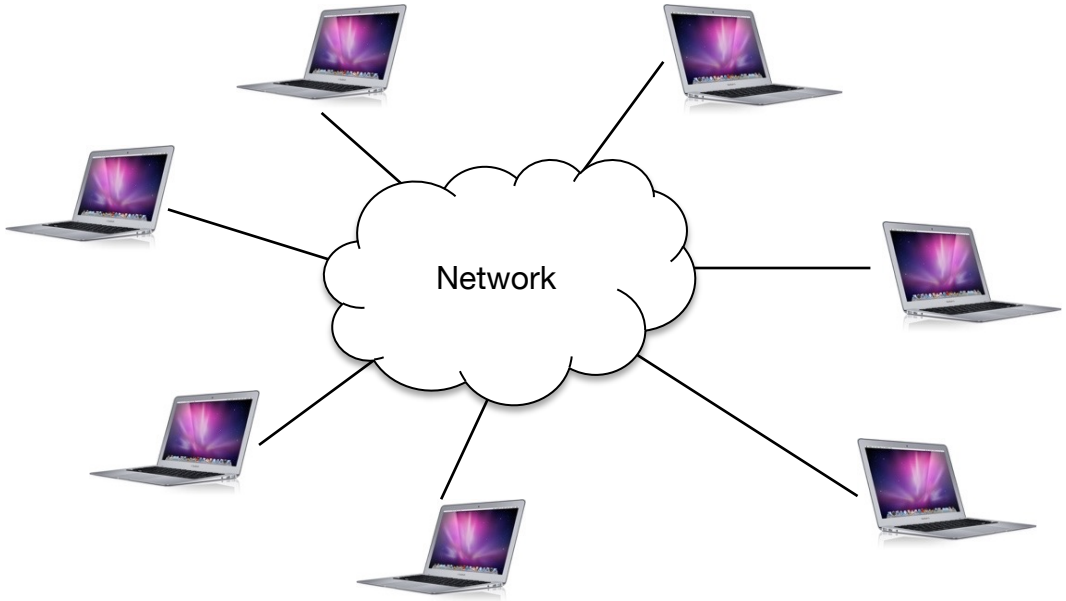
CS 417 – DISTRIBUTED SYSTEMS

Week 1: Part 2 Networking

Paul Krzyzanowski

Inter-computer communication

Without shared memory, computers need a network to communicate



Modes of connection: two design approaches

Circuit switching (virtual circuit)

- Dedicated path (route) – established at setup
- Guaranteed (fixed) bandwidth – routers commit to resources
- Typically fixed-length packets (cells) – each cell only needs a virtual circuit ID
- Constant latency

The Internet Protocol (IP) uses packet switching

Packet switching (datagram)

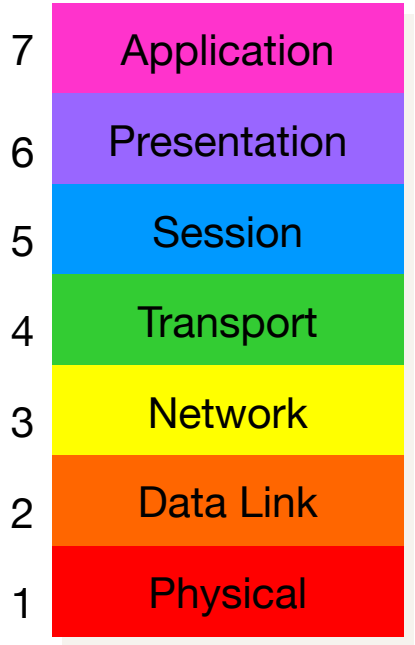
- Shared connection; competition for use with others
- Data is broken into chunks called packets
- Each packet contains a destination address
- available bandwidth $\leq$ channel capacity
- Variable latency

Network Protocol Layering

Most popular model of guiding (not specifying) protocol layers is the **OSI reference model**

Adopted and created by ISO

Specifies 7 layers of protocols



OSI = Open Systems Interconnection
From the ISO = International Organization for Standardization

OSI Reference Model: Layer 1

Transmits and receives raw data to communication medium

Does not care about contents

Media, voltage levels, speed, connectors

Deals with representing bits

1 Physical

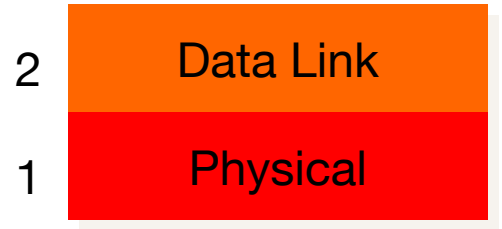
Examples: USB, Bluetooth, 1000BaseT, Wi-Fi radios

OSI Reference Model: Layer 2

Organizes data into frames before passing it down to the hardware.

Detects and corrects errors

Accepts acknowledgements from immediate receiver



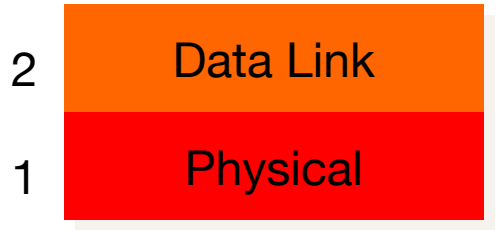
Examples: Ethernet MAC, PPP

OSI Reference Model: Layer 2

An ethernet switch is an example of a device that works on layer 2

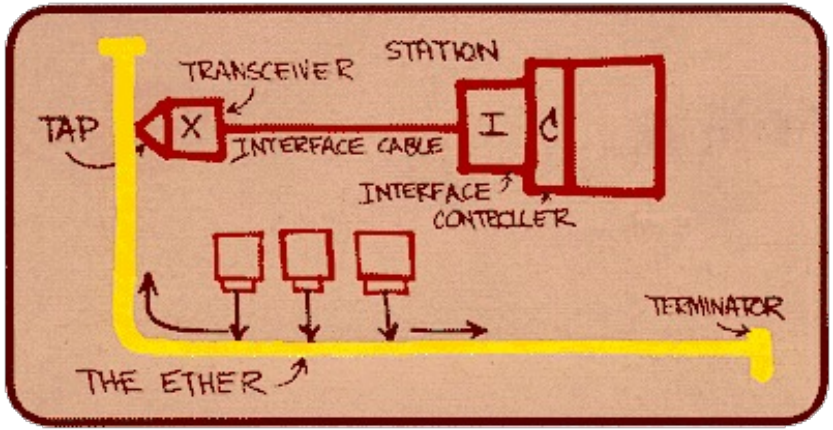
It forwards ethernet frames from one host to another as long as the hosts are connected to the switch (switches may be cascaded)

This set of connected hosts and switches defines the local area network (LAN)



Ethernet

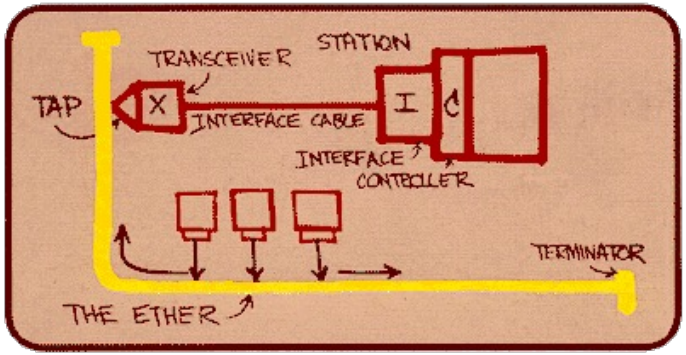
- Packet-based protocol
- Originally designed for shared (bus-based) links
- Each endpoint has a unique ethernet address
  - MAC address: 48-bit number



Created by Bob Metcalfe in 1973

Ethernet

- Packet-based protocol
- Originally designed for shared (bus-based) links
- Each endpoint has a unique ethernet address
  - MAC address: 48-bit number
  - Assigned by the manufacturer of the hardware
- Service guarantees
  - Each packet (frame) contains a CRC checksum
    - The recipient will drop a received frame if it is bad
  - No acknowledgment of packet delivery
  - Ethernet provides unreliable, in-order delivery
    - Packet loss is possible

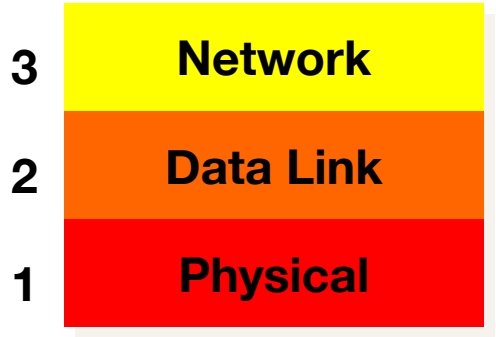


Created by Bob Metcalfe in 1973

OSI Reference Model: Layer 3

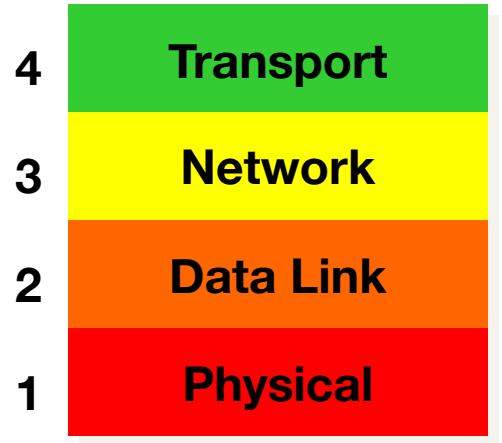
Relay and route information to destination

Manage journey of datagrams and figure out intermediate hops



Examples: IP, X.25

OSI Reference Model: Layer 4

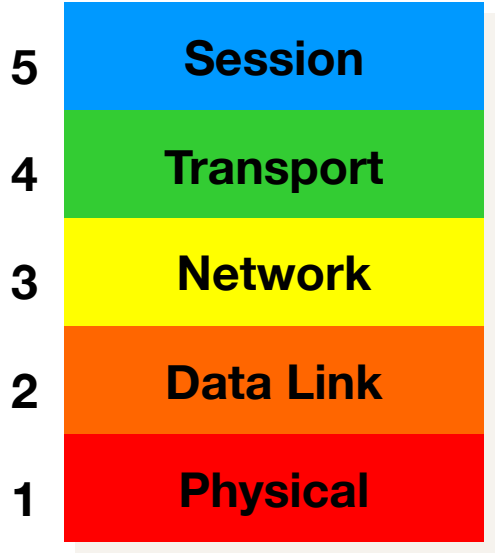


Provides an interface for end-to-end (application-to-application) communication: sends & receives segments of data. Manages flow control. May include end-to-end reliability

Allows an application to set up a message stream to another application.

Examples: TCP, UDP

OSI Reference Model: Layer 5



Services to coordinate dialogue and manage data exchange

Software implemented switch

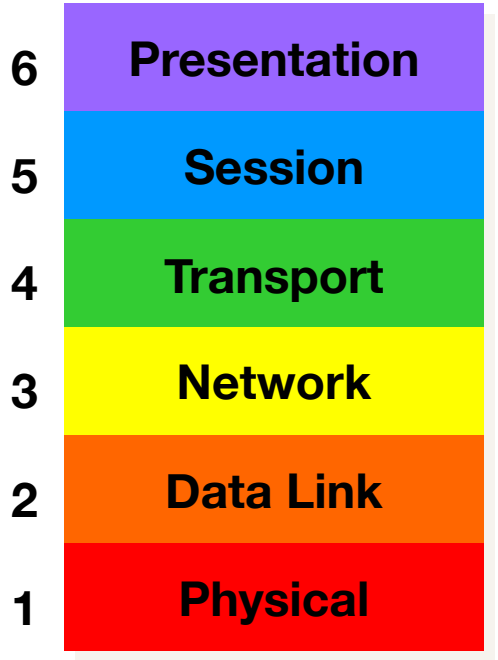
Manage multiple logical connections

Keep track of who is talking: establish & end communications

Deals with data streams

Examples: HTTP 1.1, SSL

OSI Reference Model: Layer 6



Data representation

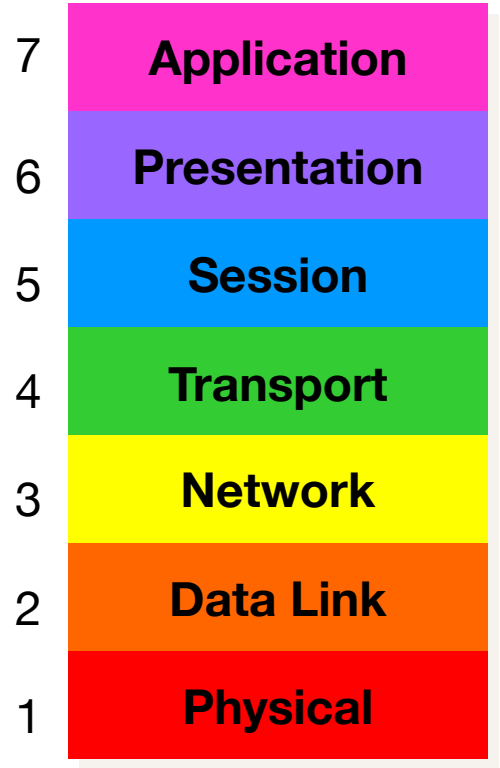
Concerned with the meaning of data bits

Convert between machine representations

Deals with objects

Examples: XDR, ASN.1, MIME, JSON, XML

OSI Reference Model: Layer 7

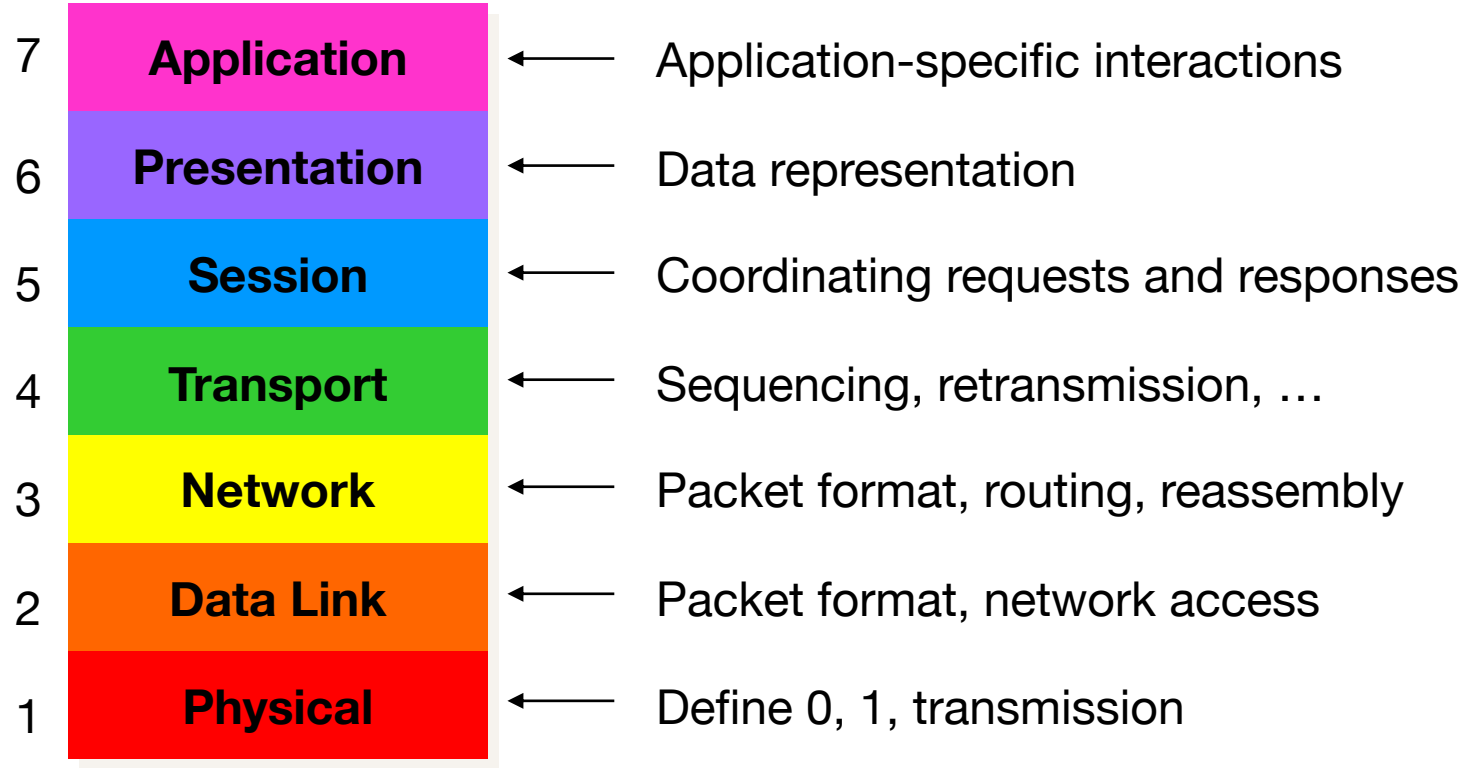


Collection of application-specific protocols

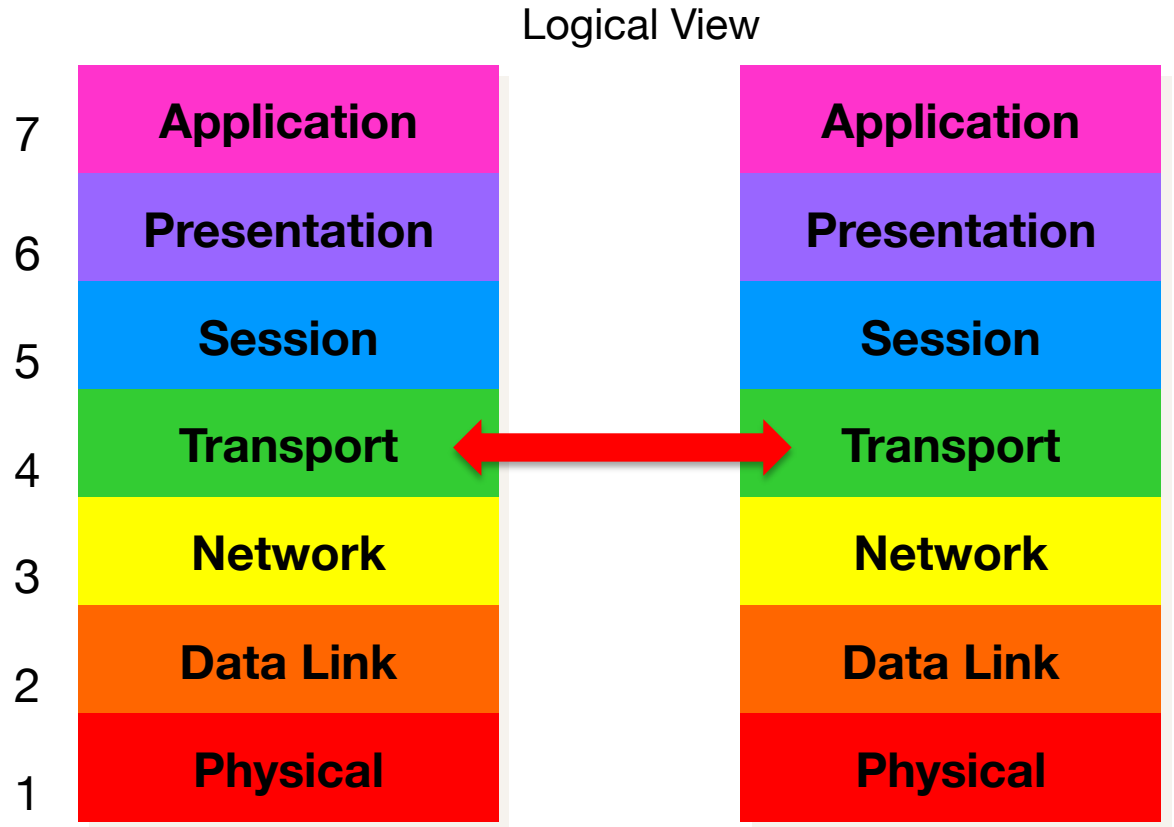
Deals with app-specific protocols

Examples:
- web (HTTP)
- email (SMTP, POP, IMAP)
- file transfer (FTP)
- directory services (LDAP)

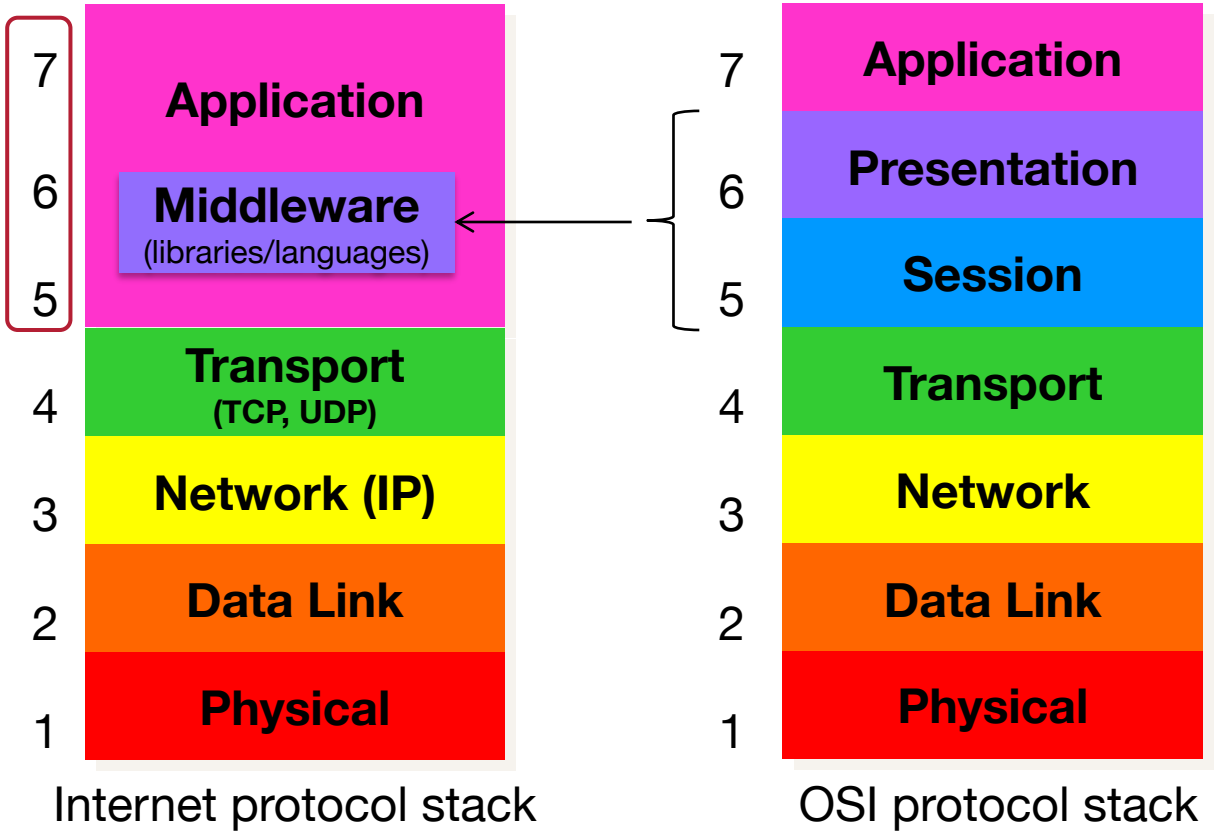
Network Protocols



A layer communicates only with its counterpart



IP protocol layers vs. OSI layers



Protocol Encapsulation

At any layer:

- The higher-level protocol headers are just treated like data
- Lower-level protocol headers can be ignored

An ethernet switch or ethernet driver sees this:

A router or IP driver sees this:

A TCP driver sees this:

An application sees this:

The End