EE 4711

Data Communications and Computer Networks

May 17, 2019

CRMA Electrical Engineering

EE 4711: Data Communications and Computer Networks

Layering

Managing Complexity

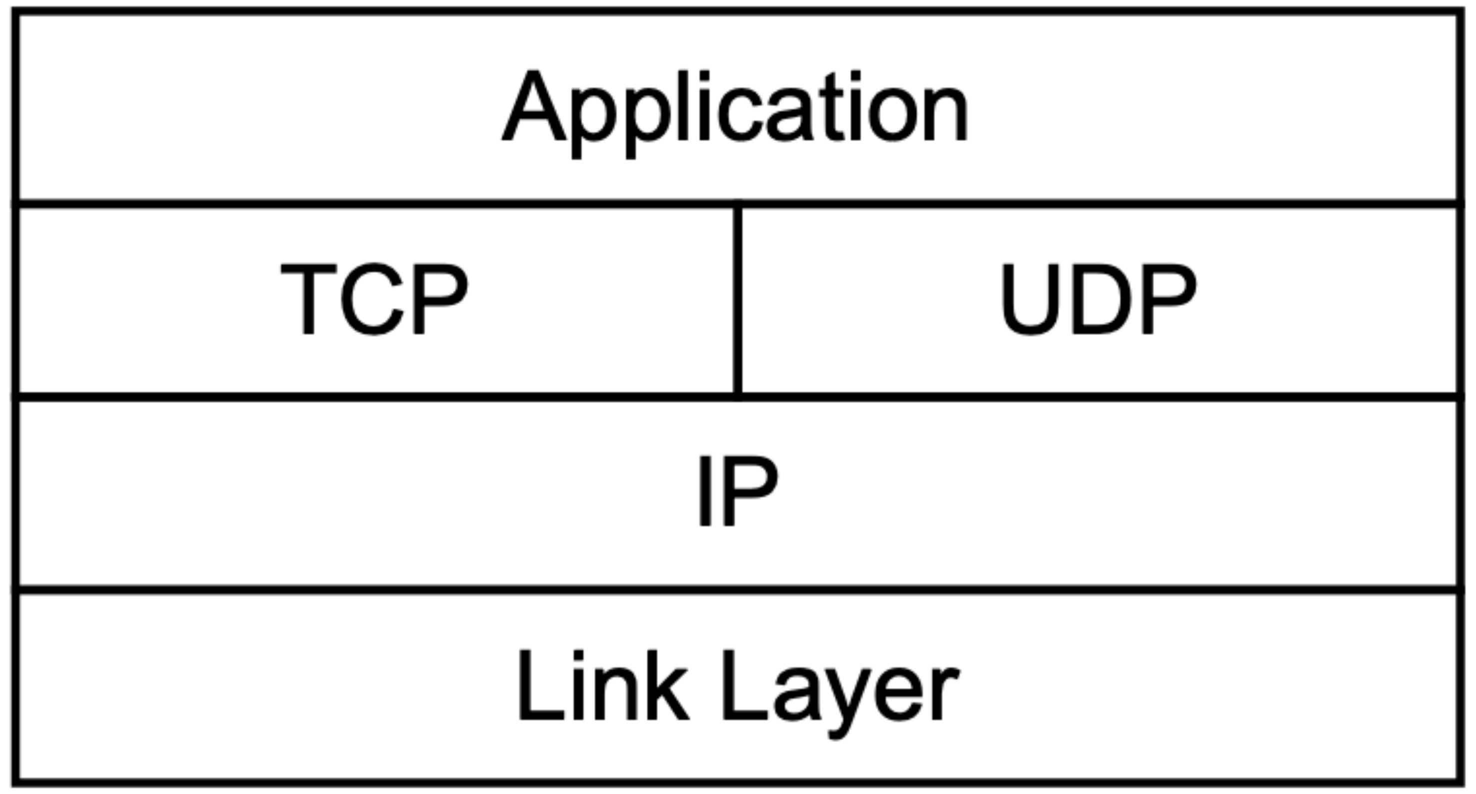
- Very large number of computers
- Incredible variety of technologies
  - Each with very different constraints
- No single administrative entity
- Evolving demands, protocols, applications
  - Each with very different requirements!
- How do we make sense of all this?

CRMA Electrical Engineering

Breakdown into tasks

- Bits on wire
- Packets on wire
- Deliver packets to hosts across local network
- Deliver packets to host across networks
- Deliver packets reliably, to correct process
- Do something with the data

Layering



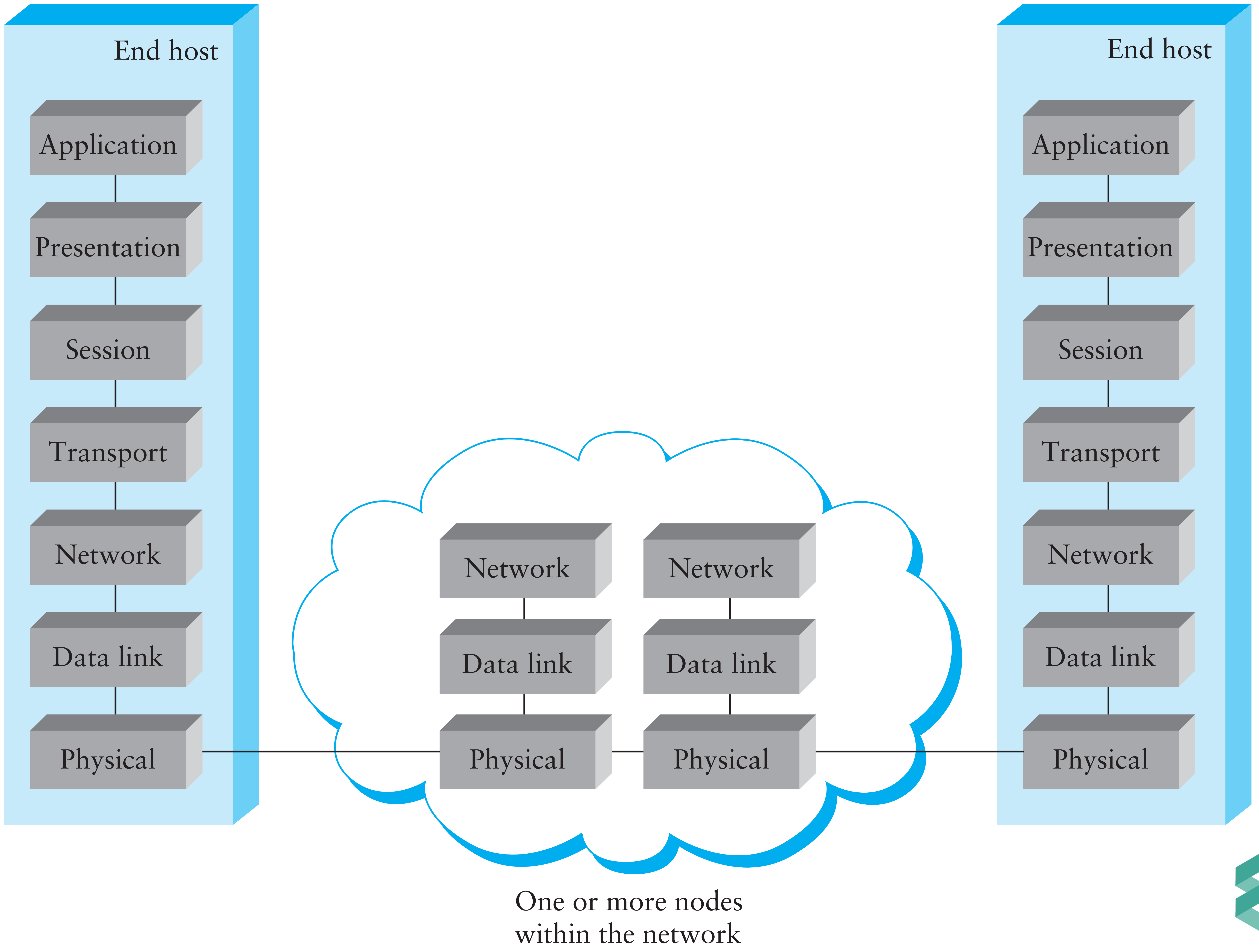
- **Separation of concerns**
  - Break problem into separate parts
  - Solve each one independently
  - Tie together through common interfaces: abstraction
  - Encapsulate data from the layer above inside data from the layer below
  - Allow independent evolution

Layers of the OSI Model

- **Physical Layer:** Transmission/reception of raw bits
- **Data Link Layer:** Maps bits into frames, dictates sharing of common medium, corrects/detects errors , re-orders frames
- **Network Layer:** Routes packets to destination, may perform fragmentation and re-assembly.
- **Transport Layer:** Flow (congestion) control, error control, transparent transport to upper layers
- **Session Layer:** Establishes connection among hosts, duplex, half-duplex, graceful connection termination, combination of streams
- **Presentation Layer:** Negotiation of format of data exchanged between hosts
- **Application layer:** Application services such as FTP, X.400 (mail), HTTP

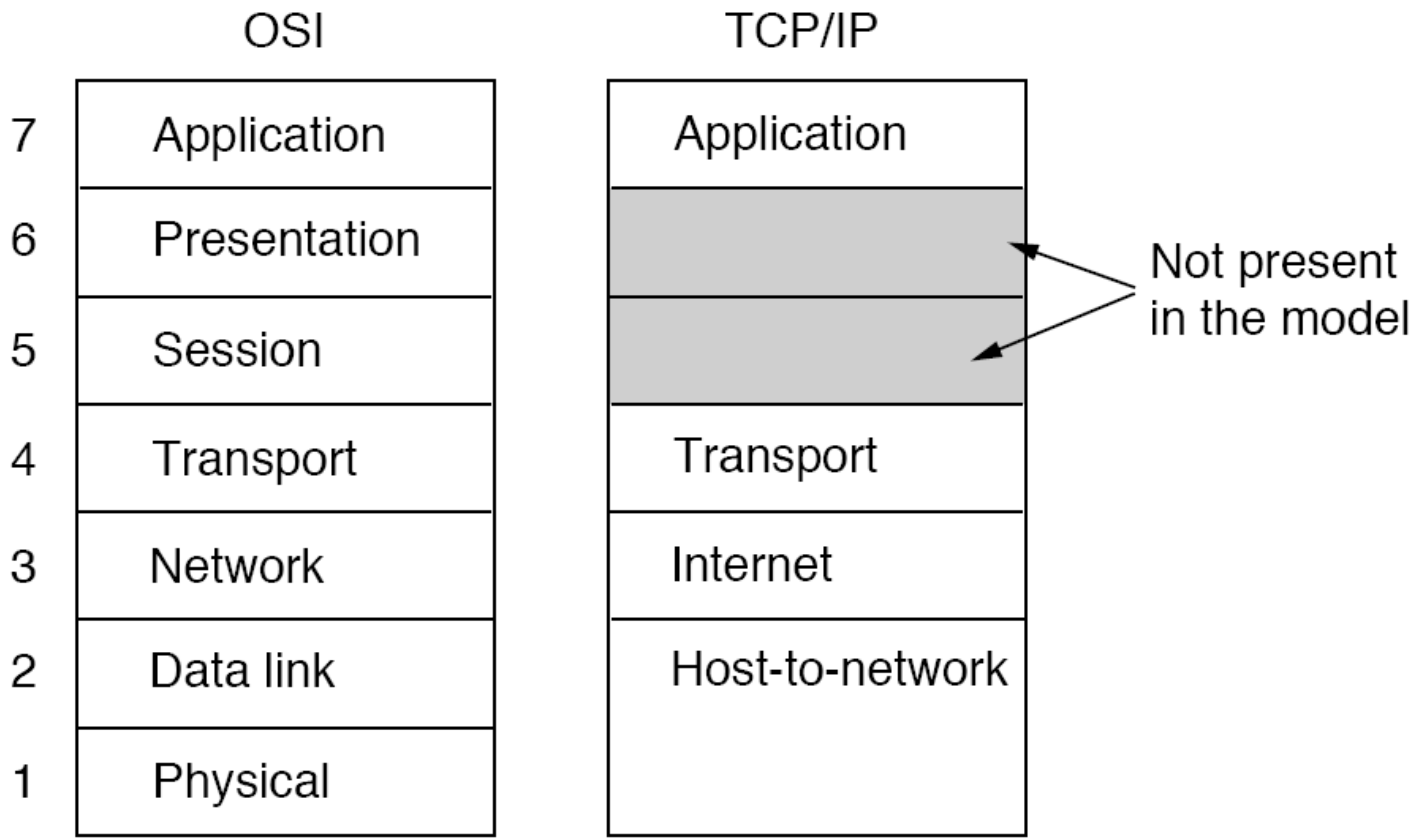
OSI Reference Model

- OSI: Open Systems Interconnection
- 7 layers X. protocol specifications for each layer
- Acts like a reference model rather than a real-world protocol graph
- First three layers are implemented in all nodes



CRMA Electrical Engineering

Comparison of the two architectures

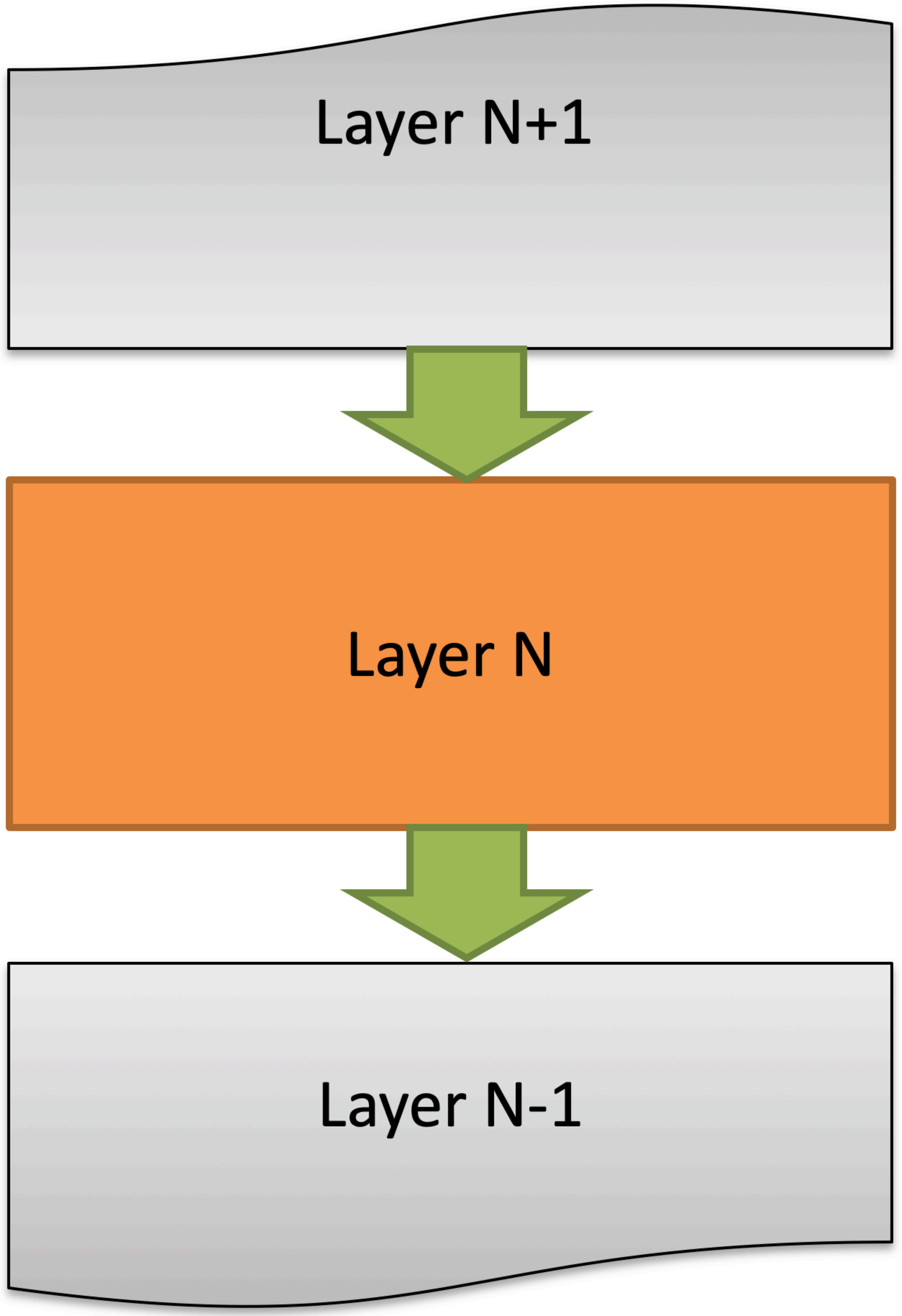


CRMA Electrical Engineering

Motivation for IP Networks

- Communication should continue despite failures
  - Survive equipment failure or physical attack
  - Traffic between two hosts continue on another path
- Support multiple types of communication services
  - Differing requirements for speed, latency, & reliability
  - Bidirectional reliable delivery vs. message service
- Accommodate a variety of networks
  - Both military and commercial facilities
  - Minimize assumptions about the underlying network

Layers, Services, Protocols



Service: abstraction provided to layer above

API: concrete way of using the service

Protocol: rules for communication within same layer

Layer N uses the services provided by N-1 to implement its protocol and provide its own service

CRMA Electrical Engineering

Layers, Services, Protocols

Application	Service: user-facing application. Application-defined messages
Transport	Service: multiplexing applications Reliable byte stream to other node (TCP), Unreliable datagram (UDP)
Network	Service: move packets to any other node in the network IP: Unreliable, best-effort service model
Link	Service: move frames to other node across link. May add reliability, medium access control
Physical	Service: move bits to other node across link

Protocols

- What do you need to communicate?
  - -Definition of message formats
  - -Definition of the semantics of messages
  - -Definition of valid sequences of messages
    - -Including valid timings
- Also, who do you talk to? ...

What gets implemented at the end host

- Bits arrive on wire, must make it up to application
- Therefore, all layers must exist at host!

What gets implemented in the network?

- Bits arrive on wire

  Physical layer necessary

- Packets must be delivered to next-hop and across local networks

  Data link layer necessary

- Packets must be delivered between networks for global delivery

  Network layer necessary

- The network doesn’t support reliable delivery

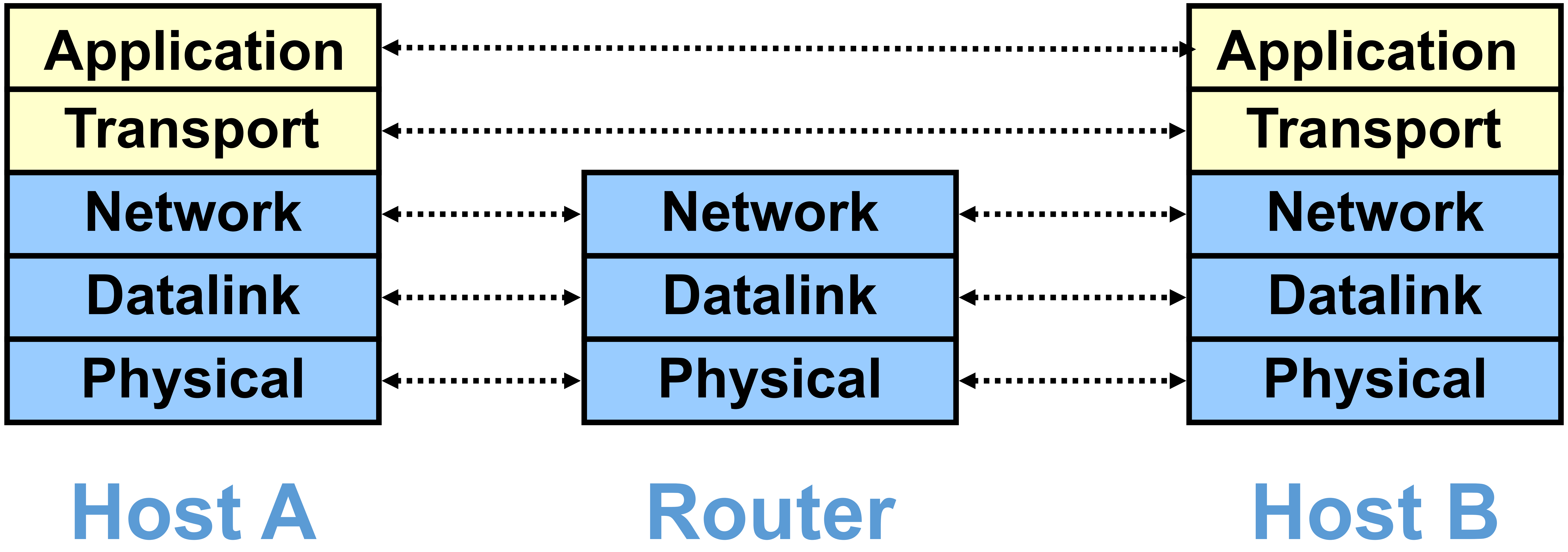
  Transport layer (and above) not supported

Switches vs. Routers

- Switches do what routers do, except they don't participate in global delivery, just local delivery
- Switches only need to support Physical and Data link
  - Don't need to support Network layer
- Routers support Physical, Datalink and Network layers
- Won't focus on the router/switch distinction
  - When I say switch, I almost always mean router
  - Almost all boxes support network layer these days

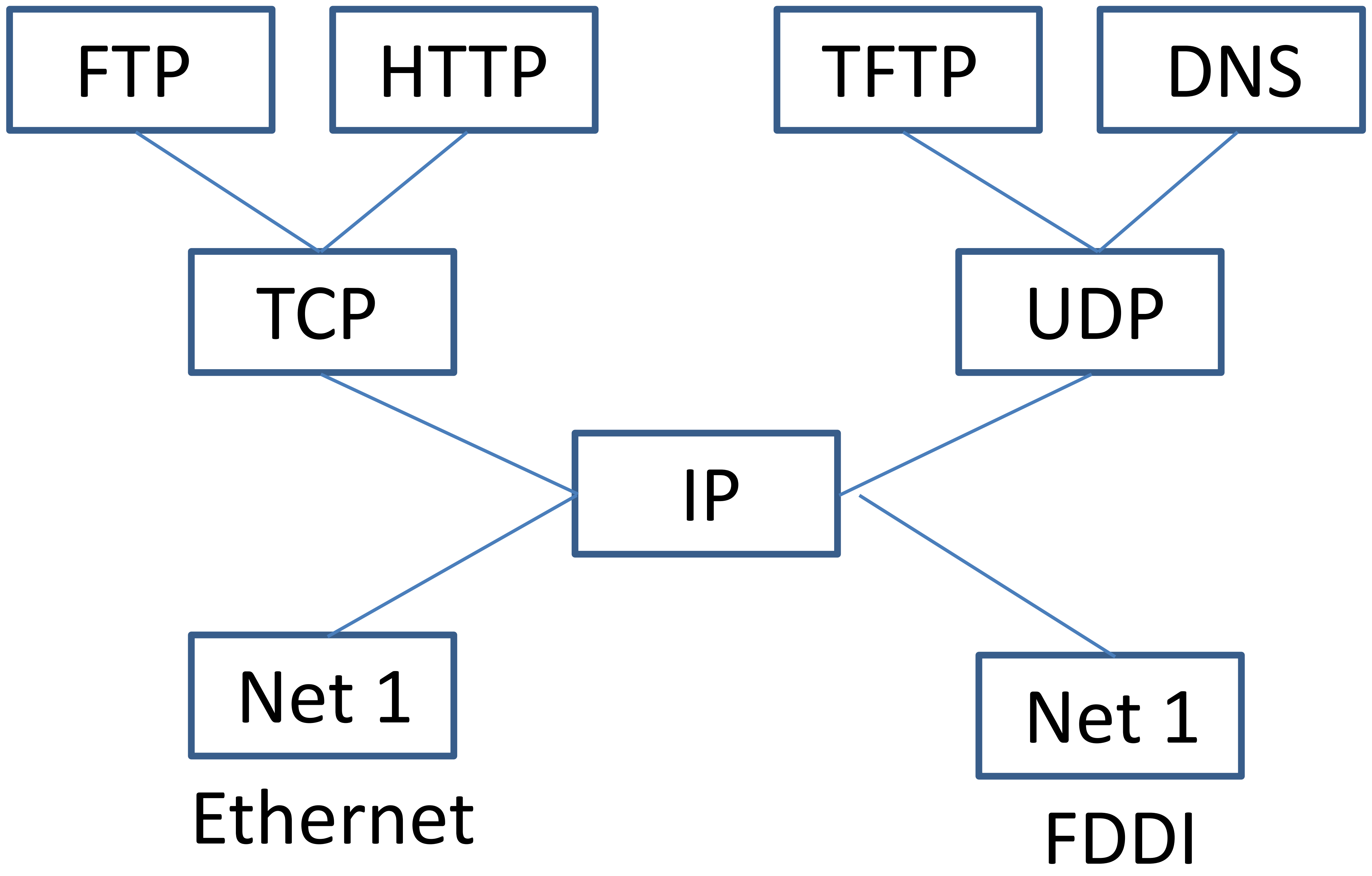
Simple diagram

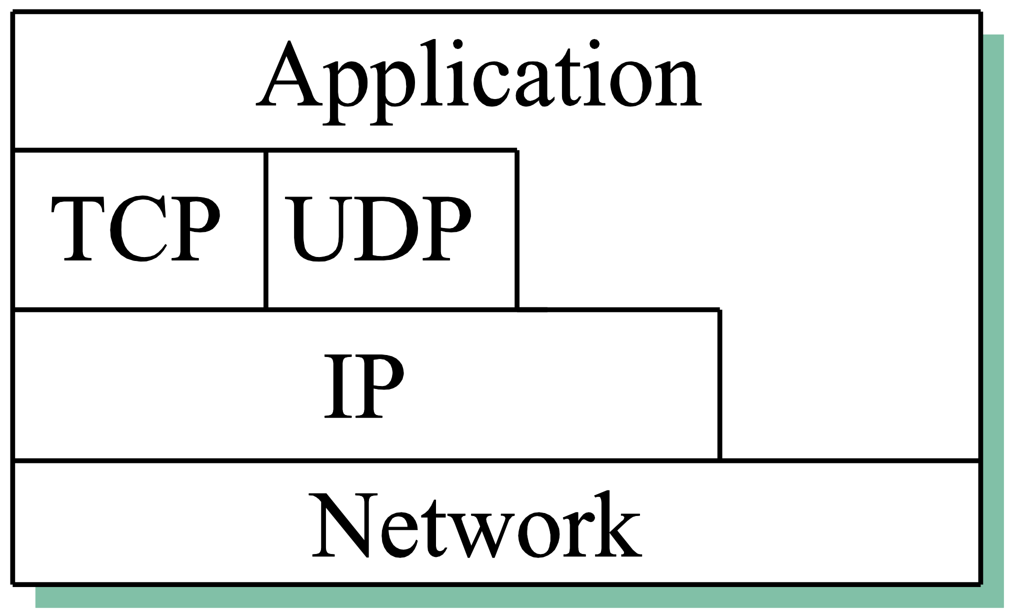
- Lower three layers implemented everywhere
- Top two layers implemented only at hosts



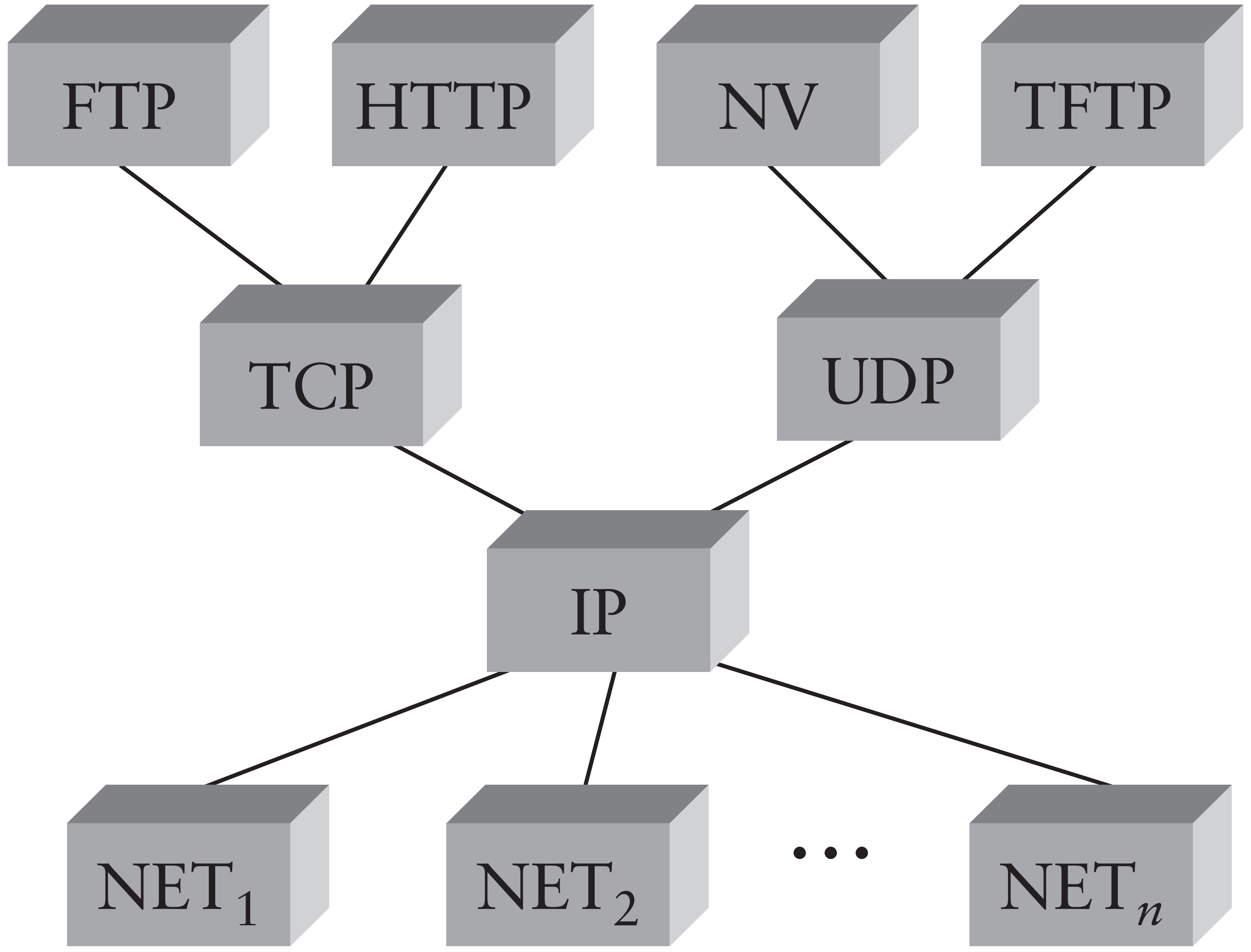
CRMA Electrical Engineering

Summary: The Internet Architecture





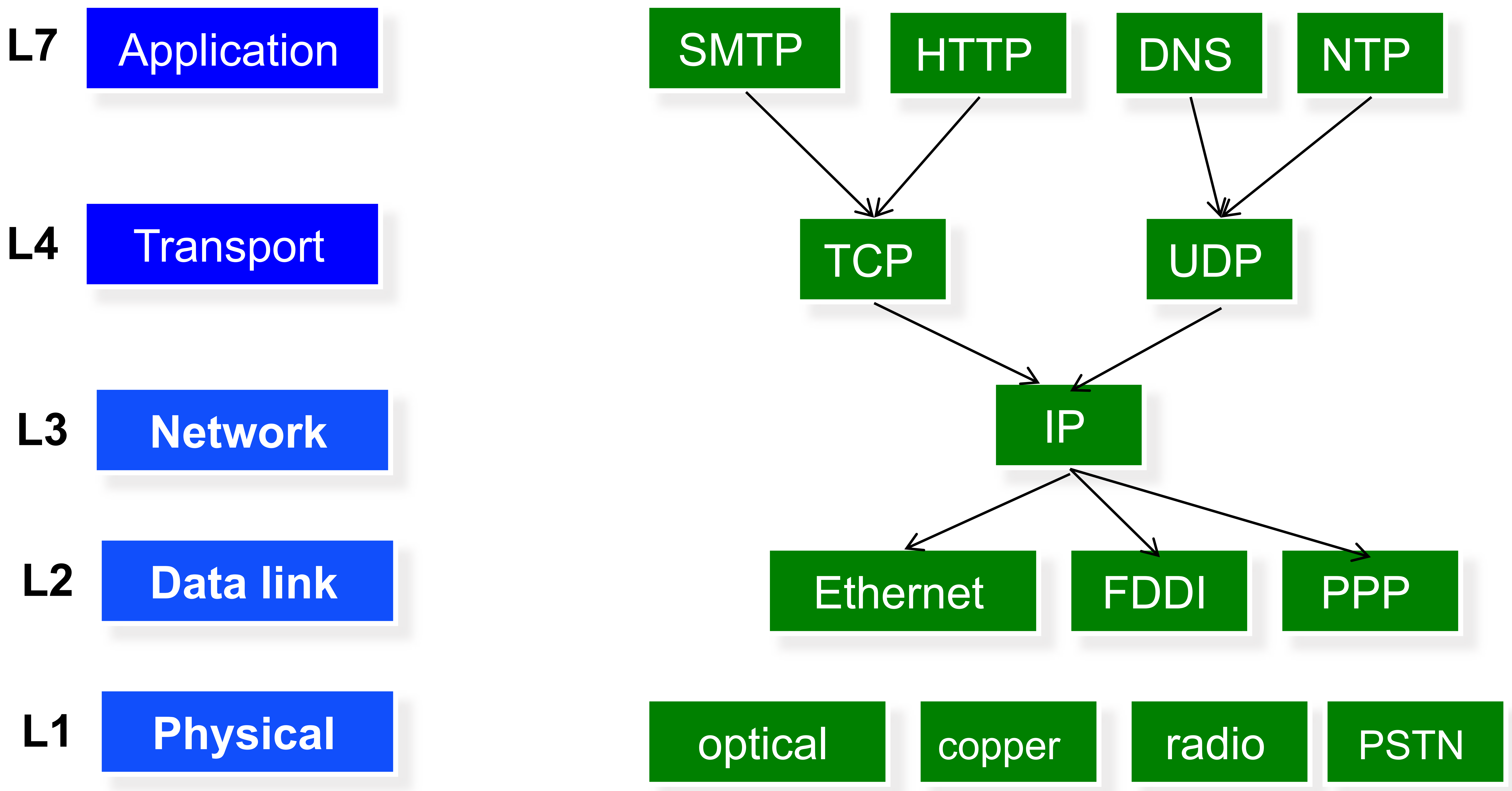
IP as the Narrow Waist



- Many applications protocols on top of UDP & TCP
- IP works over many types of networks
- This is the “Hourglass” architecture of the Internet.
  - If every network supports IP, applications run over many different networks (e.g., cellular network)

CRMA Electrical Engineering

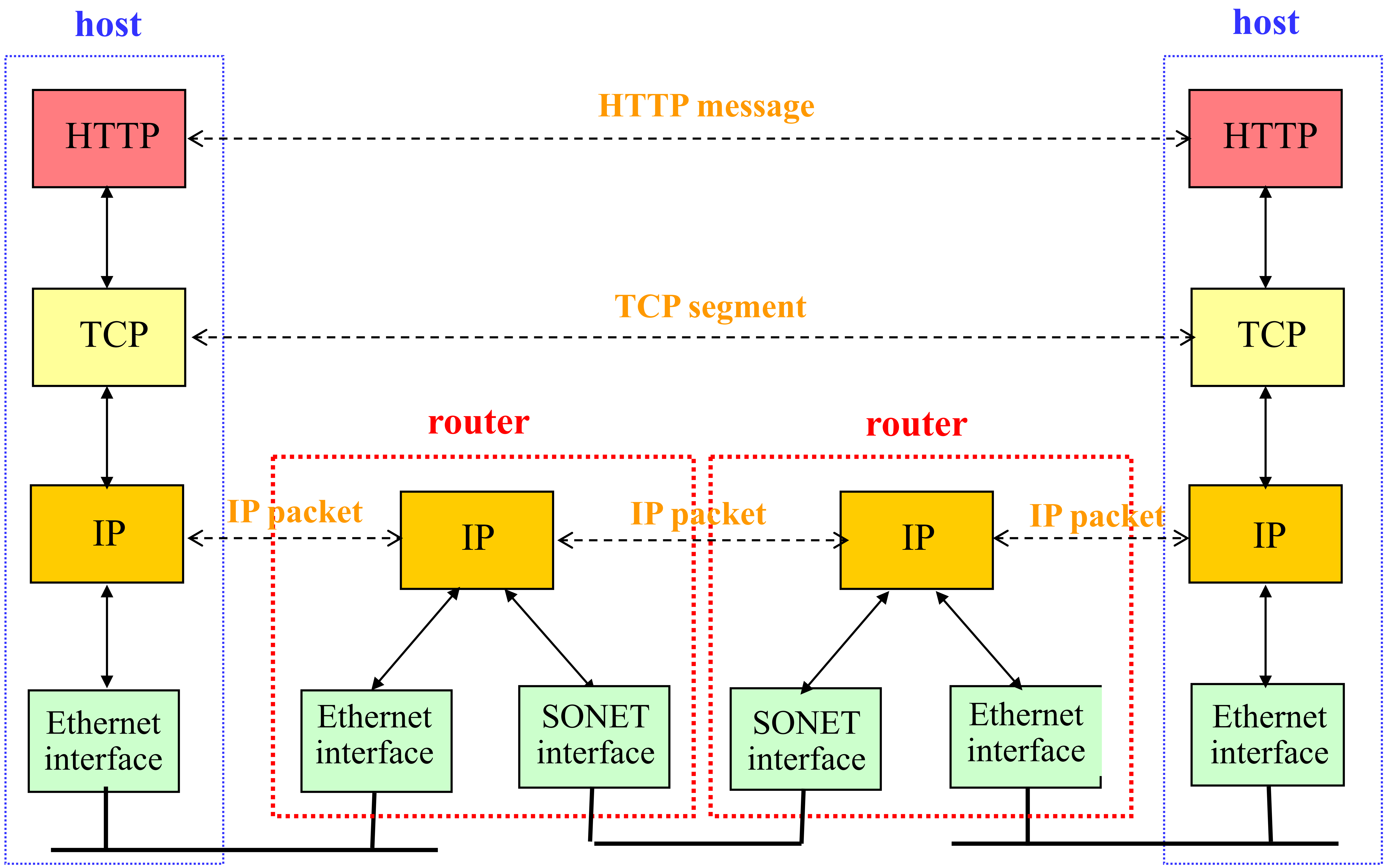
Protocols at different layers



There is just one network-layer protocol, IP. The “narrow waist” of the Internet hourglass

CRMA Electrical Engineering

A Protocol-Centric Diagram



CRMA Electrical Engineering

Naming/Addressing

- Each node typically has a unique* name
  - When that name also tells you how to get to the node, it is called an address
- **Each layer can have its own naming/addressing**
- **Routing** is the process of finding a path to the destination
  - In packet switched networks, each packet must have a destination address
  - For circuit switched, use address to set up circuit
- Special addresses can exist for broadcast/multicast/anycast

Challenges

- Decide on how to factor the problem
  - What services at which layer?
  - What to leave out?
- For example:
  - IP offers pretty crappy service, even on top of reliable links... why?
  - TCP: offers reliable, in-order, no-duplicates service. Why would you want UDP?

Network Layer: Internet Protocol (IP)

- Used by most computer networks today
  - Runs over a variety of physical networks, can connect Ethernet, wireless, modem lines, etc.
- Every host has a unique 4-byte IP address (IPv4)
  - E.g., www.crma.ac.th –> 164.115.18.43
  - The network knows how to route a packet to any address
- Need more to build something like the Web
  - Need naming (DNS)
  - Interface for browser and server software
  - Need demultiplexing within a host: which packets are for web browser, Skype, or the mail program?

CRMA Electrical Engineering

Transport: UDP and TCP

- **UDP and TCP most popular protocols on IP**
  - Both use 16-bit port number & 32-bit IP address
  - Applications bind a port & receive traffic on that port
- **UDP – User (unreliable) Datagram Protocol**
  - Exposes packet-switched nature of Internet
  - Adds multiplexing on top of IP
  - Sent packets may be dropped, reordered, even duplicated (but there is corruption protection)
- **TCP – Transmission Control Protocol**
  - Provides illusion of reliable 'pipe' or 'stream' between two processes anywhere on the network
  - Handles congestion and flow control

CRMA Electrical Engineering

Uses of TCP

- **Most applications use TCP**
  - Easier to program (reliability is convenient)
  - Automatically avoids congestion (don’t need to worry about taking down the network)

Uses of TCP

- **Servers typically listen on well-known ports:**
  - 20: File Transfer Protocol (FTP) Data Transfer
  - 21: File Transfer Protocol (FTP) Command Control
  - 22: Secure Shell (SSH) Secure Login
  - 23: Telnet remote login service, unencrypted text messages
  - 25: Simple Mail Transfer Protocol (SMTP) E-mail routing
  - 53: Domain Name System (DNS) service
  - 80: Hypertext Transfer Protocol (HTTP) used in the World Wide Web
  - 110: Post Office Protocol (POP3)
  - 119: Network News Transfer Protocol (NNTP)
  - 123: Network Time Protocol (NTP)
  - 143: Internet Message Access Protocol (IMAP) Management of digital mail
  - 161: Simple Network Management Protocol (SNMP)

What are some of the benefits of protocols and layering?

Interoperability

- **Many implementations of many technologies**
  - Hosts running FreeBSD, Linux, Windows, MacOS, ...
  - People using Mozilla, Explorer, Opera, ...
  - Routers made by cisco, juniper, ...
  - Hardware made by IBM, Dell, Apple, ...
- **And it changes all the time.**

But they can all talk together because they use the same protocol(s)

Abstraction & Reuse

- **Multiple choices of protocol at many layers**
  - Physical: copper, fiber, air, carrier pigeon
  - Link: ethernet, token ring, SONET, FDDI
  - Transport: TCP, UDP, SCTP
- **But we don’t want to have to write “a web (HTTP) browser for TCP networks running IP over Ethernet on Copper” and another for the fiber version…**
  - Protocols provide a standard interface to write to
  - Layers hide the details of the protocols below

Decoupling aids innovation

- Technologies at each layer pursued by very different communities
- Innovation at each layer can proceed in parallel

What are some of the drawbacks of protocols and layering?

Drawbacks of Layering

- **Layer N may duplicate lower layer functionality**
  - e.g., error recovery to retransmit lost data
- **Information hiding may hurt performance**
  - e.g., packet loss due to corruption vs. congestion
- **Headers start to get really big**
  - e.g., typical TCP+IP+Ethernet is 54 bytes
- **Layer violations when the gains too great to resist**
  - e.g., TCP-over-wireless
- **Layer violations when network doesn't trust ends**
  - e.g., firewalls