Networking

Basic Network Abstraction

- A process can create "endpoints", used to exchange messages with other processes
- Each endpoint has a unique address
- A message is a byte array
- Processes can:
  - receive messages on endpoints
  - send messages on endpoints

Network "protocol"

- Agreement between processes about what makes a well-formed message
  - Syntax
    - Layout of bits, bytes, fields, etc
  - Semantics
    - what messages, and fields within messages, mean
- Example: HTTP get request and response
  - Retrieves information identified by a URI (Universal Resource Identifier)

Layering

- The network abstraction is usually layered
  - like Dijkstra's THE Operating system

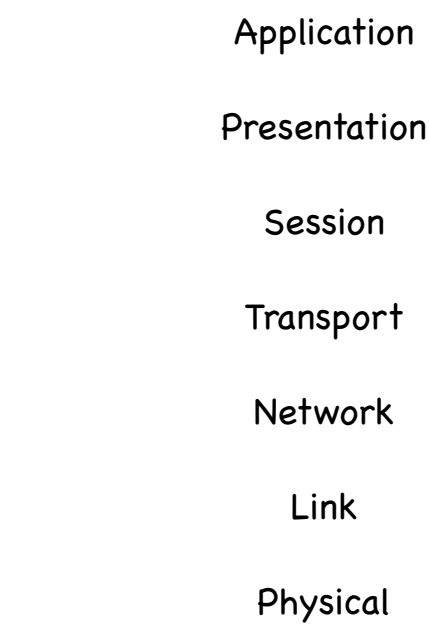
Layer 5	User ("Not implemented by us" EWD)
Layer 4	User Programs
Layer 3	I/O
Layer 2	OS-Console communication
Layer 1	Memory management (Pager)
Layer 0	Multiprogramming (semaphores, interrupts, scheduling)

Layers support Modularity

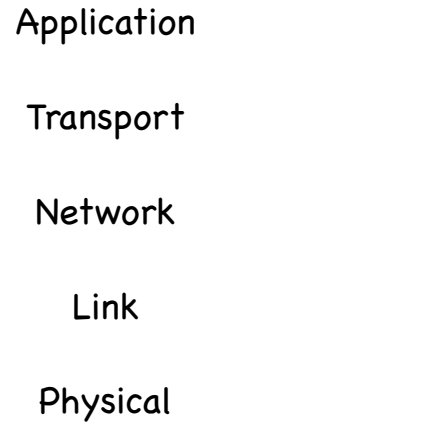
- Each layer
  - relies on services from layer below
  - exports services to layer above
- Interfaces between layers
  - Hide implementation details
  - Ease maintenance, updates
    - changes in the implementation of layer's service are transparent to the rest of the system

Network Layering

- The network abstraction is usually layered



Proposed 7-Layer
ISO/OSI Reference Model
(1970s)

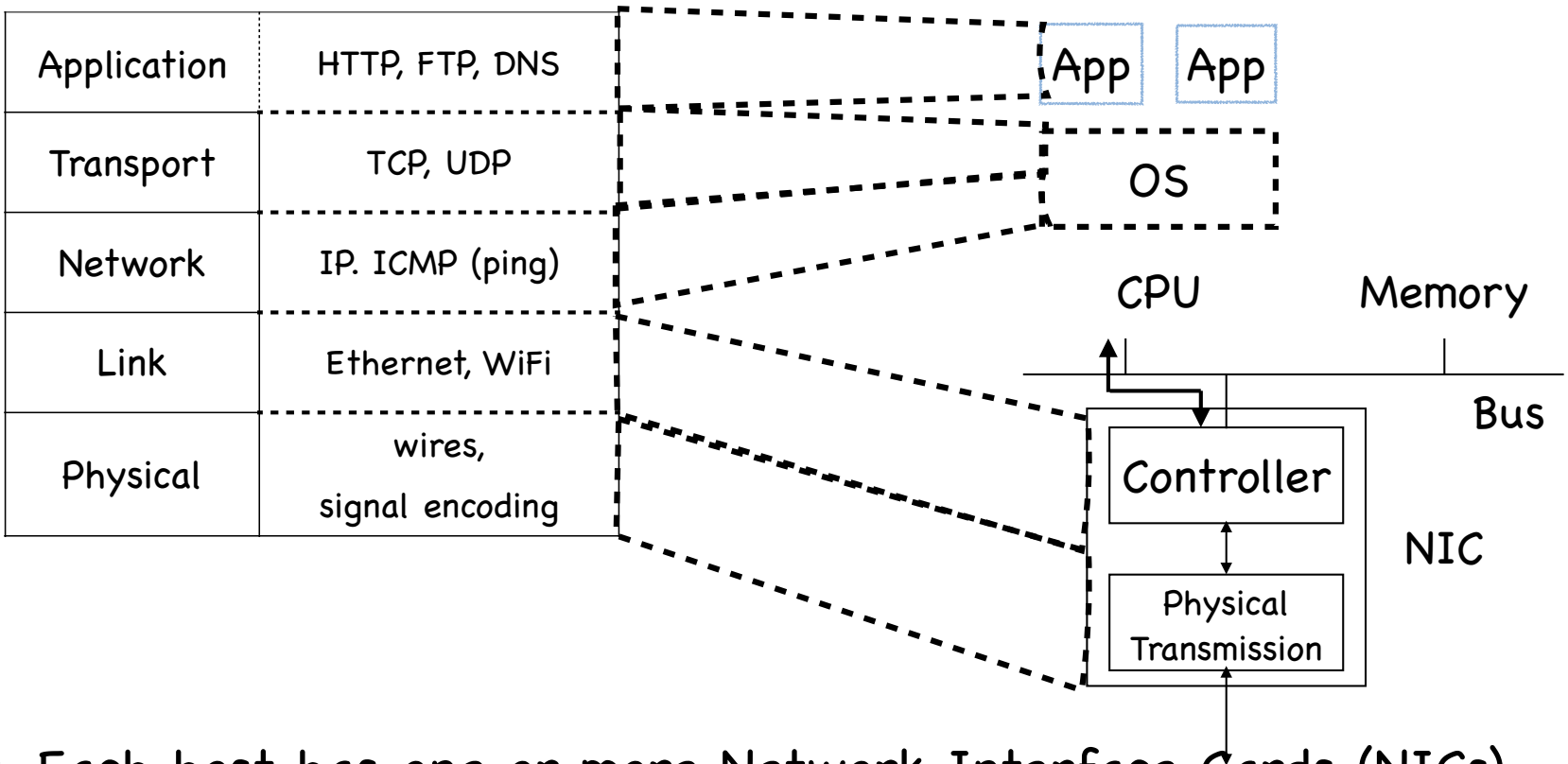


Actual 5-Layer
Internet Protocols Stack

Internet Protocol Stack

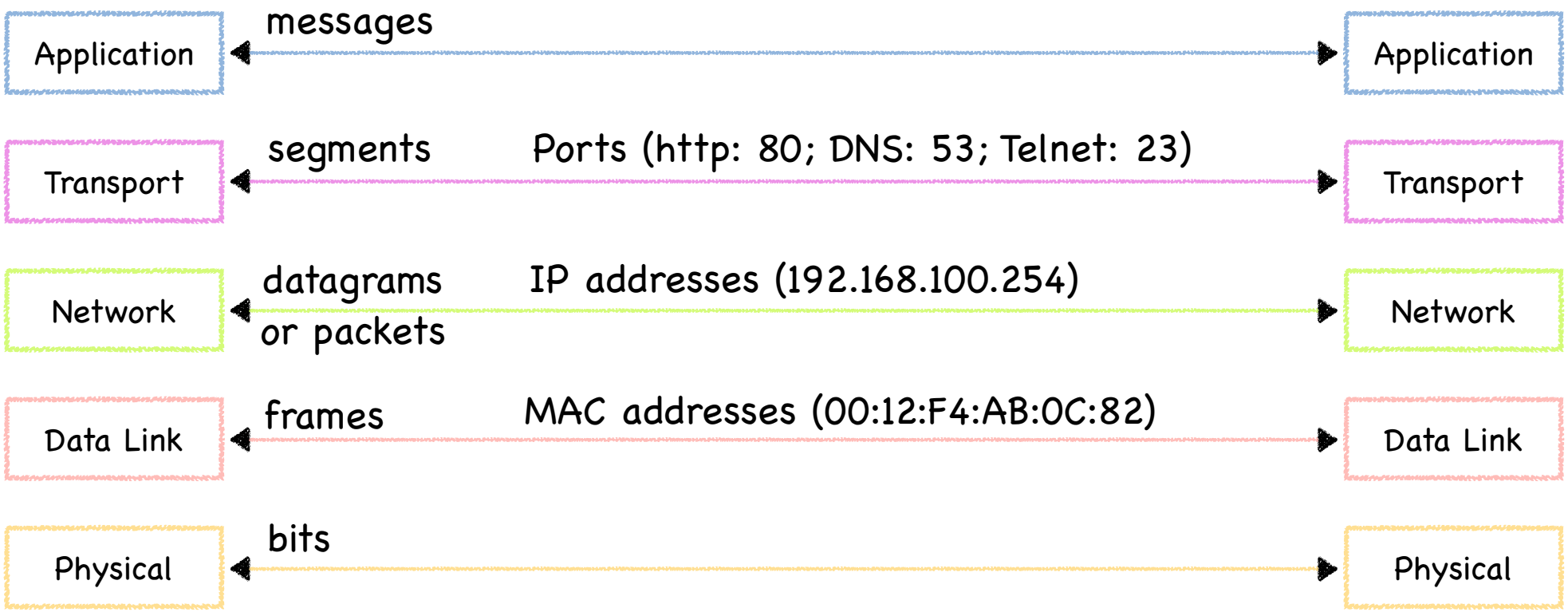
Application	Exchanges messages	HTTP, FTP, DNS
Transport	Transports messages; Exchanges segments	TCP, UDP
Network	Transports segments; exchanges datagrams	IP, ICMP (ping)
Link	Transports datagrams; Exchanges frames	Ethernet, WiFi
Physical	Transport frames; exchanges bits	Wires, signal encoding

Who does what?

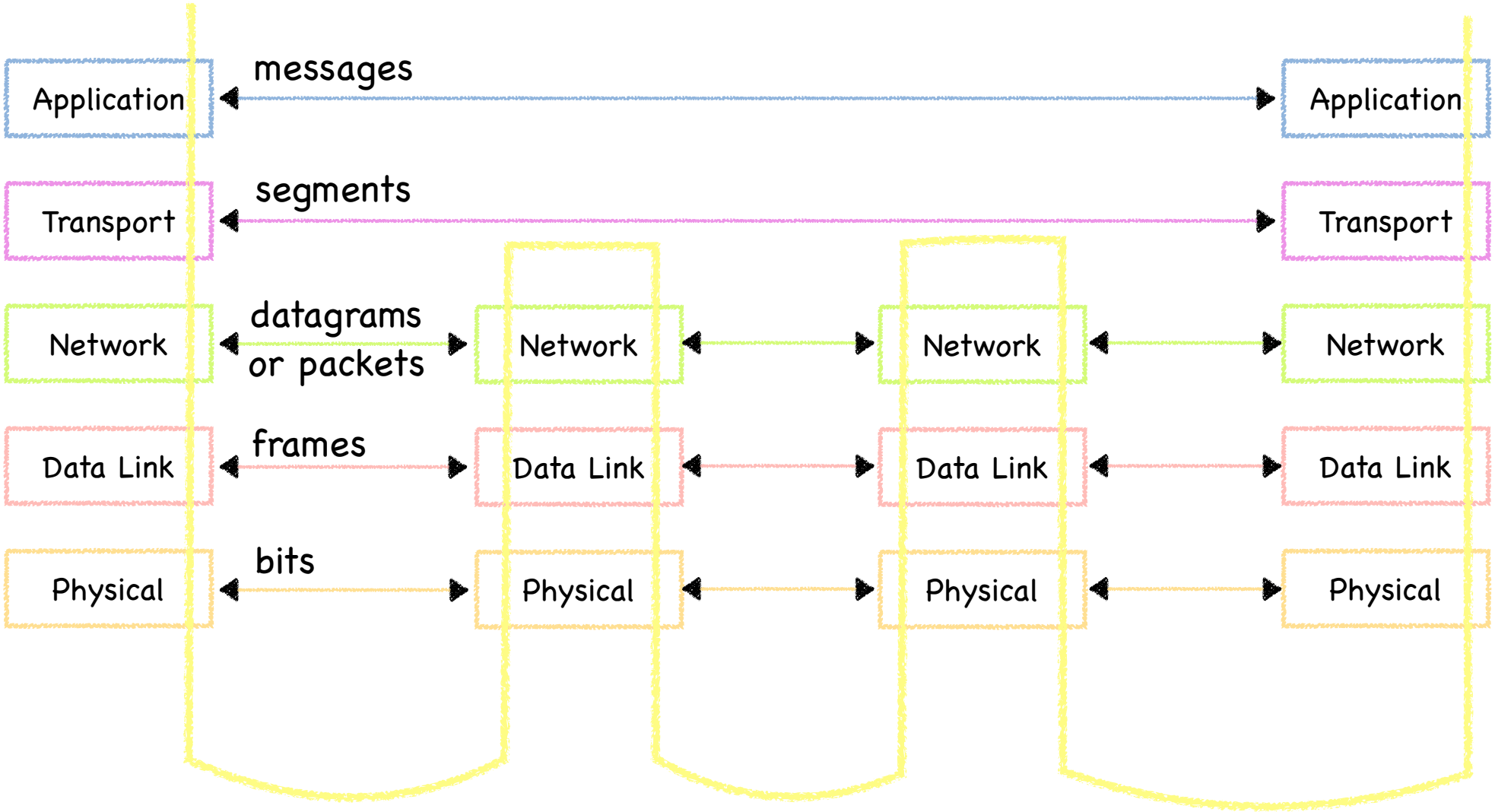


- Each host has one or more Network Interface Cards (NICs)
  - Attached to host's system buses
  - Combination of hardware, software, firmware

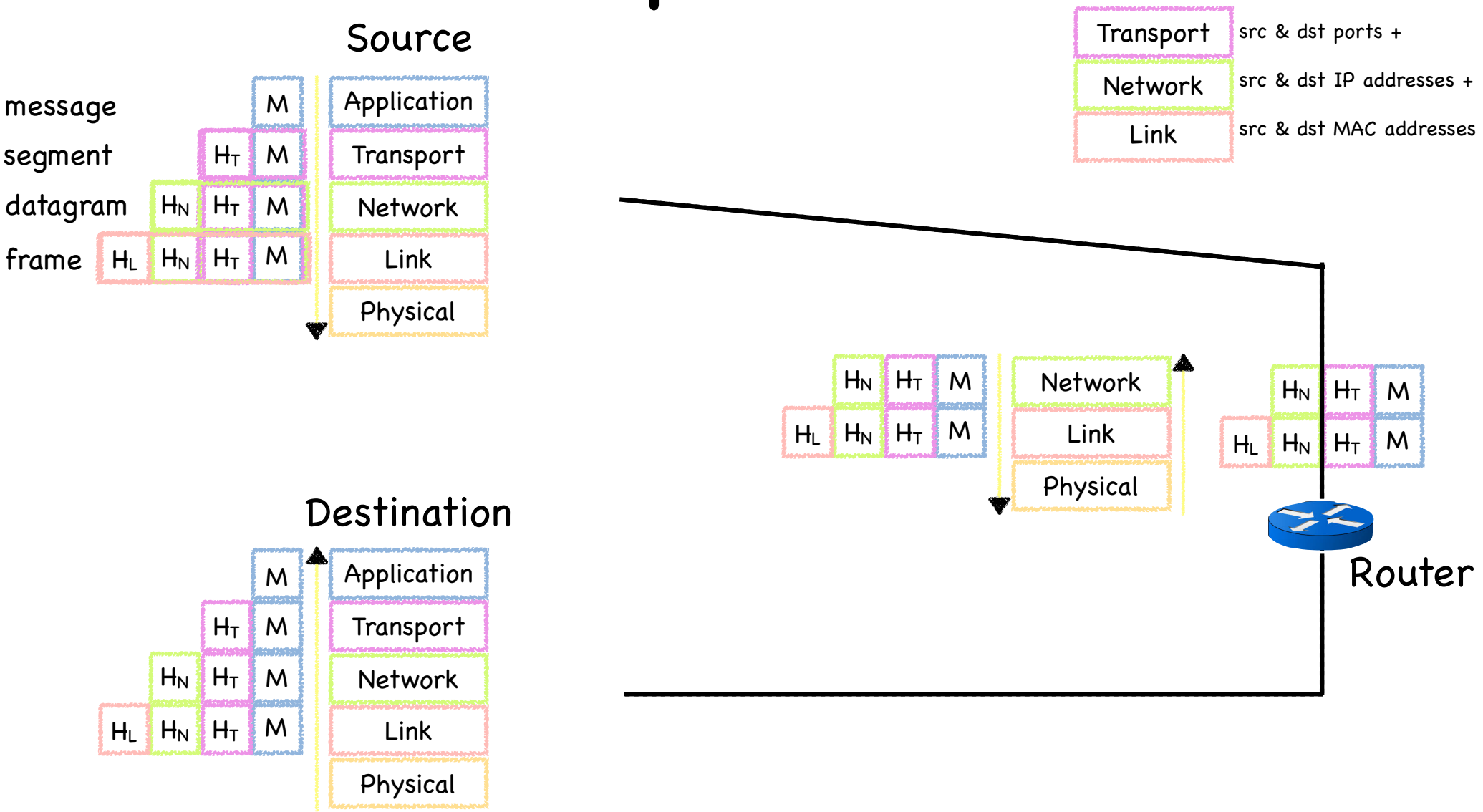
The Big Picture



The Big Picture



Encapsulation



End-to-End Argument

Saltzer, Reed & Clark, 1981

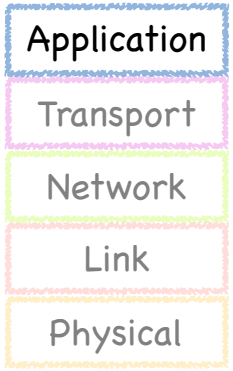
- If a function can be completely and correctly implemented only with the knowledge and help of the application standing at the endpoints of the communication system,
- then providing that function as a feature of the communication system itself is not possible
- Sometimes providing an incomplete version as a feature of the communication system itself may be useful as a performance enhancement

An Application of the Argument

- Should the network guarantee packet delivery?
  - Consider transferring a file
    - Sender reads file from disk & sends it; Receiver reads packets and writes them to disk
  - Wouldn't it be simpler if network guaranteed delivery?
  - No!
    - Application still needs to check file was written to disk
    - It needs to implement anyway its own retransmits

EtE Argument's Impact

- Occam's Razor for Internet architecture
  - End-to-end properties are best provided by applications, not by the network
    - Guaranteed packet delivery, ordered packet delivery, duplicate suppression, security, etc.
- Internet performs simplest packet routing and delivery service
  - Packets are sent on a best-effort basis



Application Layer

Application Layer Protocols

- HTTP

  persistent/non persistent; stateless, hence cookies

- Electronic Mail

  SMTP; POP3; IMAP

- DNS

  Manages naming of internet hosts

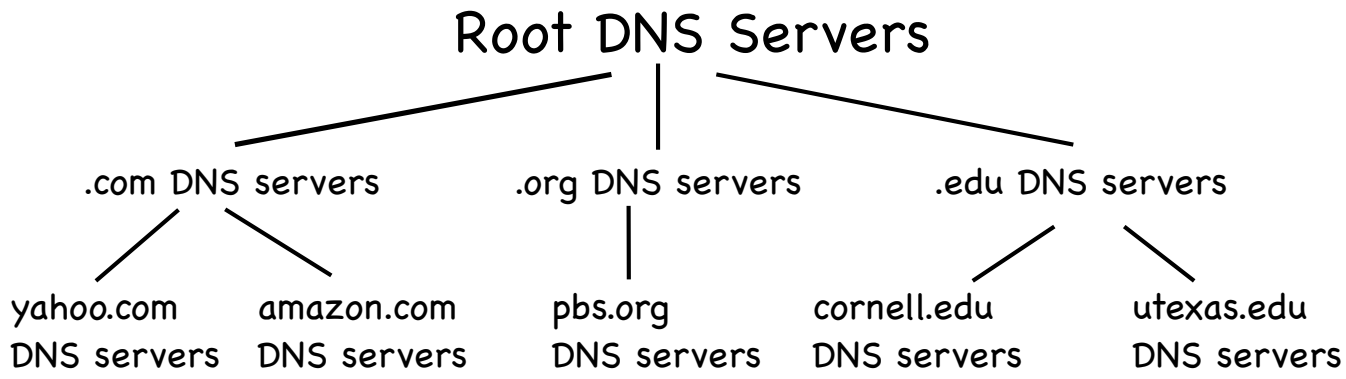
Naming

- People
  - SSN, NetId, Passport Number
- Internet Hosts, Routers
  - IP Address (32 bit) Ex: 128.84.96.12
    - Assigned to hosts by their internet service providers (ISPs)
    - Not a unique id, can be reused for a different machine
    - Determines how packets reach the address holder
  - A virtual "name" Ex: www.cs.cornell.edu
    - Human friendly

How are human friendly names translated into IP addresses?

Domain Name System (DNS)

Mockapetris '87

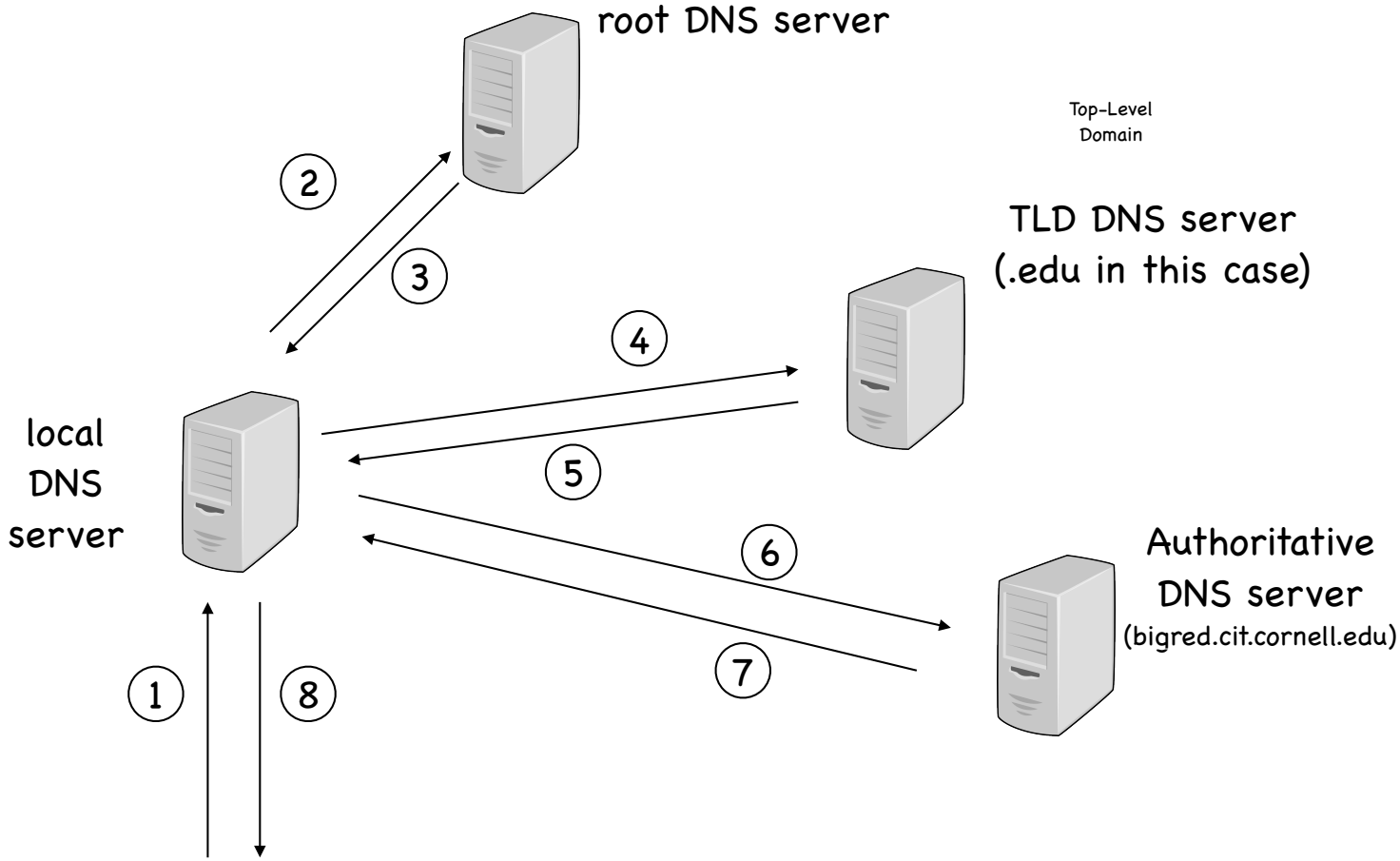


- A name service built above a hierarchical, distributed, autonomous, reliable database
  - Application level protocol: hosts & name servers communicate to resolve names
  - Introduced to replace the original Internet naming scheme
    - a single central master file downloaded everywhere by FTP
  - Components separated by dot and resolved from right to left
  - All names are global: they mean the same everywhere in the DNS

DNS Lookup

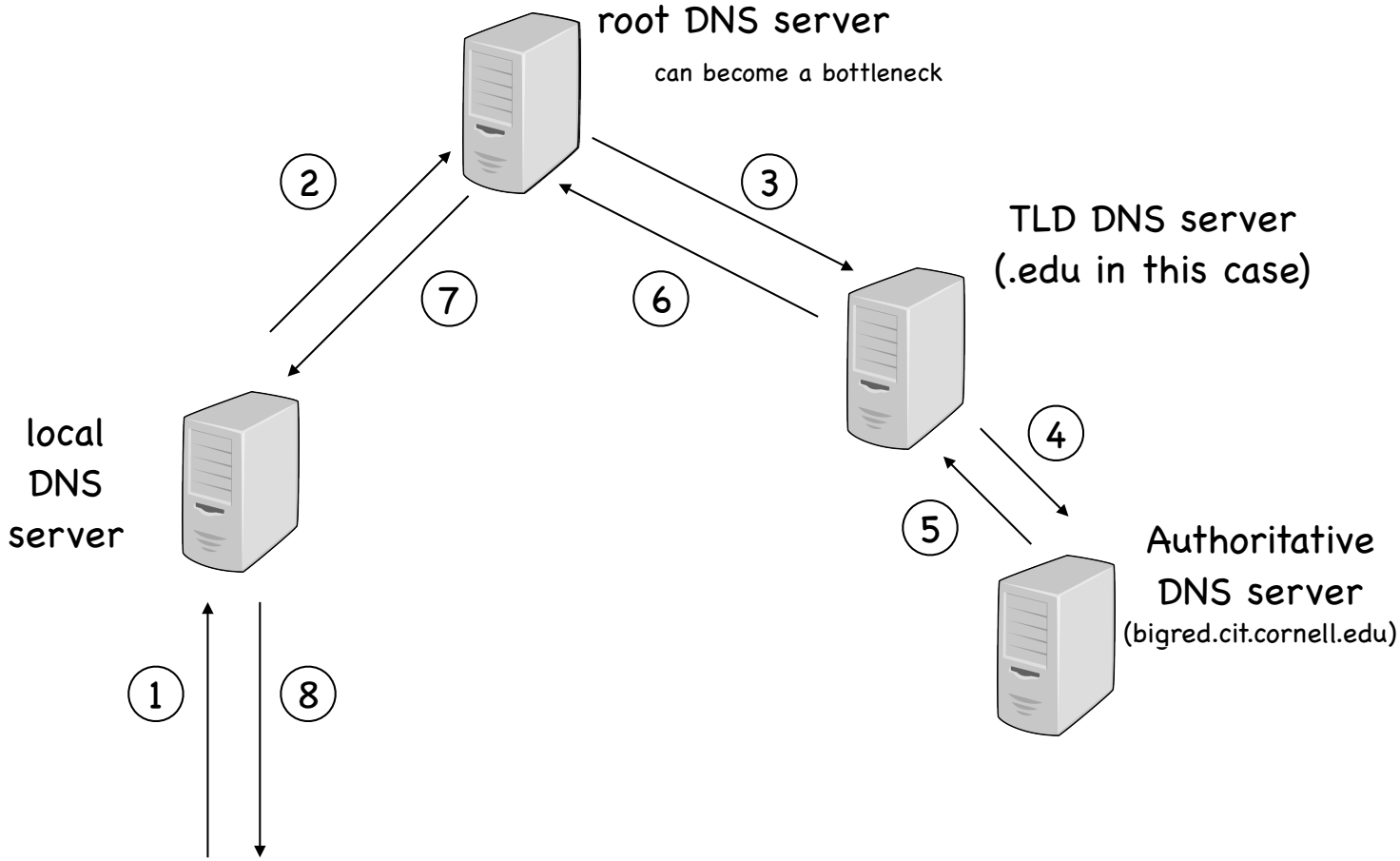
- To find the IP address of www.cs.cornell.edu (basic protocol)
  - Query root server to find IP of DNS server for .edu...
  - which, when queried next, will provide IP of DNS server for cornell.edu
  - which, when queried next, will provide IP of cs.cornell.edu
  - which, when queried next, will provide IP of www.cs.cornell.edu

DNS Name Resolution: Iterative



www.cs.cornell.edu

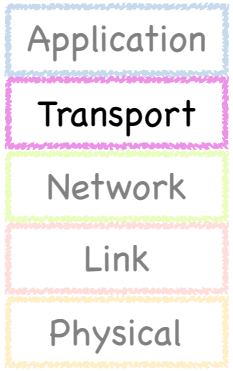
DNS Name Resolution: Recursive



www.cs.cornell.edu

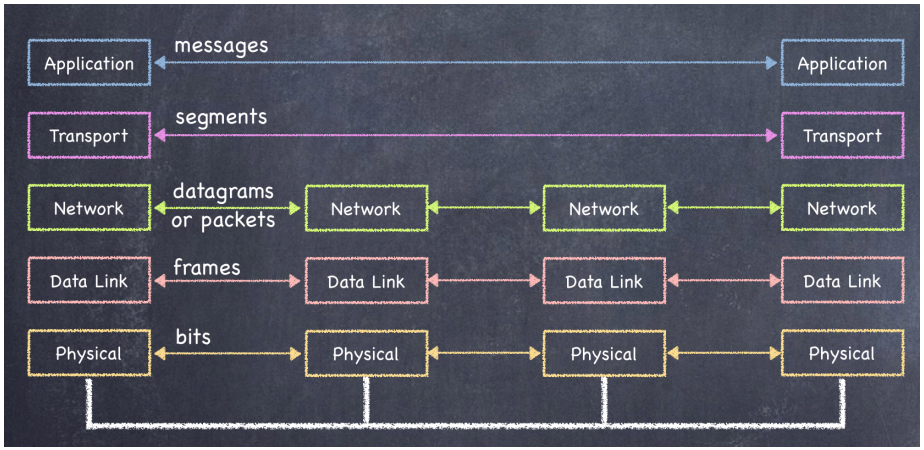
Caching

- Cache entries may be out of date
  - bindings are forgotten after TTL
  - TLD servers typically cached in local name servers
- Servers cache new bindings they learn
  - only eventual consistency
  - if binding changes, it may not be learned until TTL
  - update notify proposed IETF standard



Transport Layer: UDP & TCP

Transport Services and Protocols



- Provide logical communication between processes on different hosts
  - logical communication between hosts is left to the network layer
- Sender packages messages into segments, passes them to the network layer
- Receiver reassembles segments into messages, passes them to the application layer
- Apps can use multiple protocols (e.g., on the Internet, UDP or TCP)

Internet Transport-layer Protocols

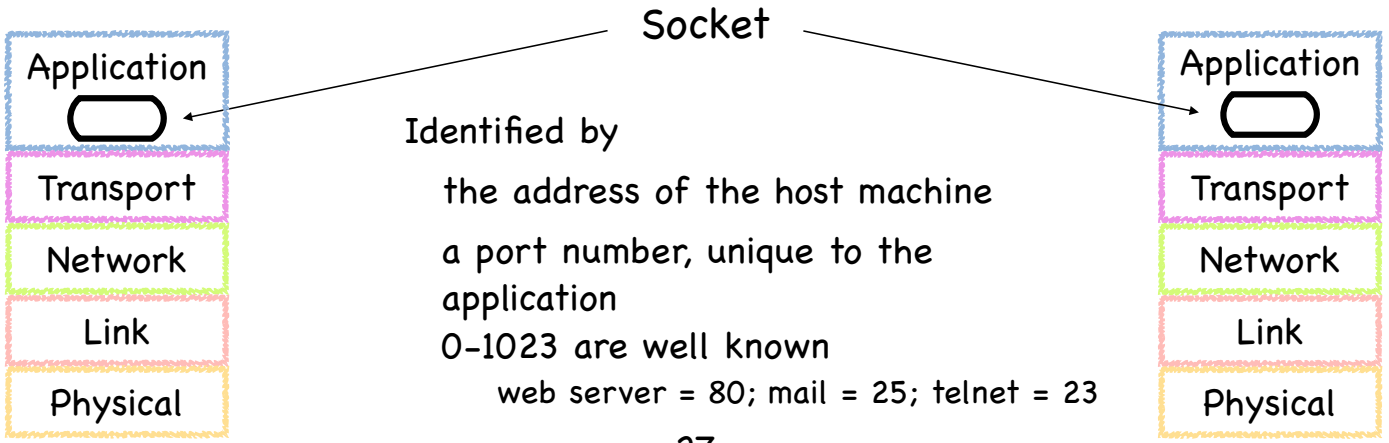
- TCP (Transmission Control Protocol)
  - Reliable, in-order delivery
  - Session-based/connection set up
  - byte-stream abstraction
    - Congestion control
    - Flow control
- UDP (User Datagram Protocol)
  - Unreliable, unordered delivery
  - No connection setup
    - no-frill extension of best-effort IP (network layer protocol)
- Services not available:
  - delay guarantees
  - bandwidth guarantees

Applications and their Transport Protocols

Application	Application-Layer Protocol	Transport Protocol
Email	SMTP	TCP
Remote terminal access	Telnet	TCP
Web	HTTP	TCP
File Transfer	FTP	TCP
Remote File Server	NFS	Typically UDP
Streaming Multimedia	Proprietary	UDP or TCP
Internet Telephony	Prioprietary	UDP or TCP
Network Management	SNMP	Typically UDP
Routing Protocol	RIP	Typically UDP
Name Translation	DNS	Typically UDP

Socket

- One endpoint of a two way communication between two application processes running on a network
  - Door between process and end-to-end transport protocol
  - Sending process pushes messages out the socket to the transport protocol
  - Transport protocol delivers message to the socket at the receiving process



The Big Picture (Sender's Edition)

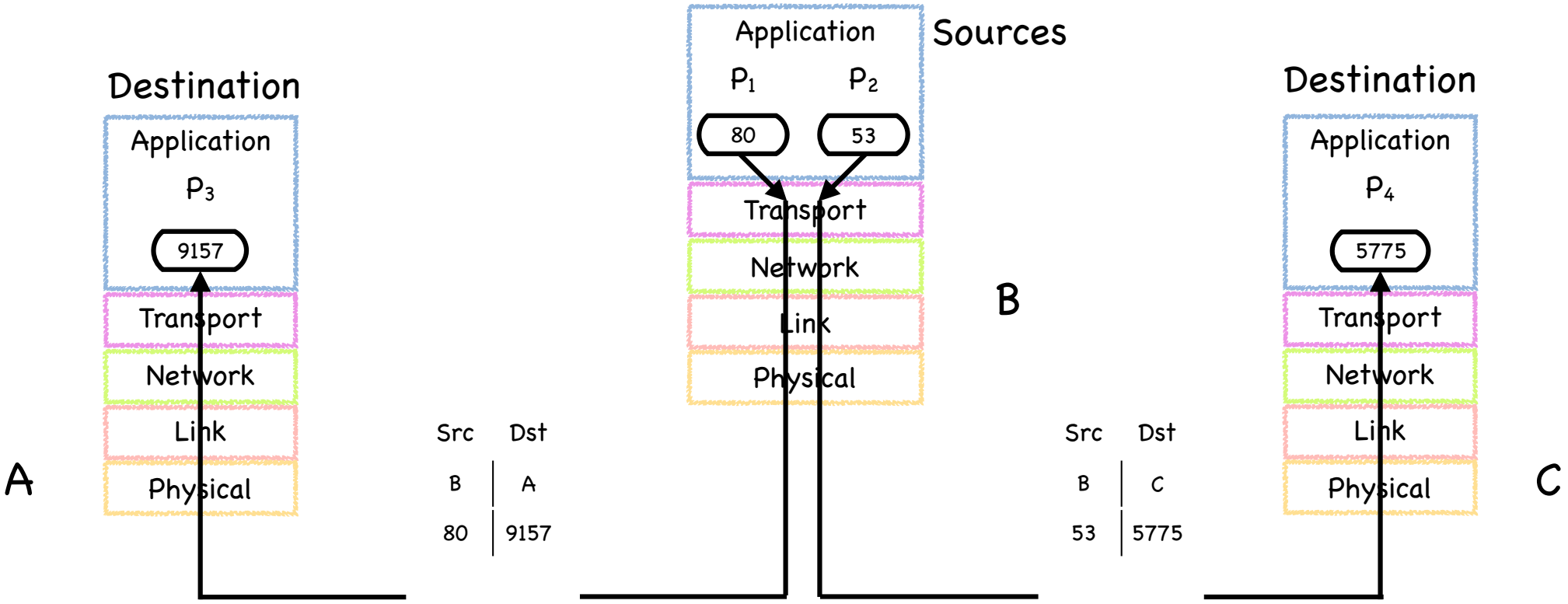
- Sending application
  - specifies IP address (to identify host) and port of destination
  - uses socket bound to a source port
- Transport layer
  - breaks application message into smaller chunks
  - add to each transport-layer header
- Network layer
  - adds network layer header (with IP address)

The Big Picture (Receiver's Edition)

- Network layer
  - removes network layer header (with IP address)
- Transport layer
  - removes from each segment transport-layer header
  - reassembles application message from segment
- Receiving application
  - receives message on destination port bound to socket

Multiplexing at the Sender

- Handles data from multiple sockets
- Adds transport header (later used for demultiplexing)



Demultiplexing at the Receiver

- Handles data from multiple sockets
- Adds transport header (later used for demultiplexing)