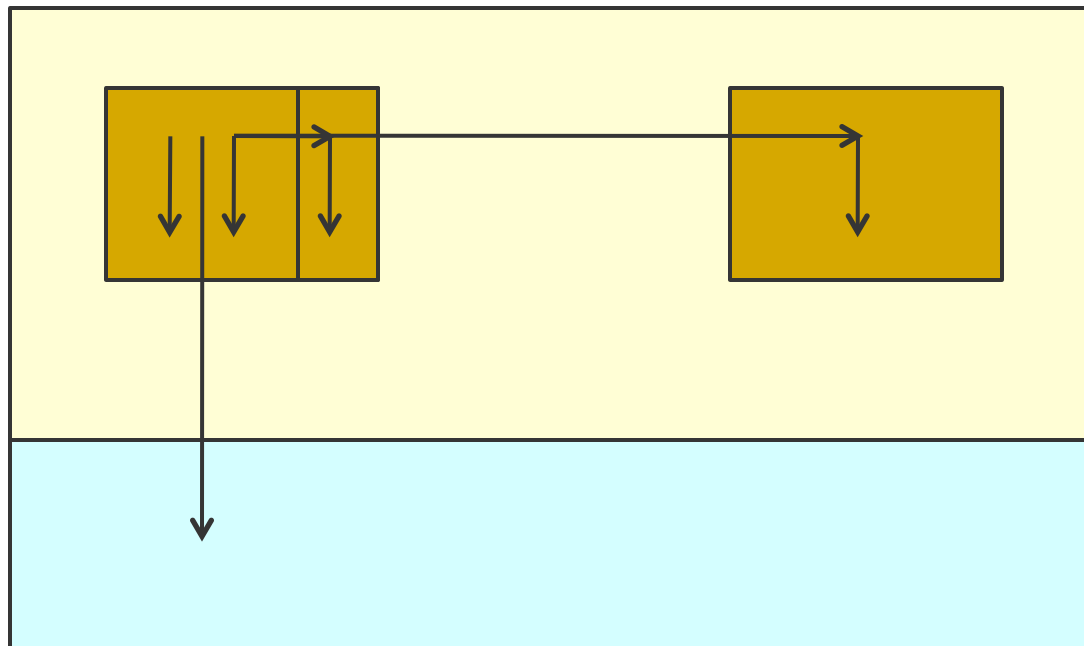




# Introduction to Networking and the Internet

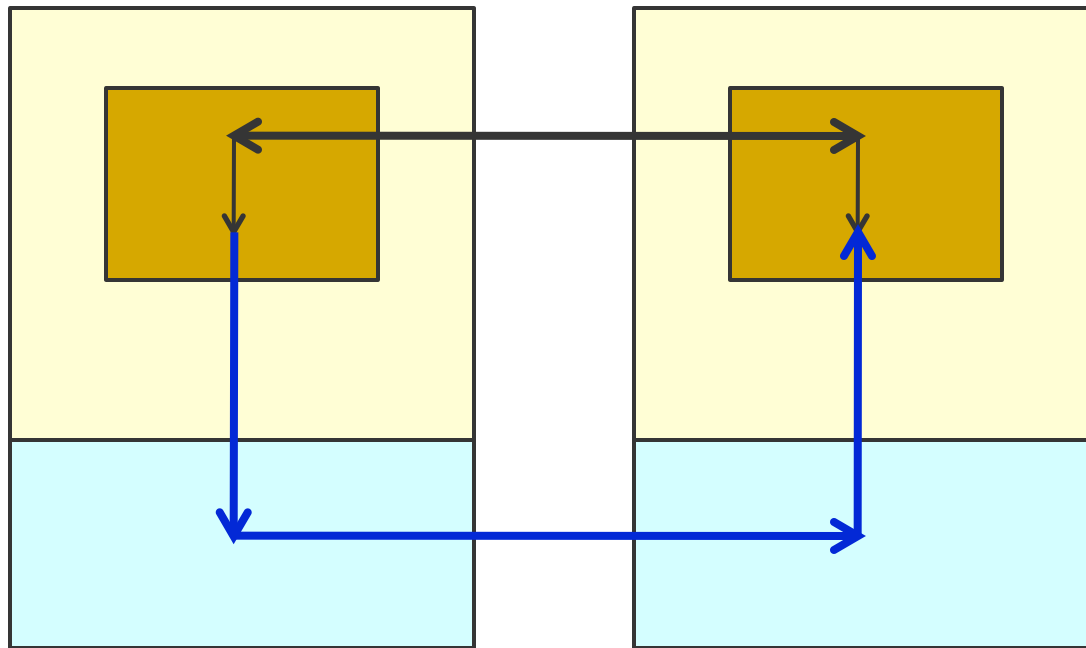
# [ Where are we? ]

- Function calls, system calls, threads and processes



# [ What's next? ]

- Networked communication and distributed applications



# [ Introduction ]

---

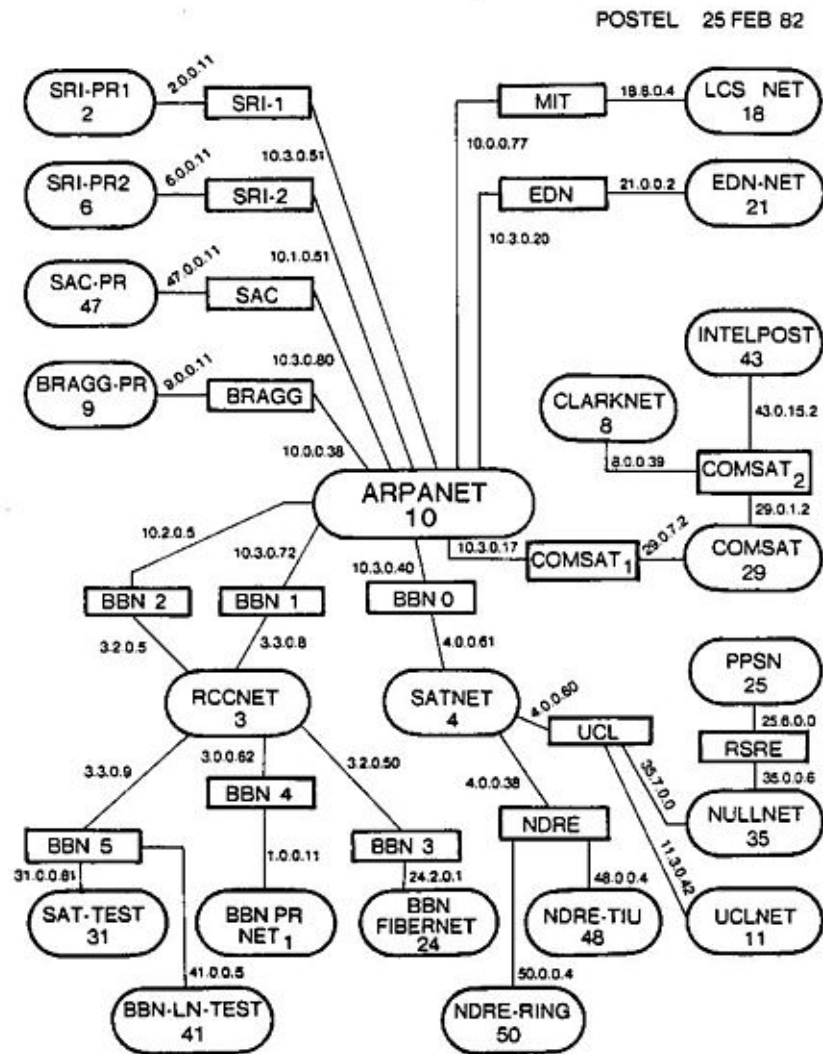
- What is the Internet?
- Network architecture
- What is a protocol?
- Protocol layers, service models



# The Internet in 1982

- Map of the Internet
- Created manually!
- Ovals: sites/networks
- Rectangles: Routers
- Individual hosts not shown

Image by Jon Postel, ISI



# [ The Internet in 2005 ]

- Less than 30% of one class of networks (class C)
- Complex research project!
- Length of each line indicates delay between two IP hosts.
- See color coding at `wikipedia:Internet`

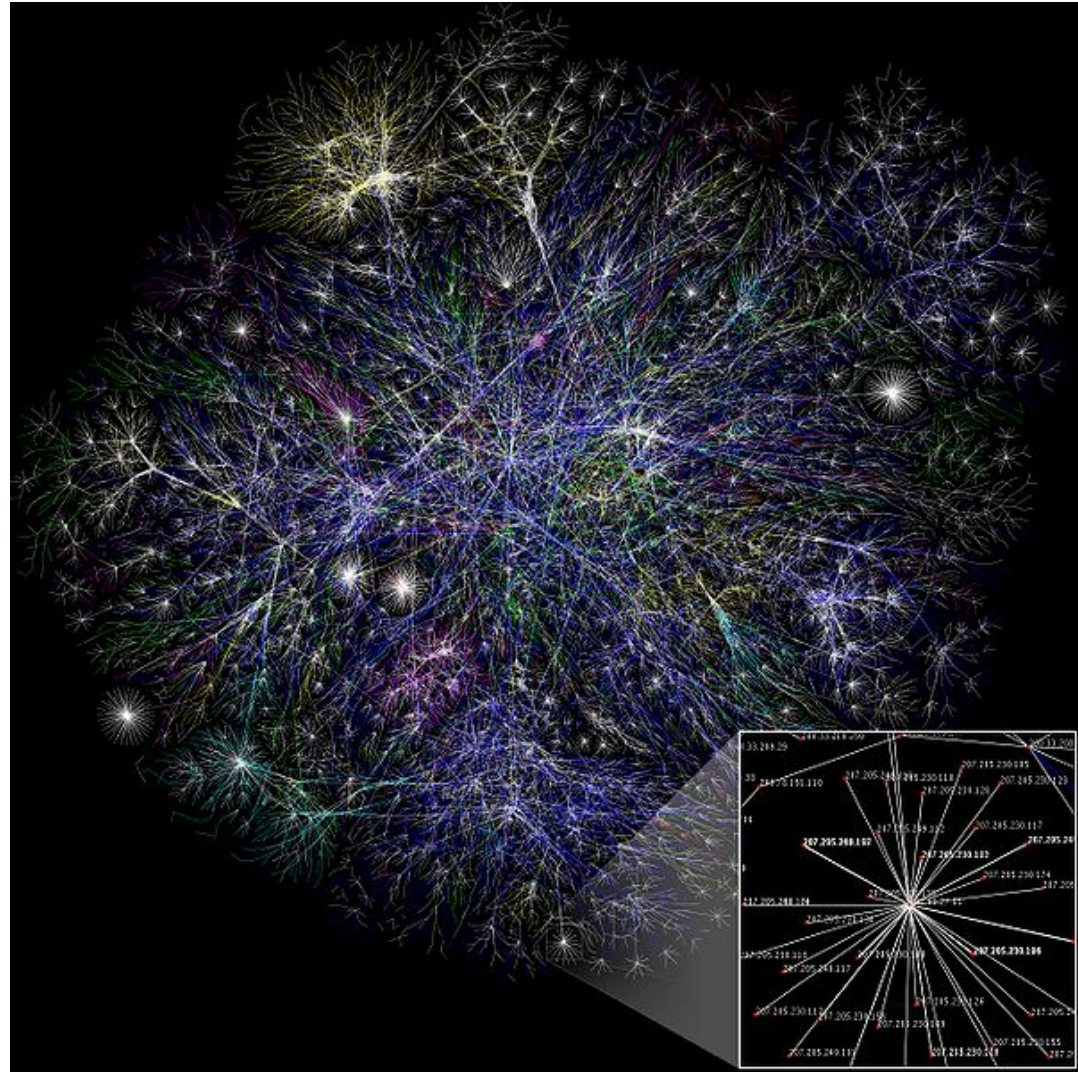


Image by Barrett Lyon,  
Opte Project, [opte.org](http://opte.org)



# [ The Reach of the Internet ]

- INFOTHELA
- Wireless networking
- Moves from village to village

CPU

Display

Battery



*Media Lab Asia, IIT Kanpur*



# [ Entrenched Computer Technologies ]

- DRAM memory?
- C, C++ programming languages?
- X86, ARM, MIPS processors
- Microsoft Word, Microsoft Excel
- ...
- TCP/IP (Foundation of the modern Internet)





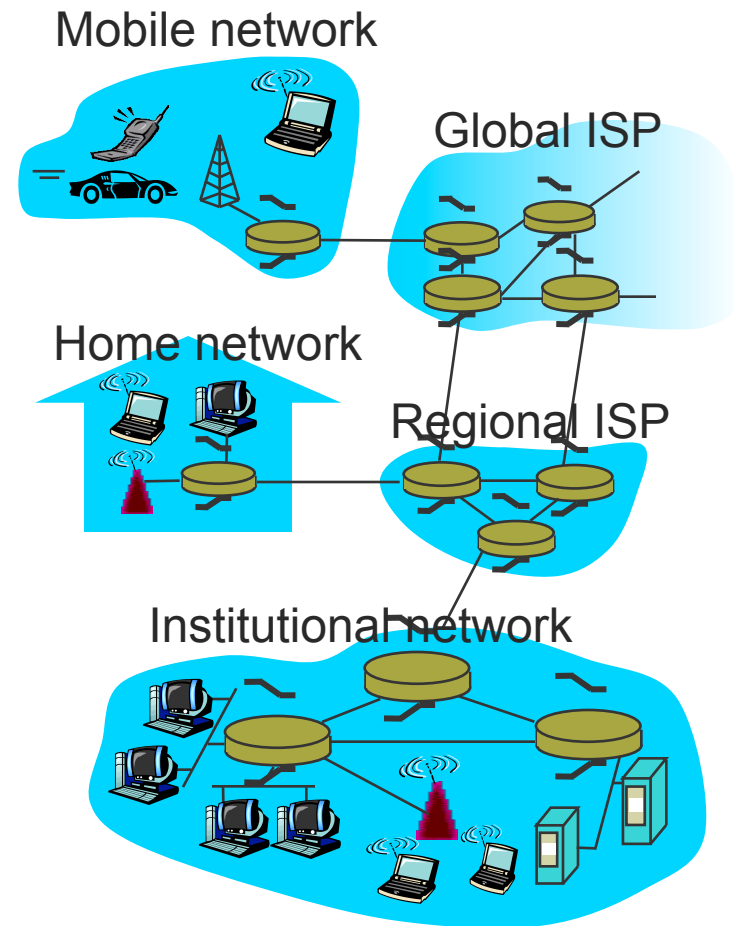
# Internet Standards

- Internet Engineering Task Force (IETF)
  - IPv4: Internet Protocol v4 (1981):  $2^{32}$  IP addresses
  - IPv6: Internet Protocol v6 (1998):  $2^{128}$  IP addresses
- Internet Corporation for Assigned Names and Numbers (ICANN)
  - IP Addresses, e.g., 128.174.245.12
  - Supervises authority for domain names, e.g.,  
`cnn.com`, `sourceforge.org`, `uiuc.edu`



# What is the Internet?

- Communication infrastructure
  - Enables distributed applications
  - Web, VoIP, email, games, e-commerce, file sharing
- Communication services
  - Provided to applications
  - Reliable data delivery from source to destination
  - “best effort” (unreliable) data delivery

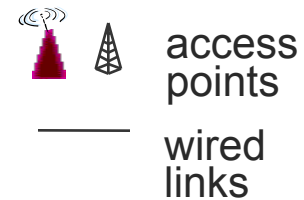


# [ Direct Connectivity ]

- Building Blocks

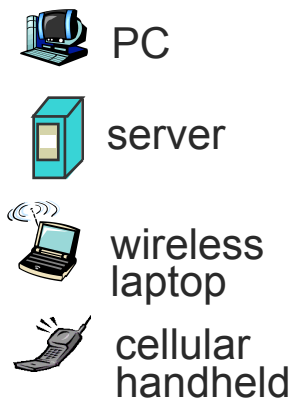
- Links

- coax cable, optical fiber, WiFi, ...



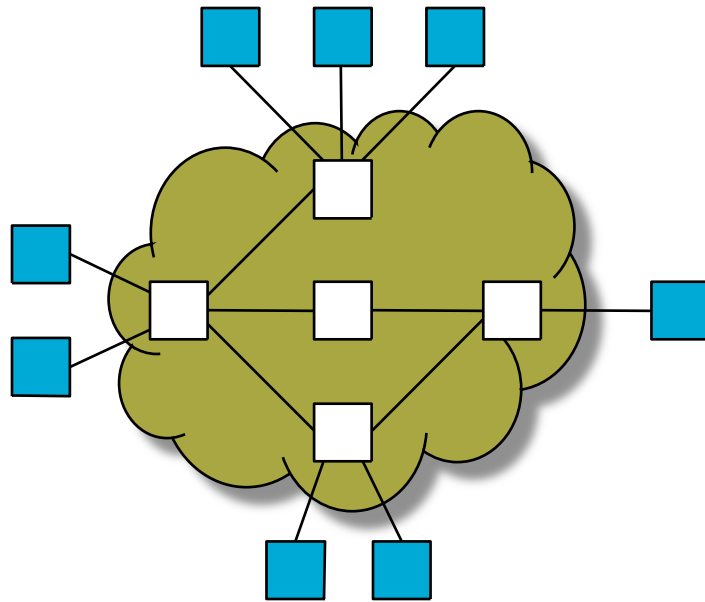
- Nodes

- workstations, routers, ...

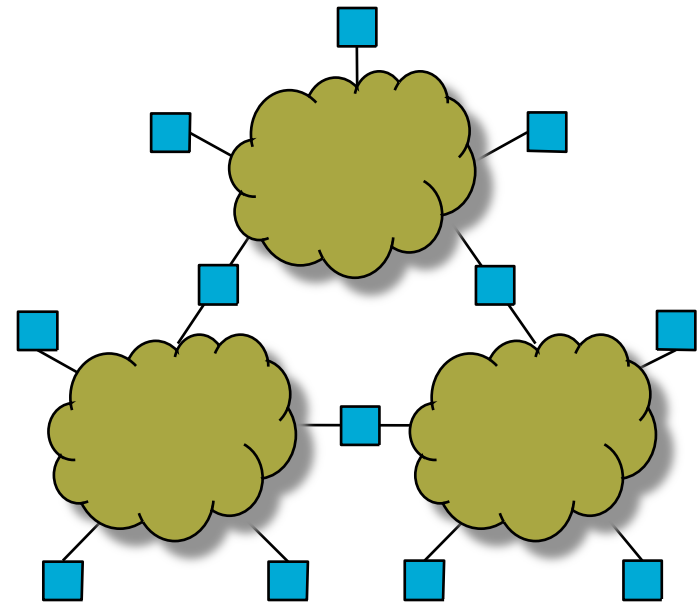


# Indirect Connectivity

- Switched Networks

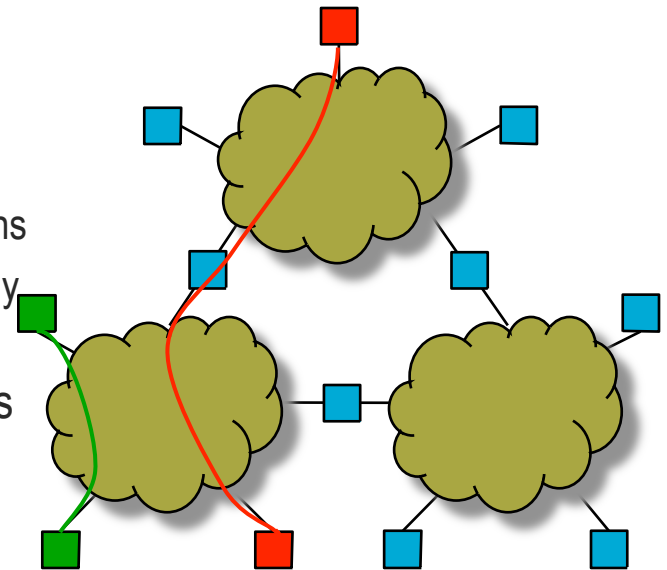


- **Internet**works



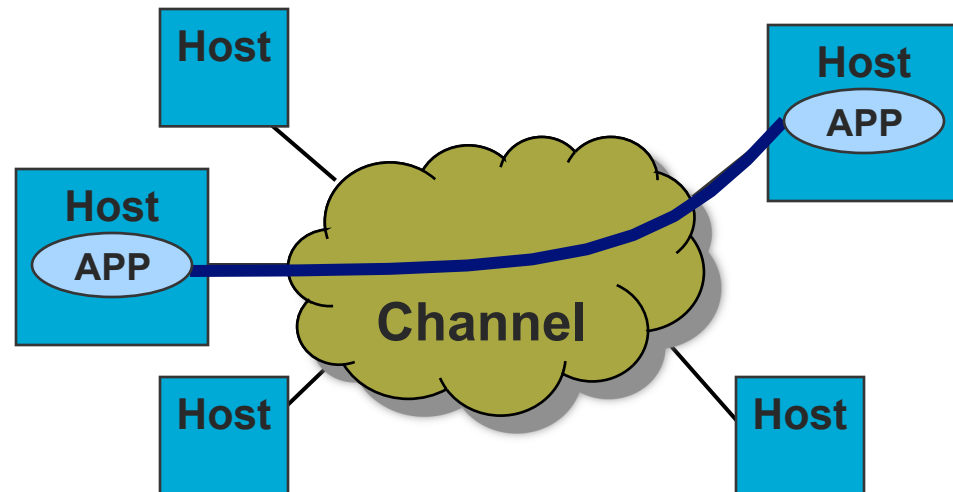
# [ Network Service ]

- Goal
  - Transfer data between end systems
- Support For Common Services
  - Idea
    - Common services simplify the role of applications
    - Hide the complexity of the network without overly constraining the application designer
  - Semantics and interface depend on applications
    - Request/reply: FTP, HTTP
    - Message stream: audio, video



# [ Channels ]

- Channel
  - The abstraction for application-level communication
- Idea
  - Turn host-to-host connectivity into process-to-process communication

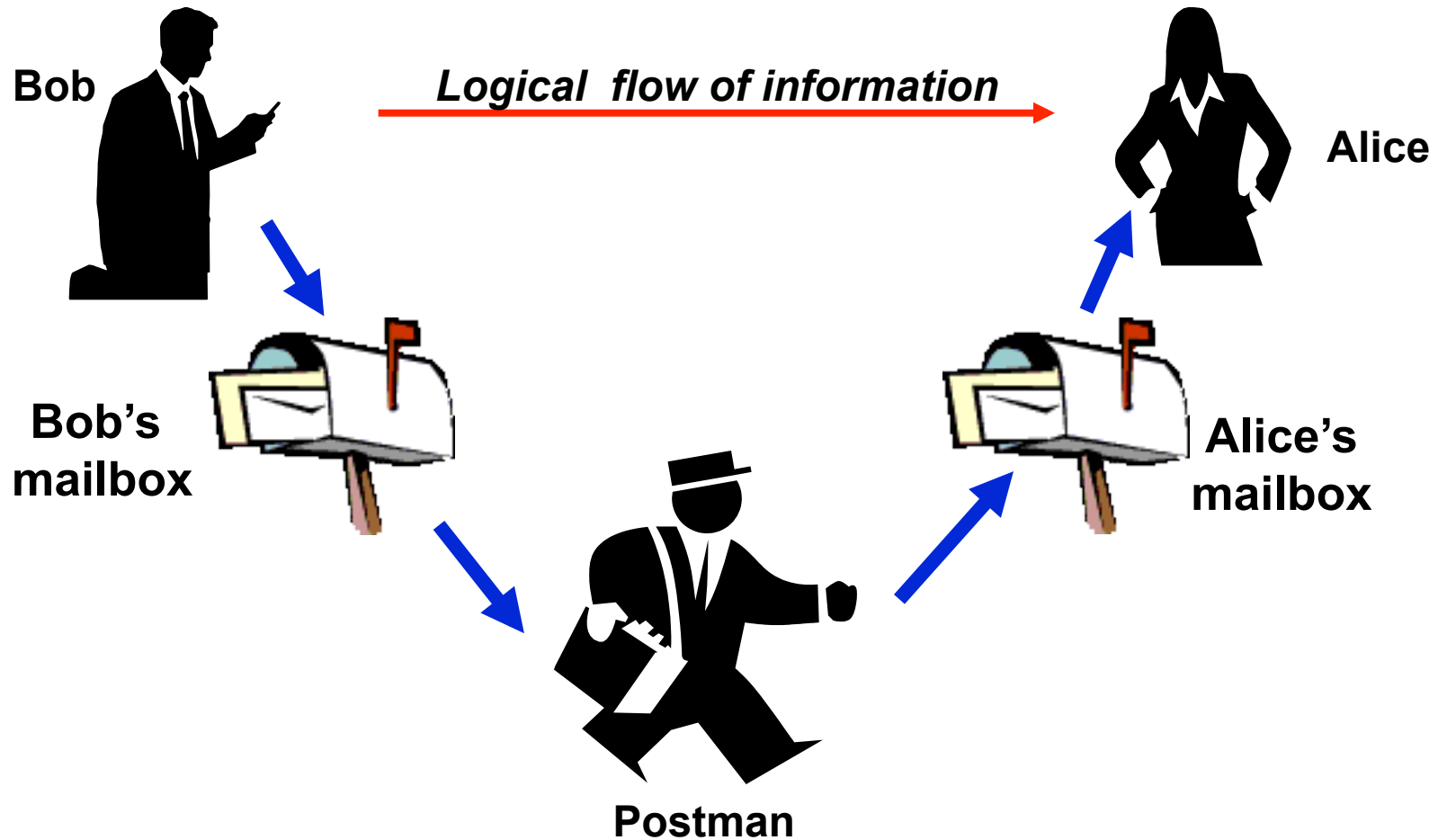


# [ Inter-process Communication ]

- Problems typically masked by communication channel abstractions
  - Bit errors (electrical interference)
  - Packet errors (congestion)
  - Link/node failures
  - Message delays
  - Out-of-order delivery
  - Eavesdropping
- Goal
  - Fill the gap between what applications expect and what the underlying technology provides



# [ Example: Sending a Letter ]





# [ Services ]

## ■ Unconfirmed service



## ■ Acknowledged service



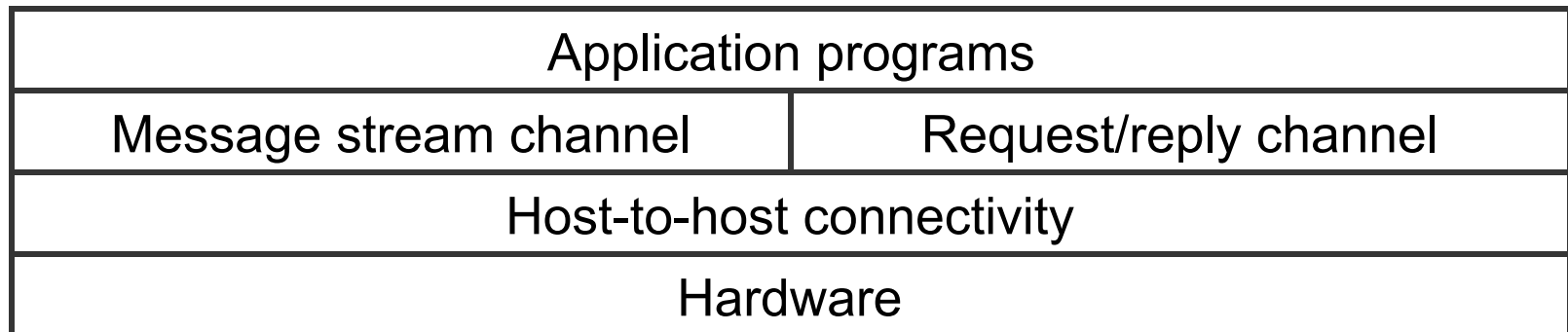
# [ Network Architecture ]

- Networks are complex!
- Many “pieces”
  - Hosts
  - Routers
  - Links of various media
  - Applications
  - Protocols
- Question
  - Is there any hope of organizing structure of network?



# [ Abstraction through Layering ]

- Abstract system into layers:
  - Decompose network design into manageable components
  - Each layer provides well-defined functionality



# [ Why layering? ]

- Reduce Complexity
  - Strict interfaces limit interactions between system components
  - Each layer must know only about 2 other layers!
- Modularity
  - Eases maintenance, updating of system
    - Implementation changes in a layer are transparent to rest of system
- Protocol
  - Instantiation of a layer!



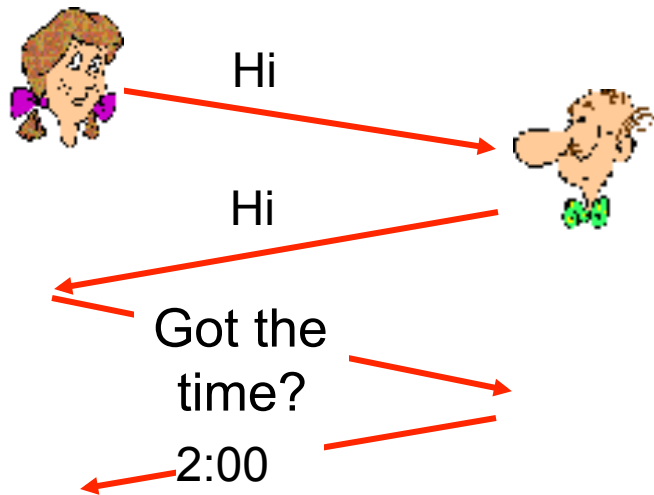
# [ What is a Protocol? ]

- Protocols are defined by
  - Specific msgs sent
  - Specific actions taken when msgs received, or other events
- Human protocols
  - “what’s the time?”
  - “I have a question”
  - Introductions
- Protocols define
  - Format
  - Order of msgs sent and received among network entities
  - Actions taken on msg transmission, receipt
- Network protocols
  - Machines rather than humans
  - All communication activity in Internet is governed by protocols

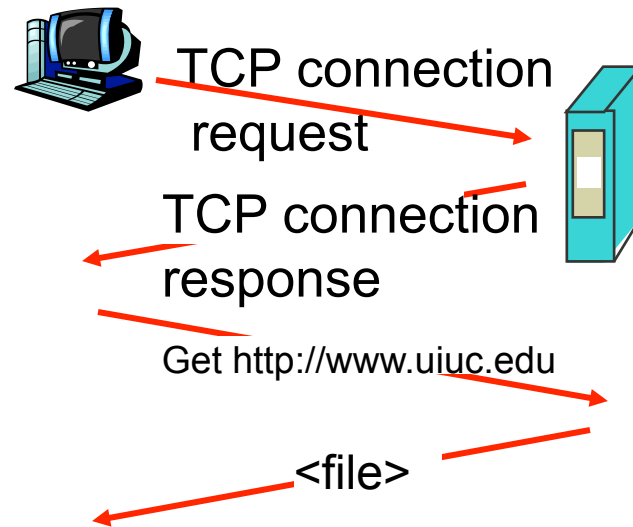


# What is a Protocol?

- A human protocol



- A computer network protocol

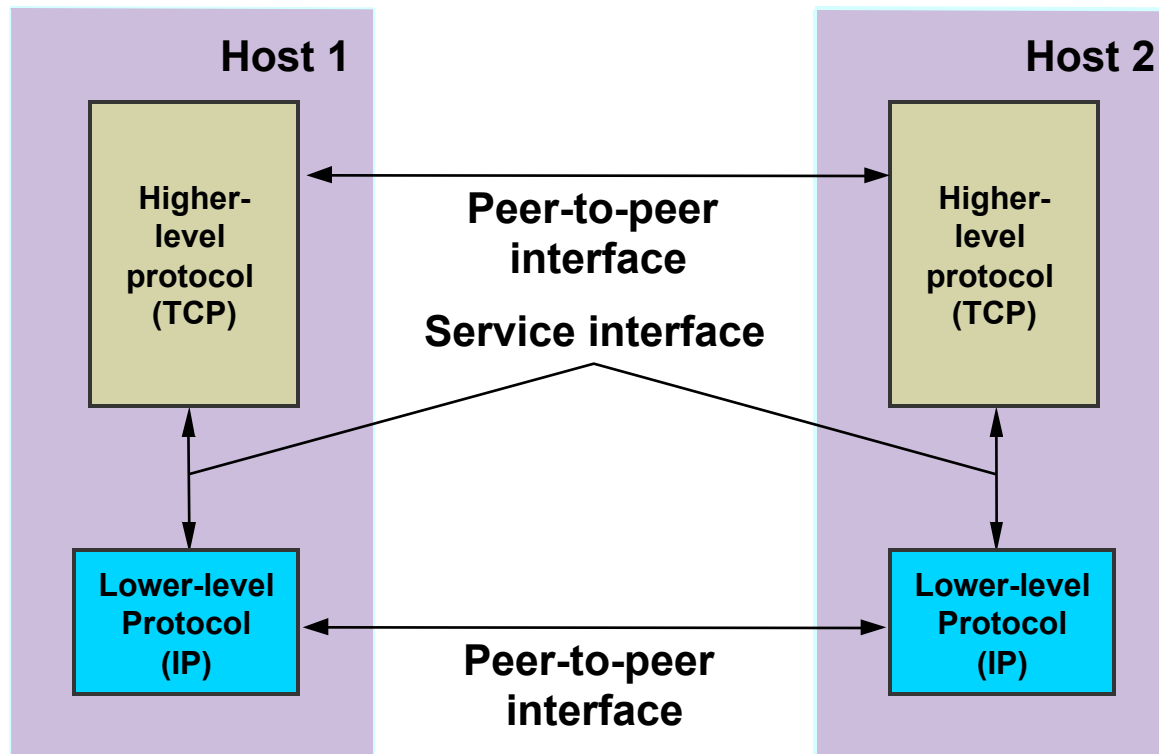


# [ Network Protocols ]

- Definition of a **Network Protocol**:
  - Abstract object forming a layer of a network system
  - Organized as a set of (a) messages; (b) action rules
- Goal: Provide a communication service that higher-layer objects use to exchange messages
  - Service interface: To objects on the same computer
  - Peer interface: To counterpart on a different machine



# [ Interfaces ]





# [ Layering Concepts ]

- Encapsulation

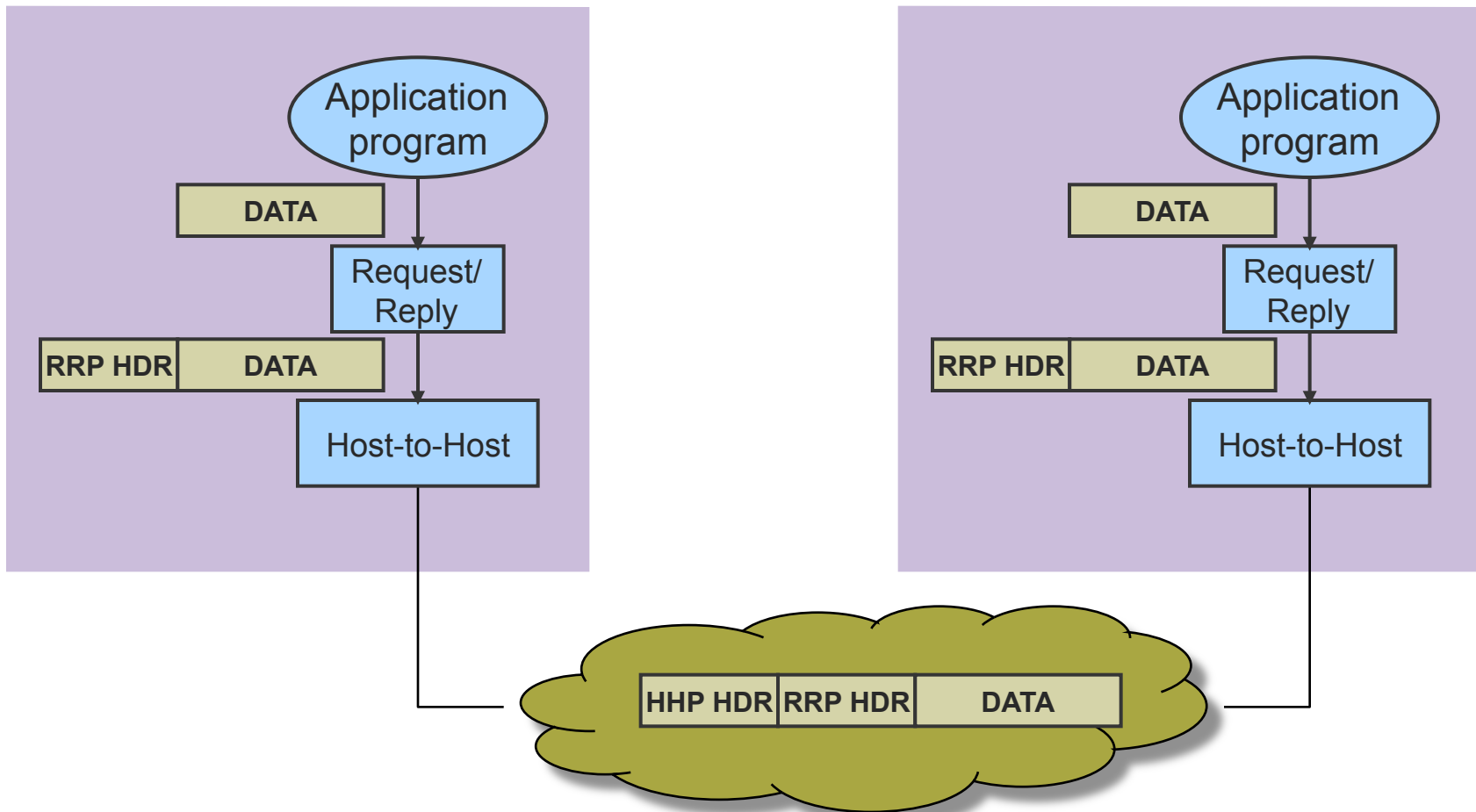
- Higher layer protocols create messages and send them via the lower layer protocols
- These messages are treated as data by the lower-level protocol
- Higher-layer protocol adds its own control information in the form of headers or trailers

- Multiplexing and Demultiplexing

- Use protocol keys in the header to determine correct upper-layer protocol

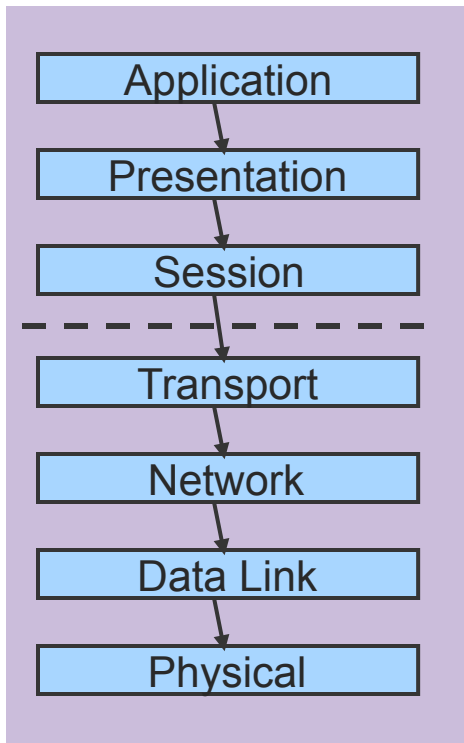


# [ Encapsulation ]



# OSI Protocol Stack

**OSI = Open Systems Interconnection**  
Industry-driven standards body chartered by ISO

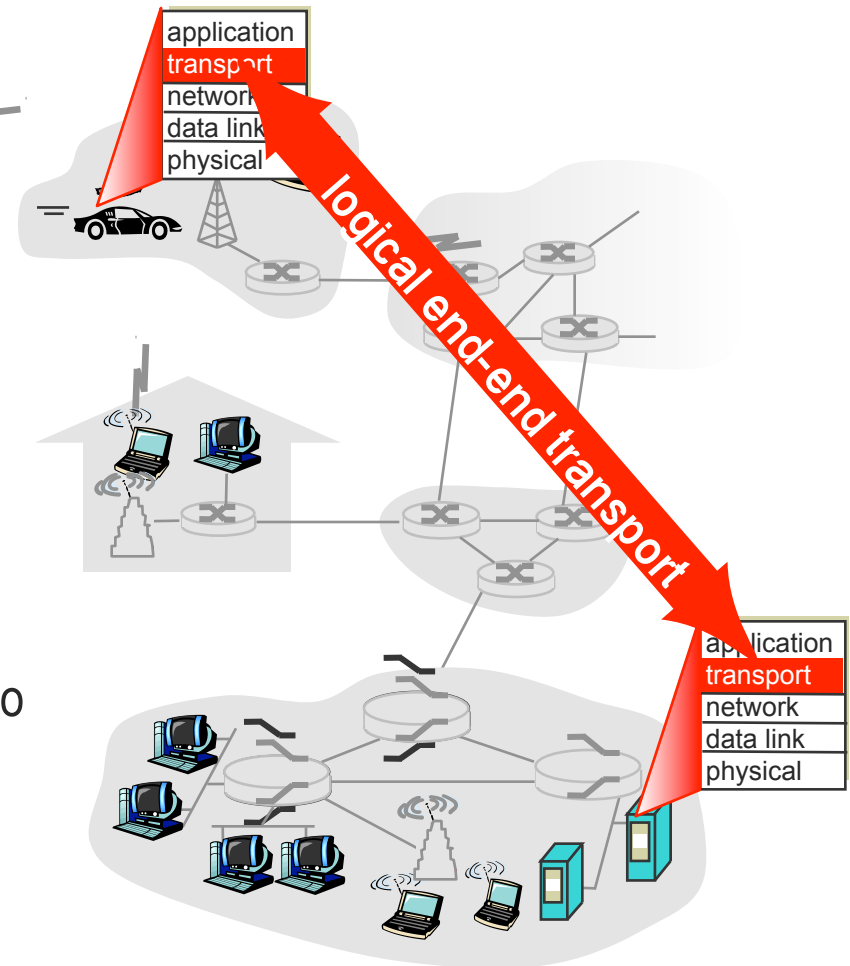


- **Application:** Application specific protocols SMTP, FTP, HTTP, ...
- **Presentation:** Format of exchanged data
- **Session:** Name space for connection mgmt
- **Transport:** Process-to-process channel TCP, UDP, ...
- **Network:** Host-to-host packet delivery IPv4, IPv6, ...
- **Data Link:** Framing of data bits Ethernet, ISDN, ...
- **Physical:** Transmission of raw bits Coax, WiFi, ...



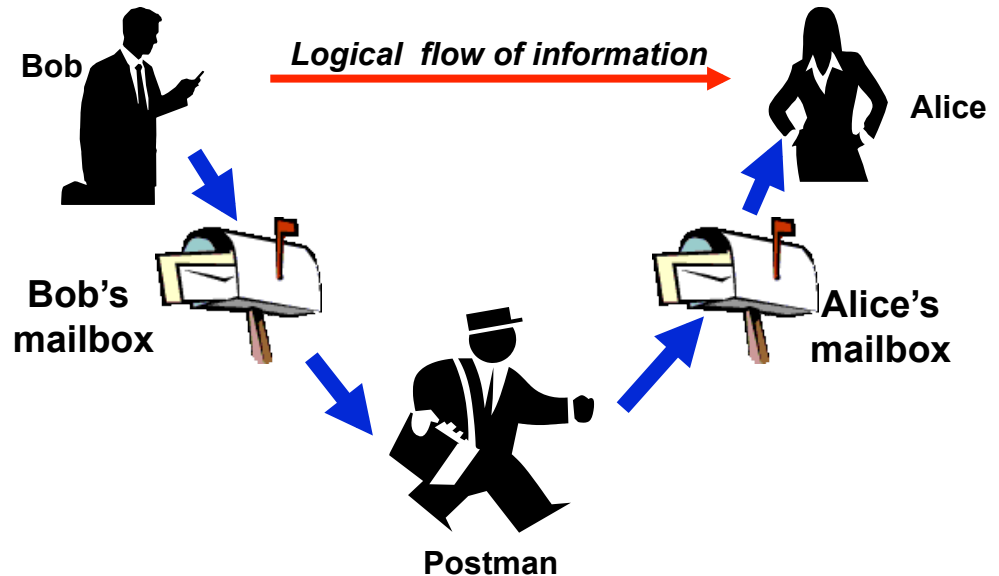
# Example: Transport Layer

- Provide logical communication between application processes running on different hosts
- Transport protocols run in end systems
  - Send side:
    - Break application messages into segments
    - Pass to network layer
  - Receive side:
    - Reassemble segments into messages
    - Pass to application layer
- More than one transport protocol available to applications
  - Internet: TCP and UDP

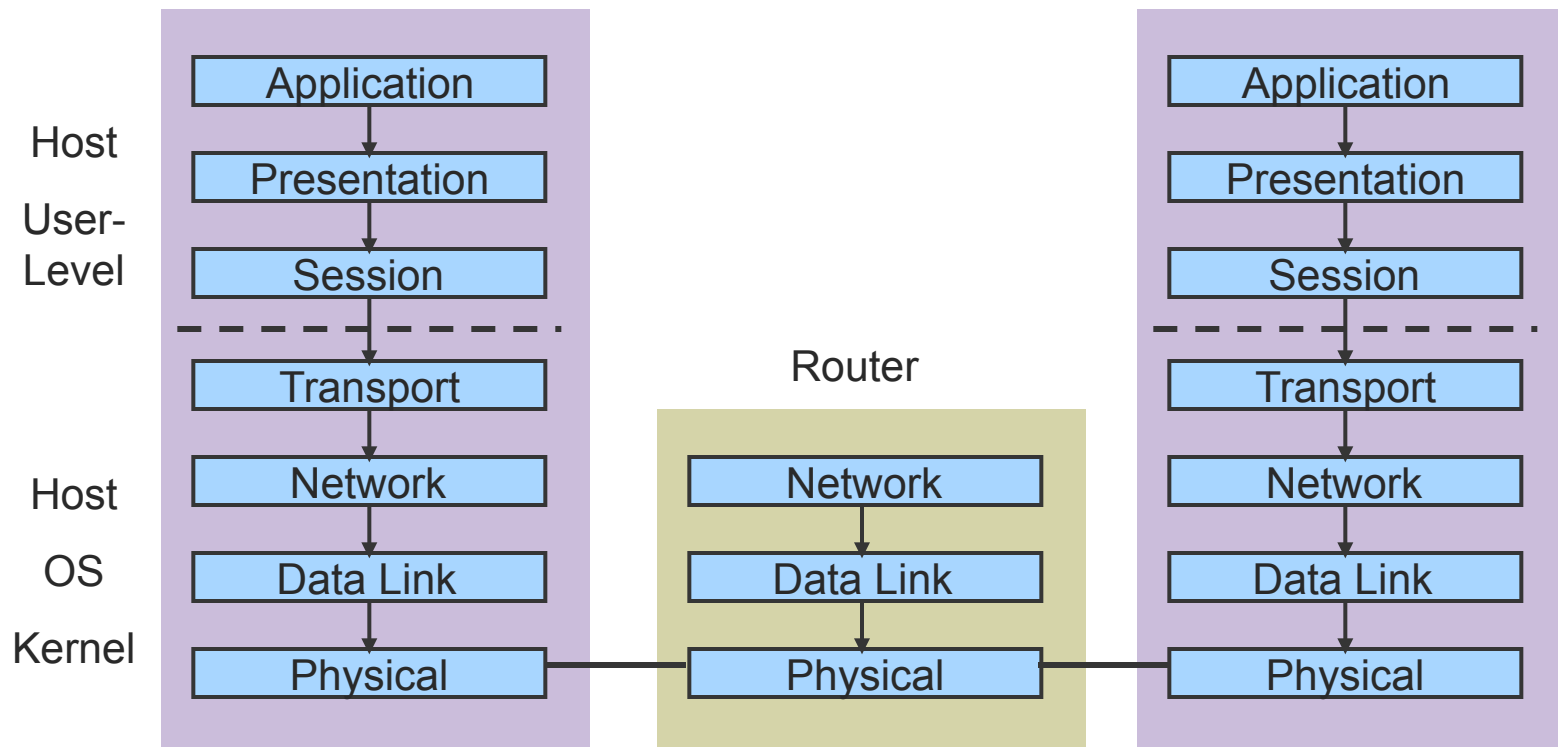


# Transport vs. Network Layer

- Transport layer
  - Logical communication between processes
  - Relies on, enhances, network layer services
- Network layer
  - Logical communication between hosts

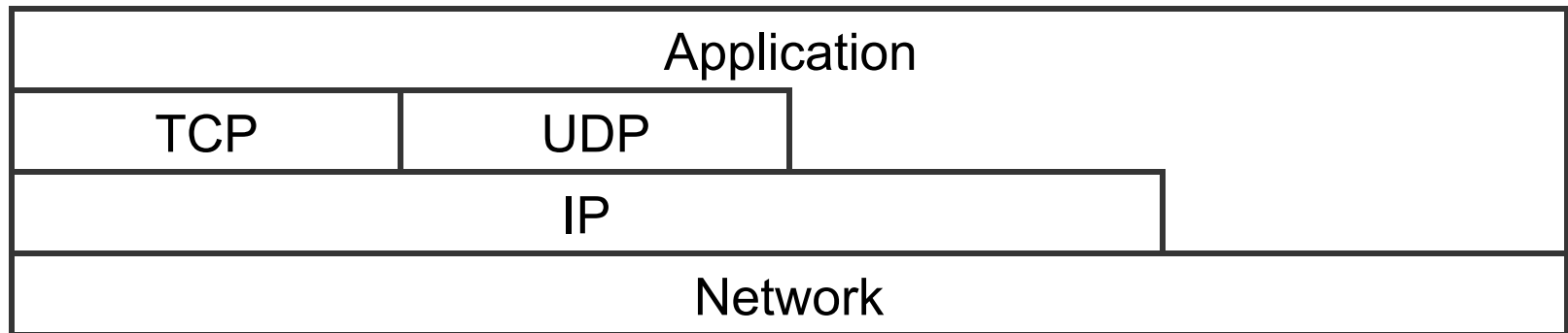


# [ OSI Protocol Stack ]



# [ Internet Architecture ]

- Features
  - No strict layering



# Internet Architecture – Hourglass Design

- Features

- Hourglass shape – IP is the focal point

