Lancaster University

Protocol Layers & Encapsulation

SCC. 203 – Computer Networks

Geoff Coulson

Week 12 Lecture 2

What we discussed so far about the structure of the Internet

Lancaster University

- The Internet is a network of heterogenous networks
  - Hosts (Servers, Laptops, Smartphones, TVs)
  - Network Devices (Routers, Switches)
  - Access technologies (Ethernet, WiFi, 4G, LTE)
  - Physical Media (Coaxial cable, twisted pair, optical fiber)
- The Internet supports diverse applications
  - Email, WWW, File Sharing, Streaming, …

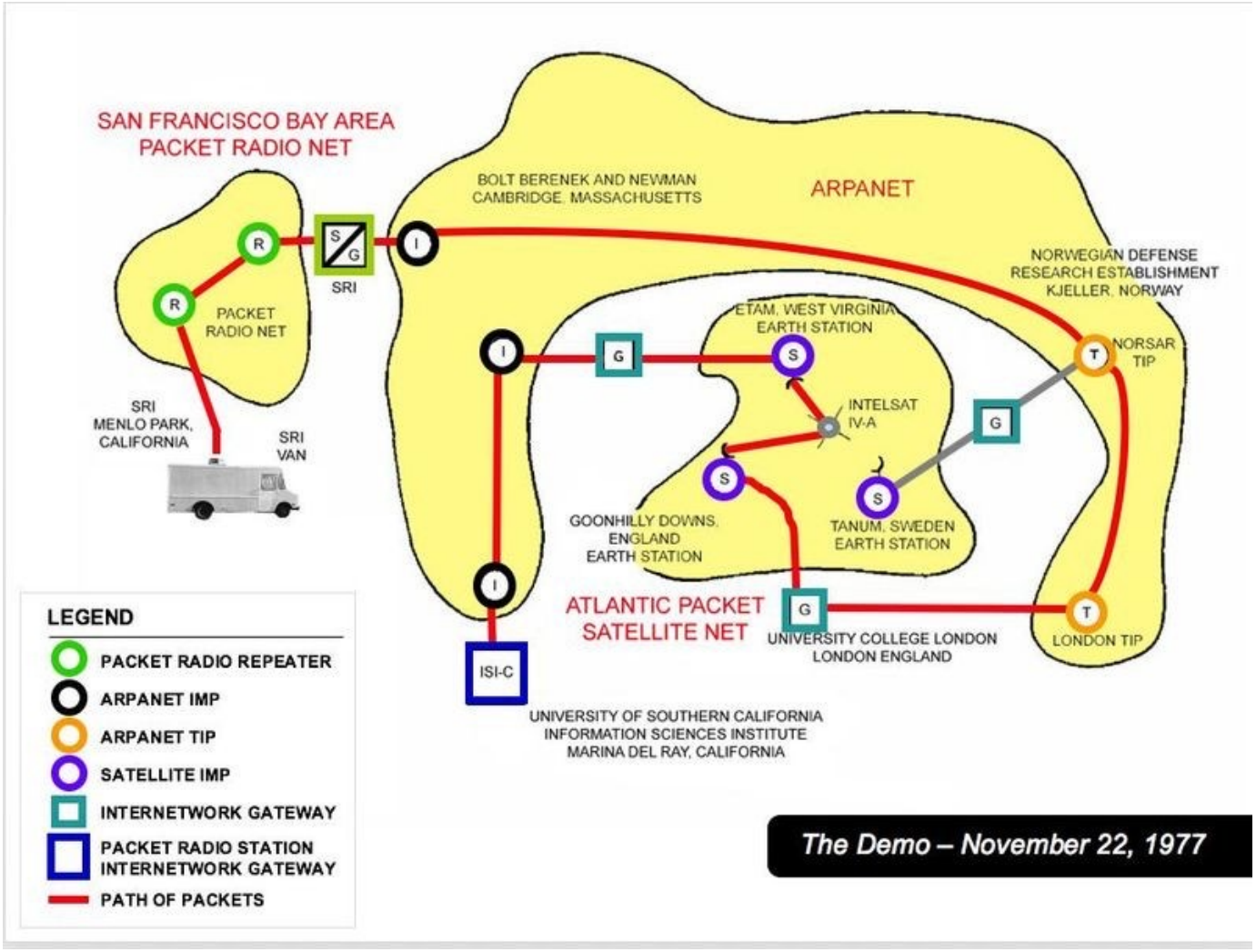
What we will discuss today

Lancaster University

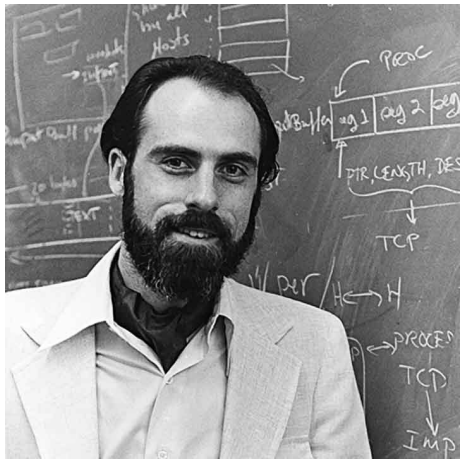
- The Internet is a network of heterogenous networks
  - Hosts (Servers, Laptops, Smartphones, TVs)
  - Network Devices (Routers, Switches)
  - Access technologies (Ethernet, WiFi, 4G, LTE)
  - Physical Media (Coaxial cable, twisted pair, optical fiber)
- The Internet supports diverse applications
  - Email, WWW, File Sharing, Streaming, …

How do all these technologies work together?!

Network intercommunication was the main design goal of the Internet



TCP/IP was designed to enable network intercommunication



A Protocol for Packet Network Intercommunication

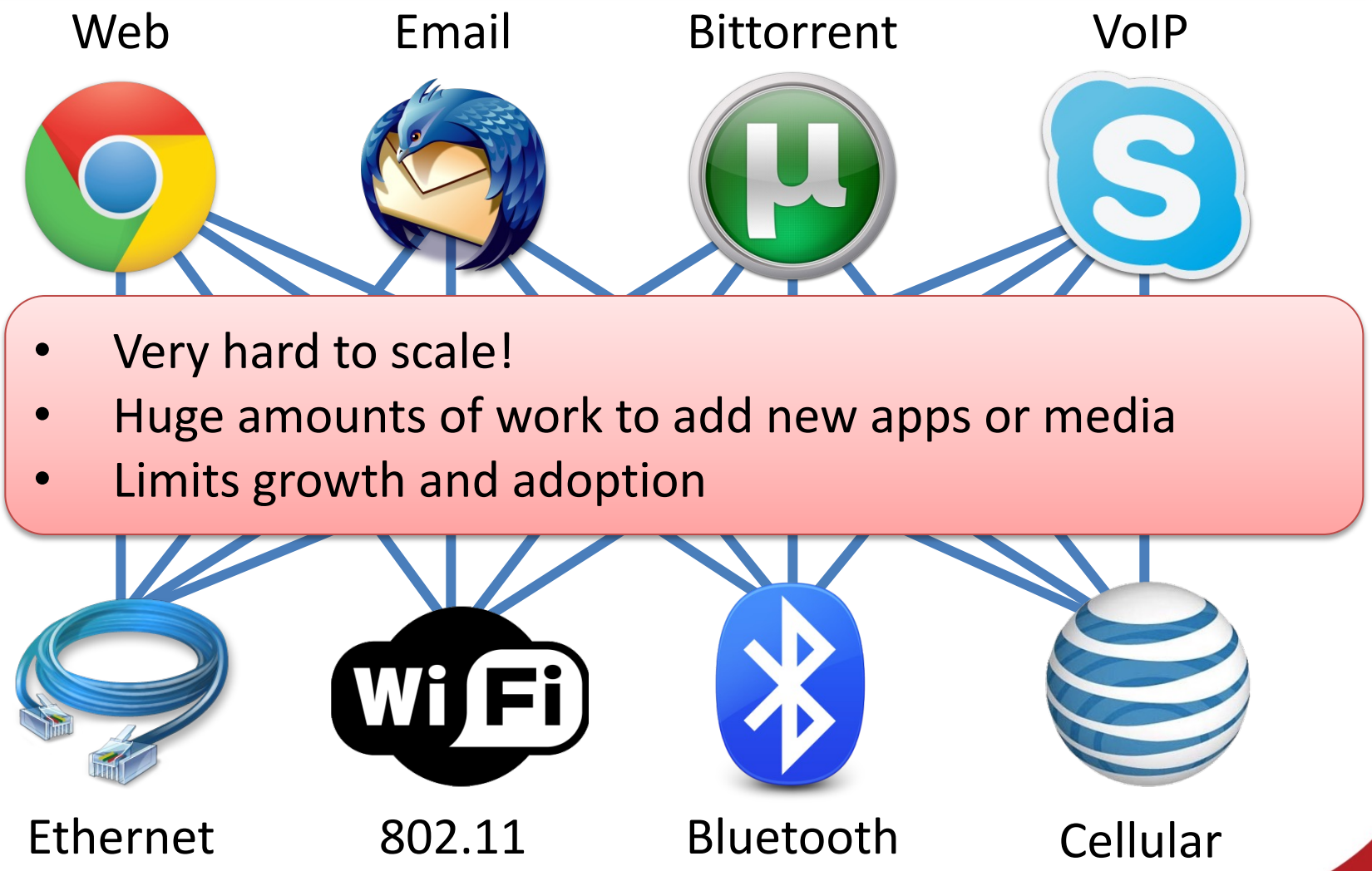
VINTON G. CERF AND ROBERT E. KAHN,
MEMBER, IEEE

Even though many different and complex problems must be solved in the design of an individual packet switching network, these problems are manifestly compounded when dissimilar networks are interconnected. Issues arise which may have no direct counterpart in an individual network and which strongly influence the way in which internetwork communication can take place.

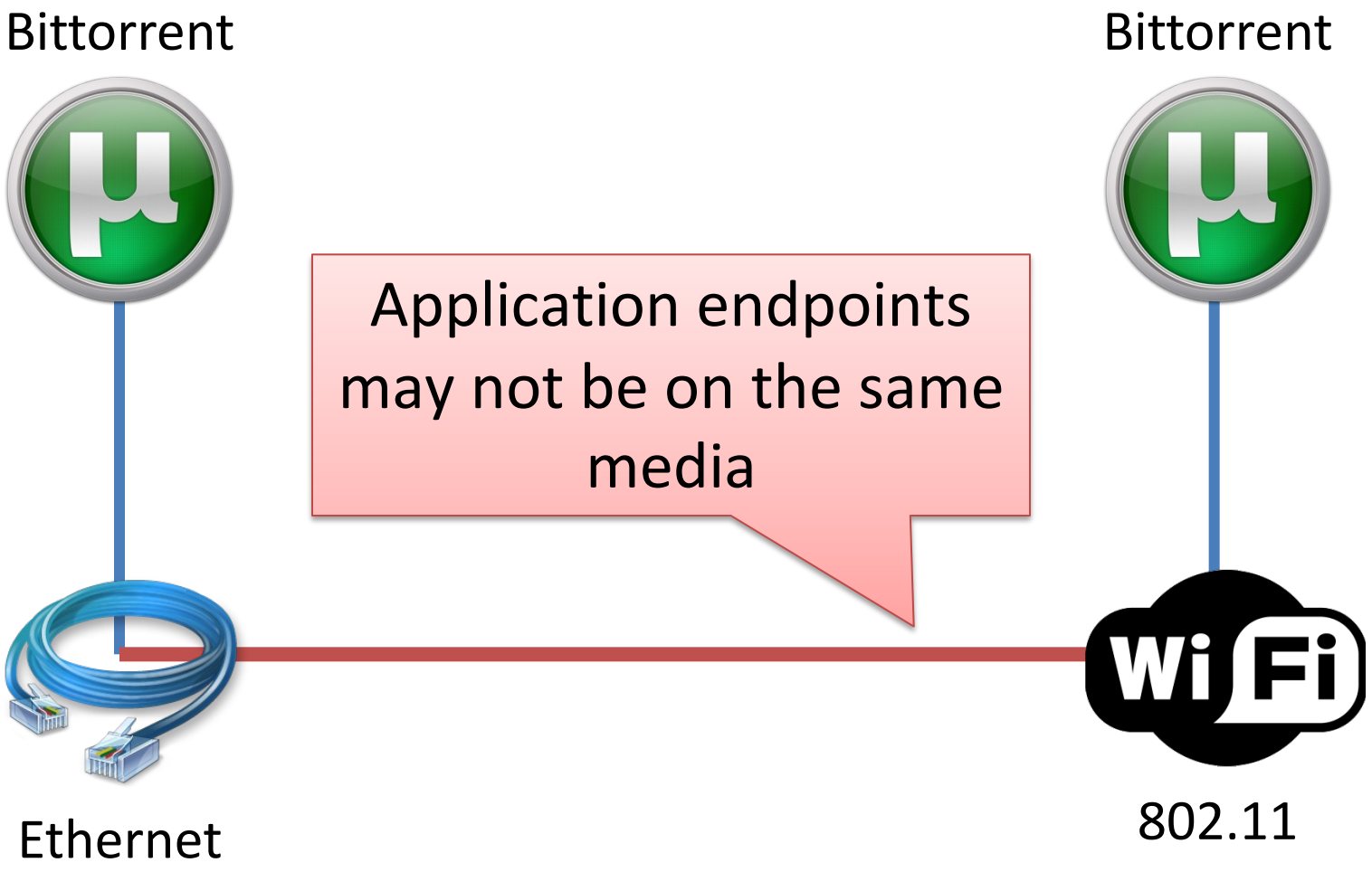
IEEE Trans on Comms, Vol Com-22, No 5 **May 1974**

https://www.cs.princeton.edu/courses/archive/fall06/cos561/papers/cerf74.pdf

Problem scenario

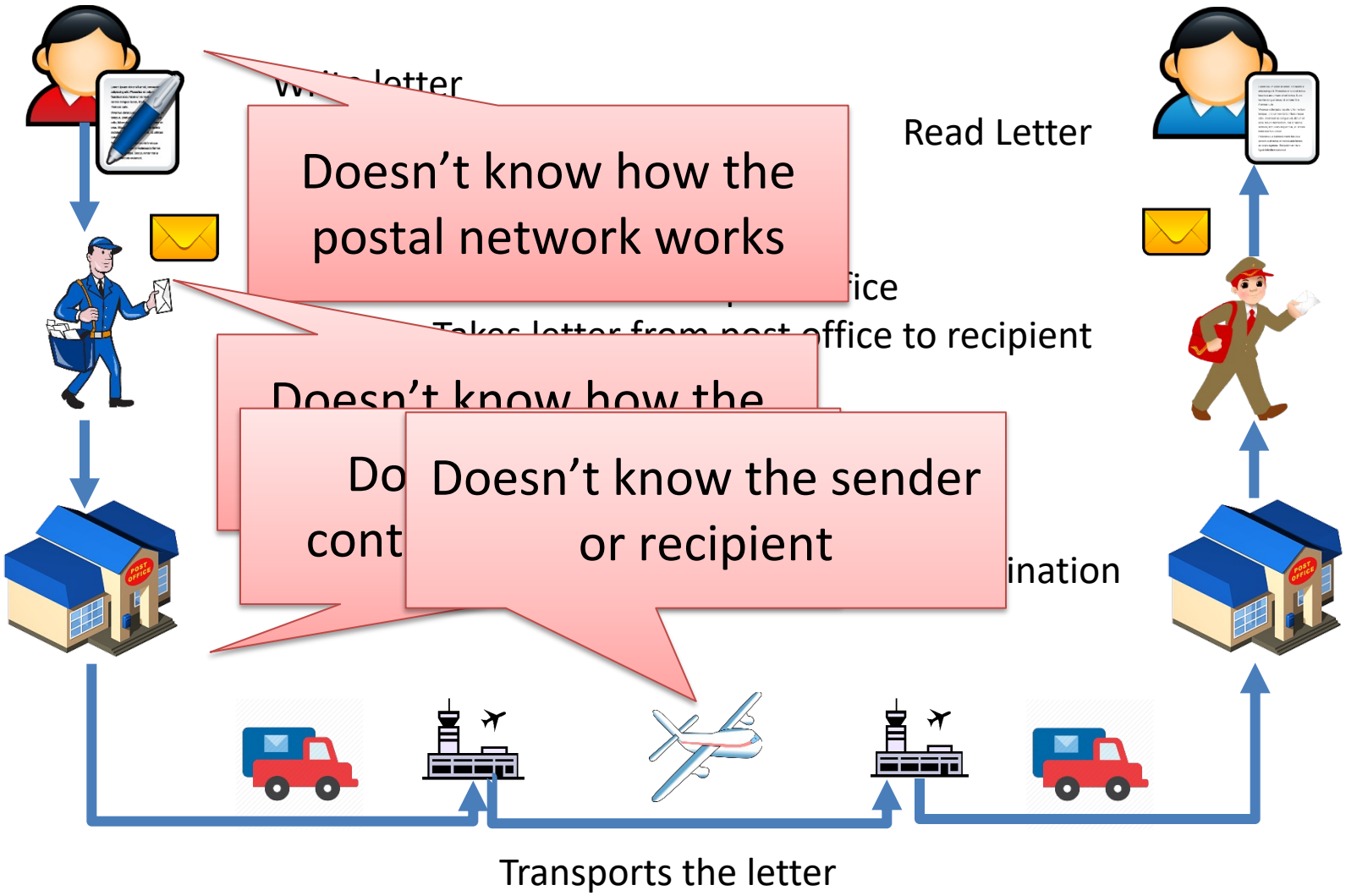


More problems

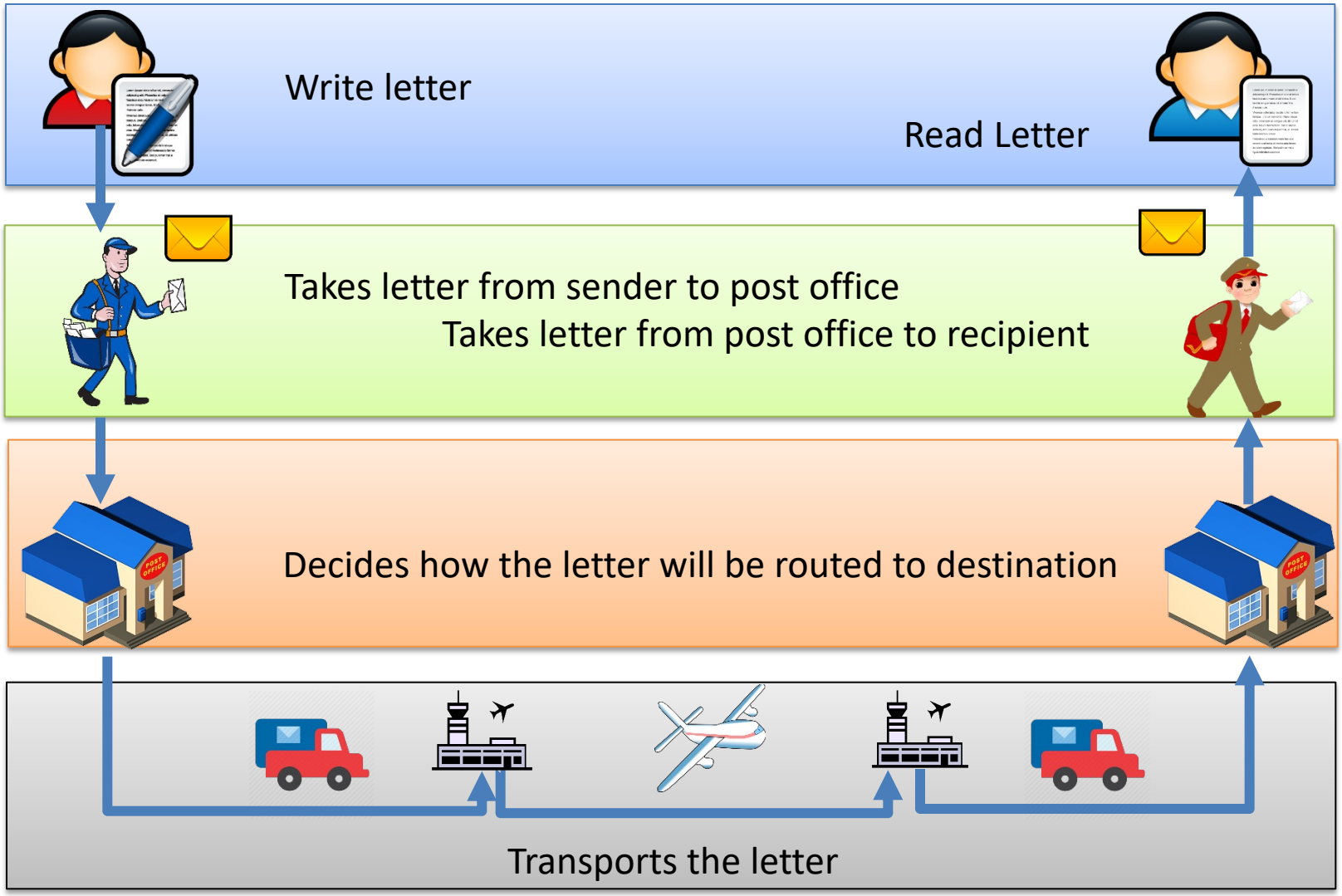


Abstraction is the solution!

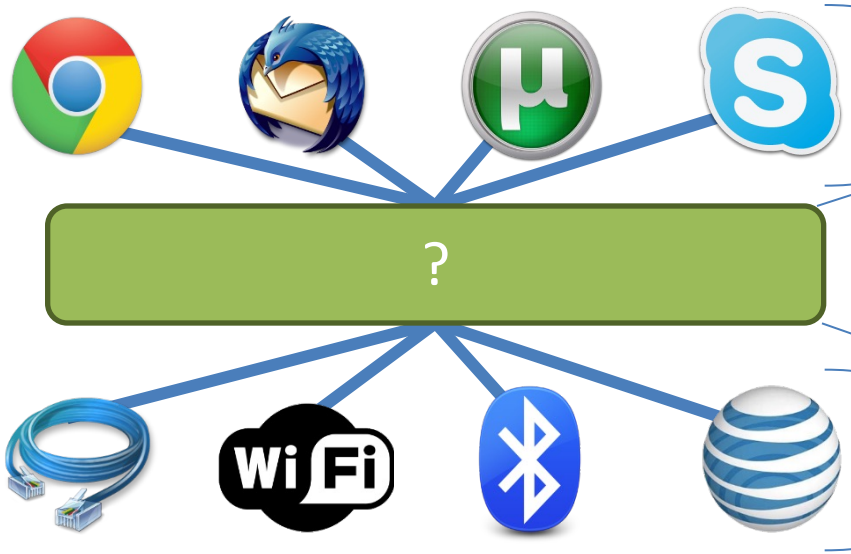
Abstraction in real life



Abstraction leads to layering



How do we divide Internet functionality into layers?

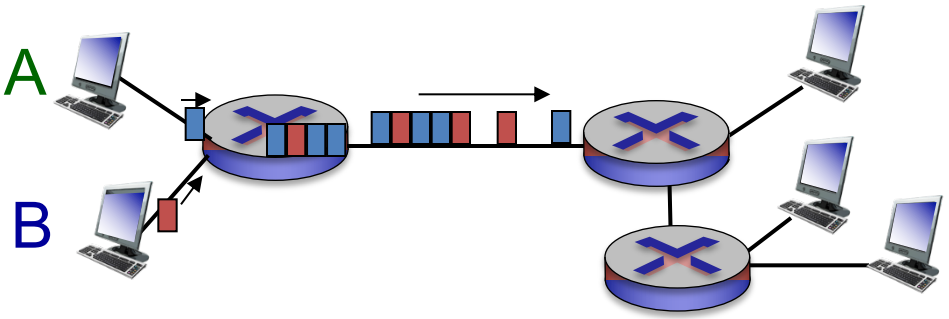


Implement **Application** Services

Transport Data as Packets

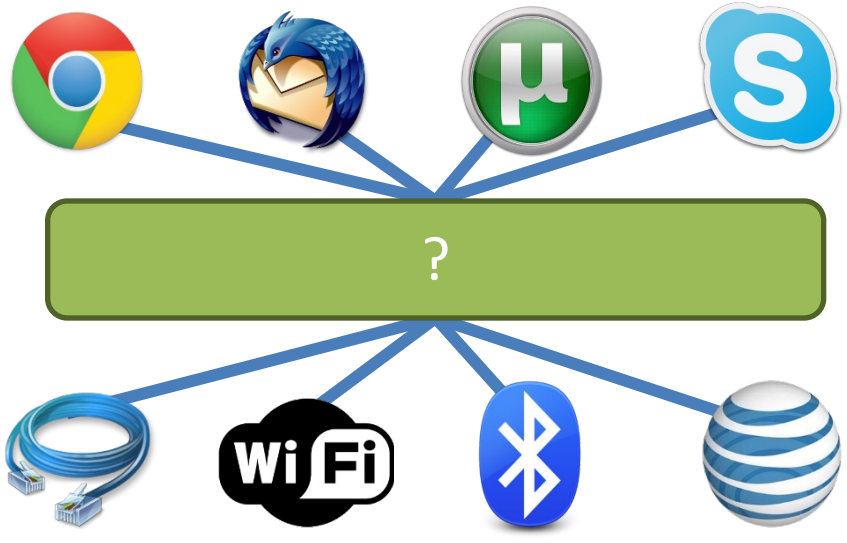
Route Packets across the **Network**

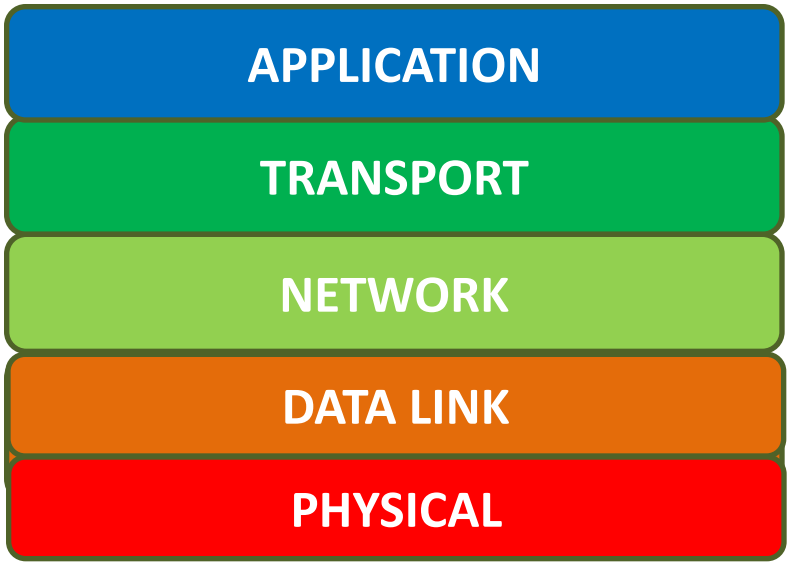
Move data between **physical links**

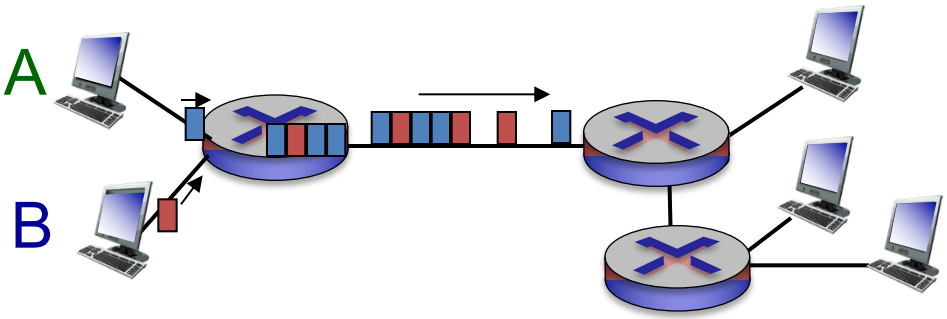


1. **Packet Switching**: the fundamental Internet principle!
2. **Routing**: necessary to interconnect distinct networks
3. **Bit Transmission**: enables data communication

The Internet Protocol Stack (TCP/IP)

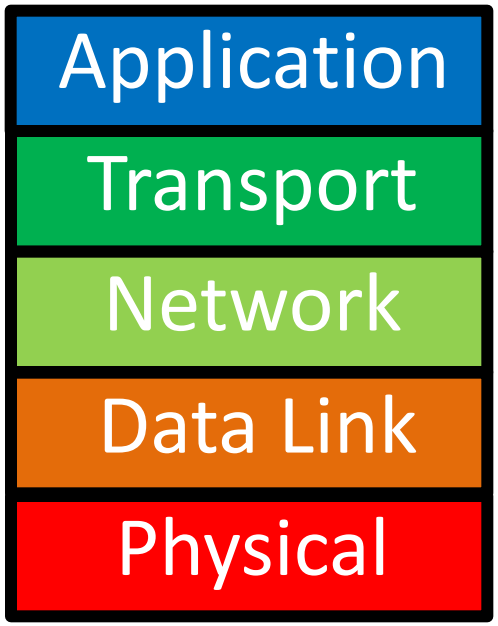






1. **Packet Switching**: the fundamental Internet principle!
2. **Routing**: necessary to interconnect distinct networks
3. **Bit Transmission**: enables data communication

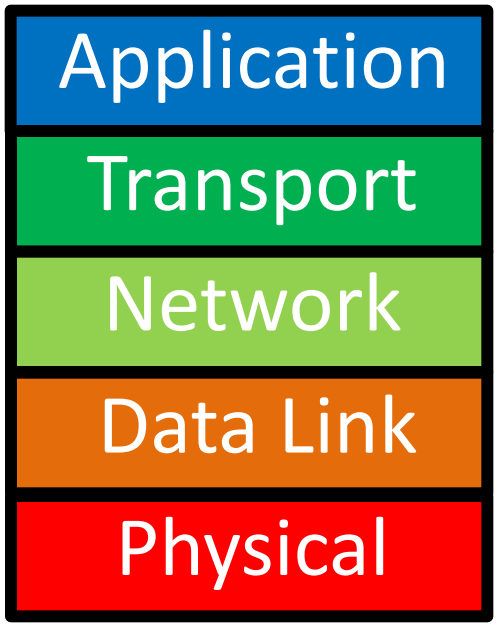
Benefits of Protocol Layering



- Structure
  - Does not specify an implementation
  - Instead, tells us how to organize functionality
- Modularity
  - Eases maintenance, updating of system
  - Allows identification, relationship of complex system's pieces
- Flexibility
  - Reuse of code across the network
- Extensibility
  - Module implementations may change

Lancaster University

Layer Features



- **Service**
  - What does this layer do?
- **Interface**
  - How do you access this layer?
- **Protocol**
  - How is this layer implemented?

Communication Service

Lancaster University

- Specification of
  - Functionality a layer offers
  - How this functionality is provided
  - Interaction between layers (service user & service provider)
- Service definitions abstract from how the service is actually implemented
  - Service interface hides complexity
  - Service is implemented through protocols

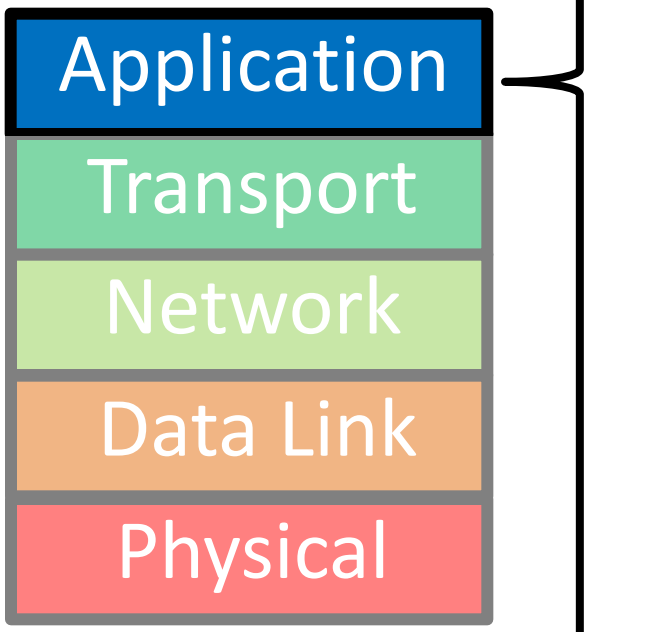
Communication Protocol

Lancaster University

- Specification of
  - How and when data is transmitted and received
  - Allowed messages and expected/required replies
  - Ordering and timing of transmissions
  - Exact format of transmitted data

Lancaster University

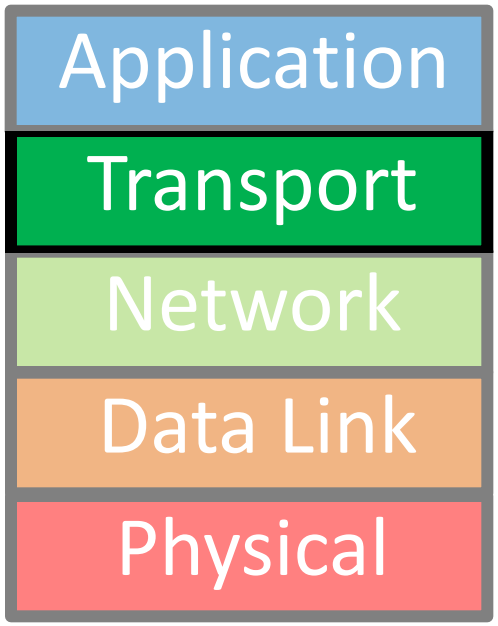
Internet Protocol Stack: Application Layer



- Service
  - Whatever we want :)
- Interface
  - Whatever you want :D
- Protocol
  - Whatever you want ;)
- Examples:
  - File Transfer (FTP)
  - World Wide Web (HTTP)
  - Email (SMTP)
  - Directory Services (DNS)

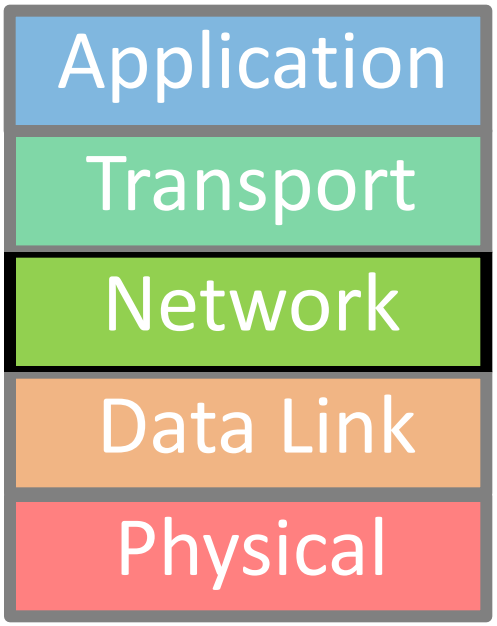
Lancaster University

Internet Protocol Stack: Transport Layer



- Service
  - Multiplexing/demultiplexing
  - Congestion control
  - Reliable, in-order delivery
- Interface
  - Send message to a destination
- Protocol
  - Port numbers
  - Reliability/error correction
  - Flow-control information
- Examples:
  - Transmission Control Protocol (TCP)
  - User Datagram Protocol (UDP)

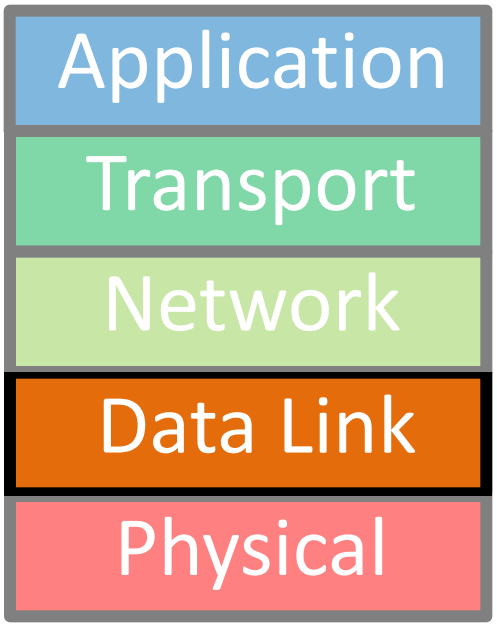
Internet Protocol Stack: Network Layer



- Service
  - Deliver packets across the network
  - Handle fragmentation/reassembly
  - Packet scheduling
  - Buffer management
- Interface
  - Send a packet to a specific destination
- Protocol
  - Define globally unique addresses
  - Maintain routing tables
- Example: Internet Protocol (IP), IPv6

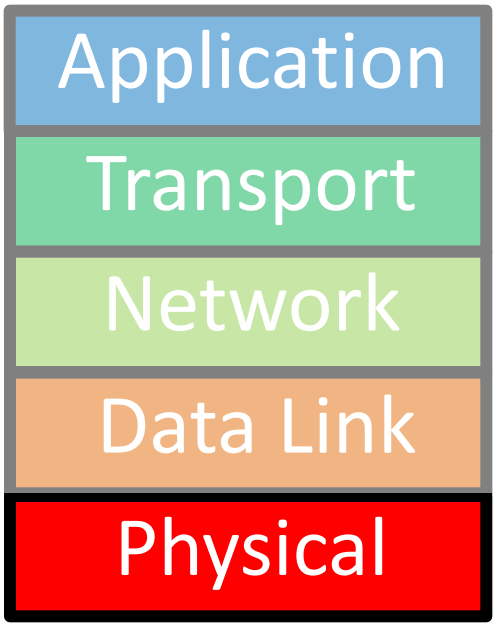
Lancaster University

Internet Protocol Stack: Data Link Layer



- Service
  - Data framing: boundaries between packets
  - Media access control (MAC)
  - Per-hop reliability and flow-control
- Interface
  - Send one packet between two hosts connected to the same media
- Protocol
  - Physical addressing (e.g. MAC address)
- Examples:
  - Ethernet
  - Wifi
  - DOCSIS

Internet Protocol Stack: Physical Layer

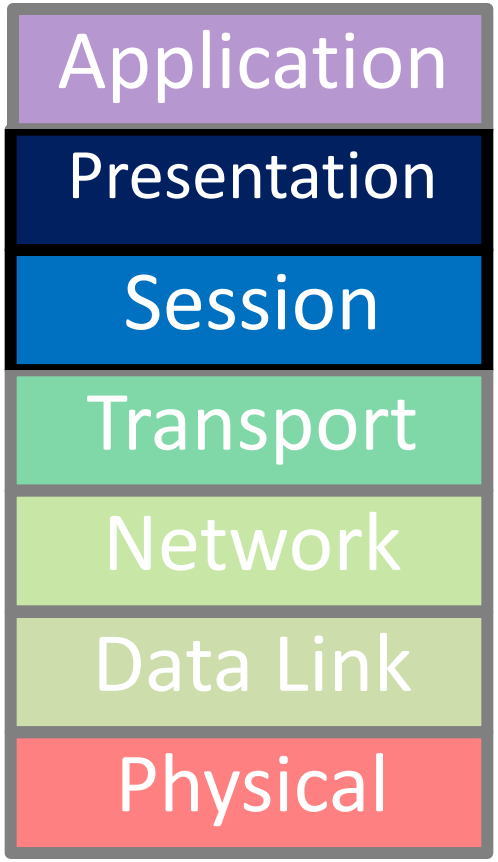


- Service
  - Move information between two systems connected by a physical link
- Interface
  - Specifies how to send one bit
- Protocol
  - Encoding scheme for one bit
  - Voltage levels
  - Timing of signals
- Examples:
  - coaxial cable
  - fiber optics
  - radio frequency transmitters
  - Twisted pair
  - Infrared (IR)

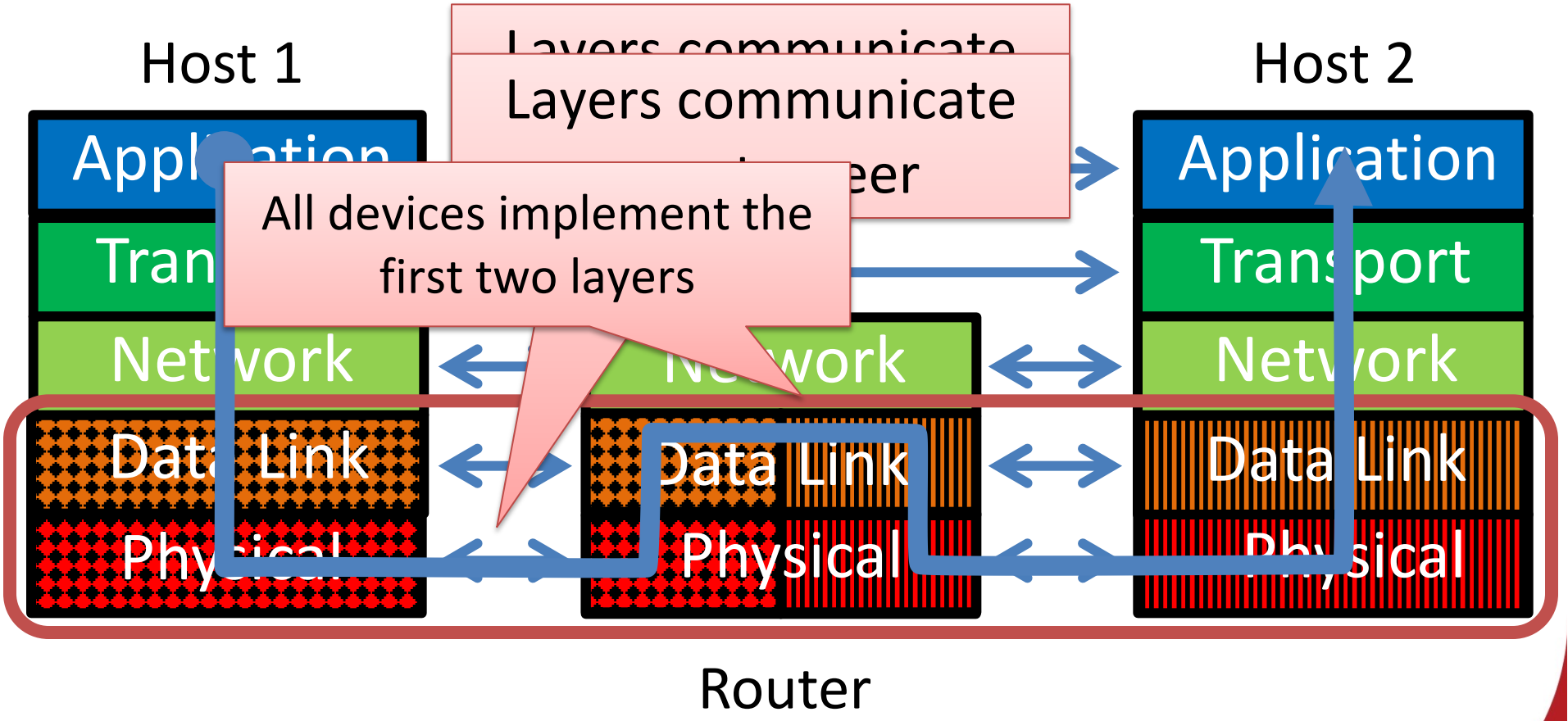
The ISO/OSI model

Lancaster University

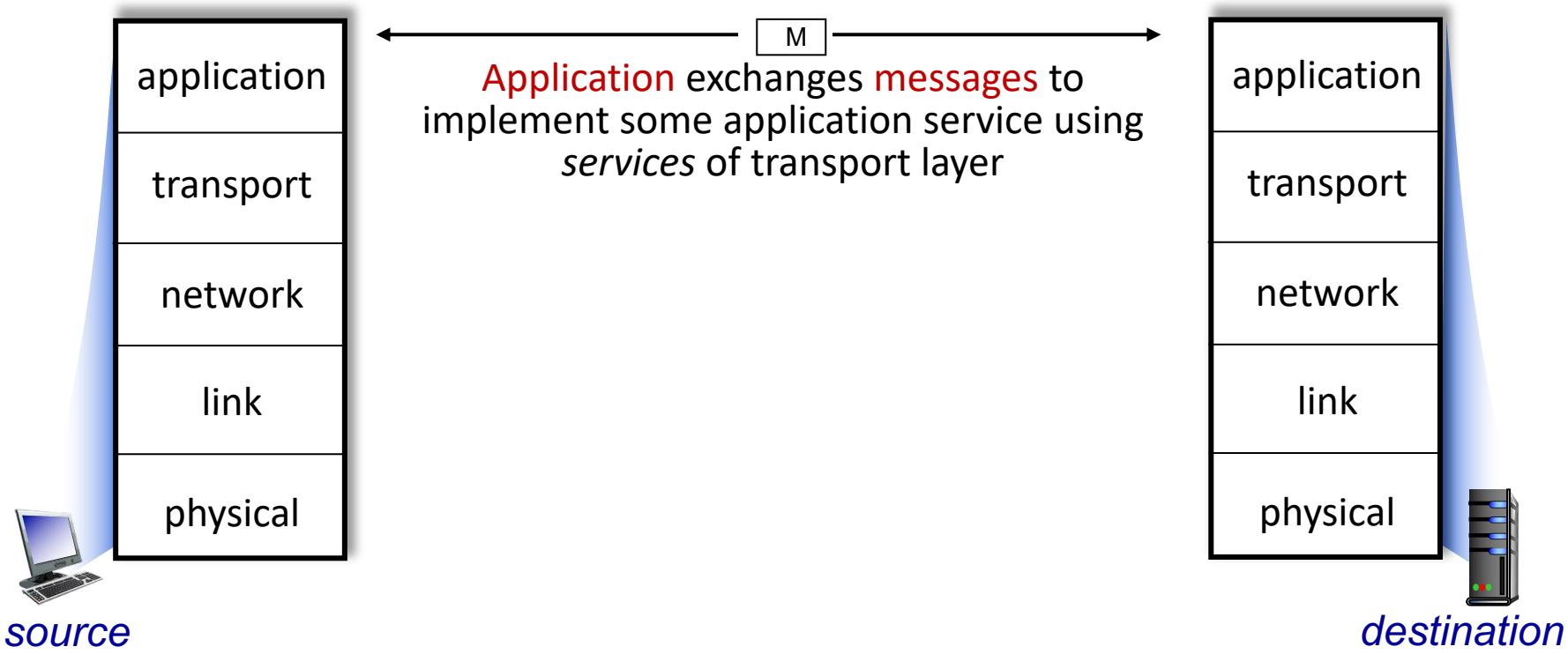
- Presentation: allow applications to interpret meaning of data, e.g., encryption, compression, machine-specific conventions
- Session: synchronization, checkpointing, recovery of data exchange
- Internet model "misses" these layers!
  - These services, *if needed,* must be implemented in application
  - Needed?



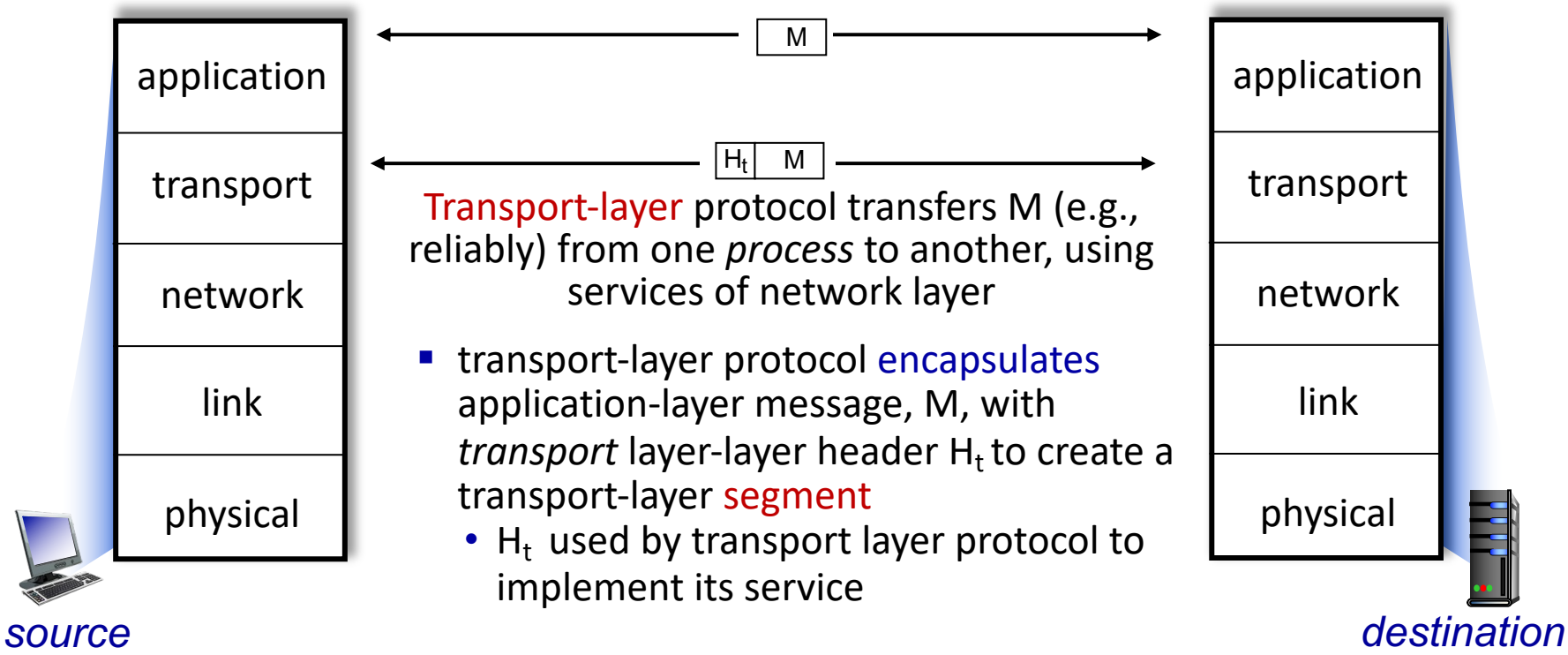
Data flow in the TCP/IP model



Services, Layering and Encapsulation



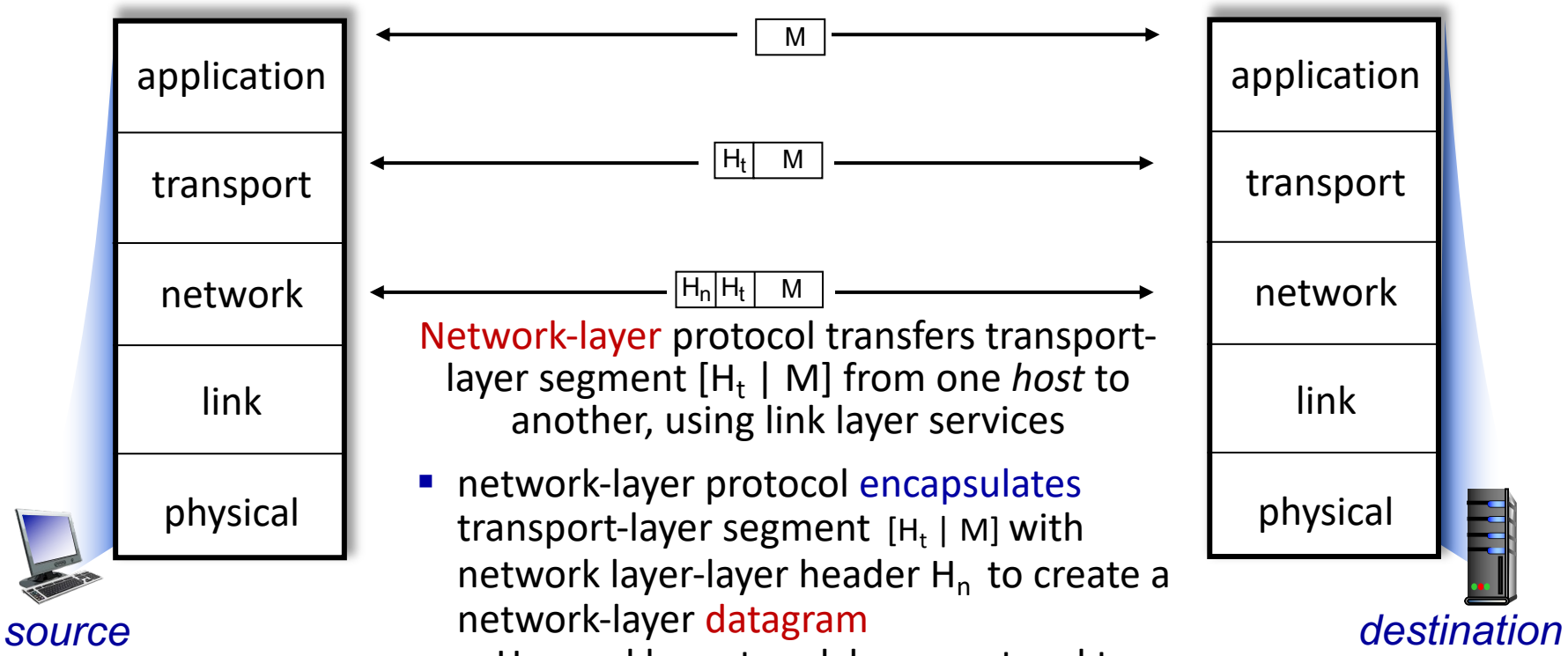
Services, Layering and Encapsulation



Transport-layer protocol transfers M (e.g., reliably) from one *process* to another, using services of network layer

- transport-layer protocol encapsulates application-layer message, M, with *transport* layer-layer header H_t to create a transport-layer segment
  - H_t used by transport layer protocol to implement its service

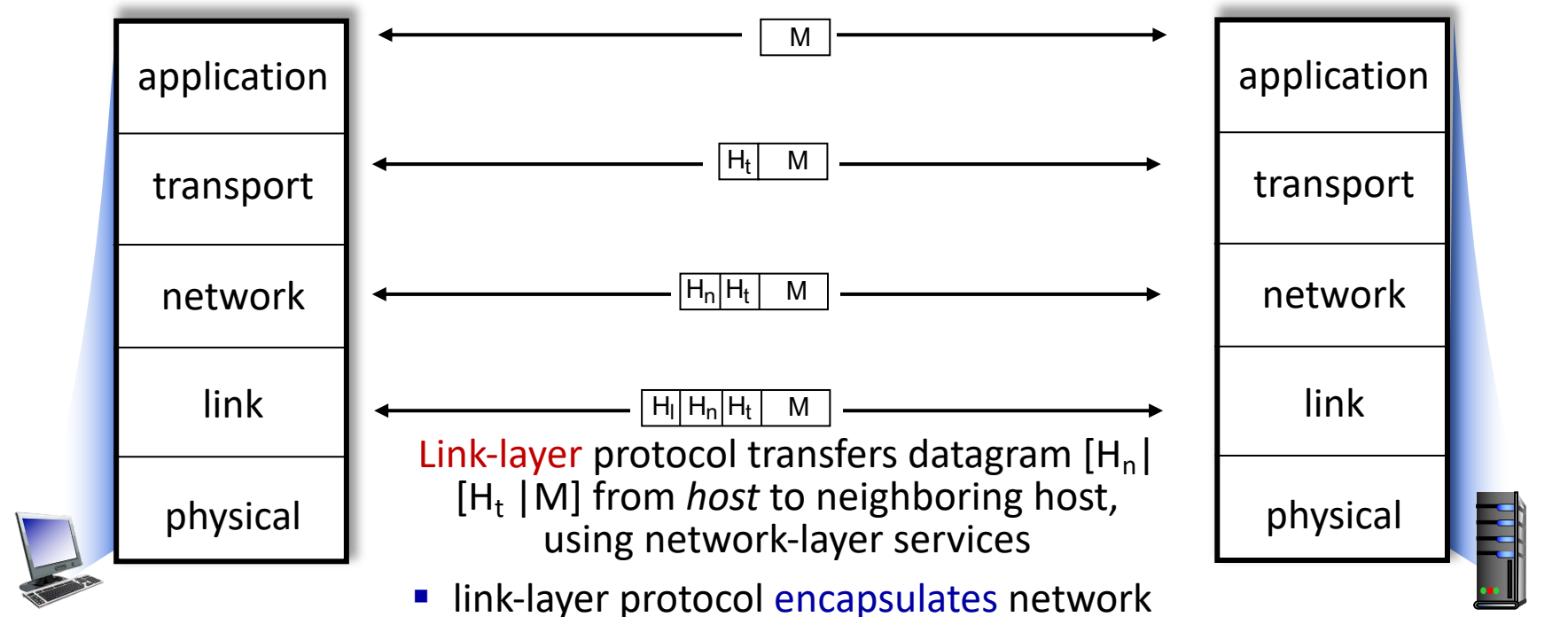
Services, Layering and Encapsulation



Network-layer protocol transfers transport-layer segment [H_t | M] from one *host* to another, using link layer services

- network-layer protocol encapsulates transport-layer segment [H_t | M] with network layer-layer header H_n to create a network-layer datagram
  - H_n used by network layer protocol to implement its service

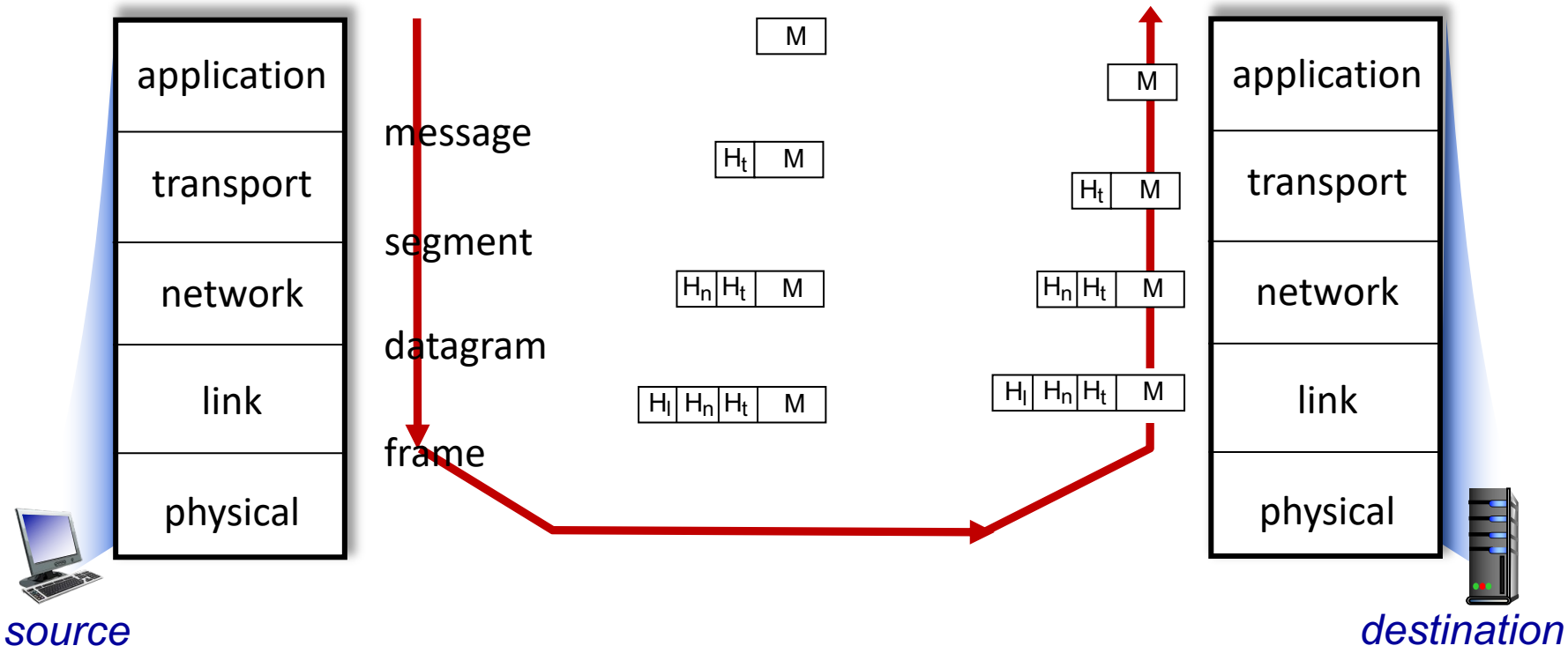
Services, Layering and Encapsulation



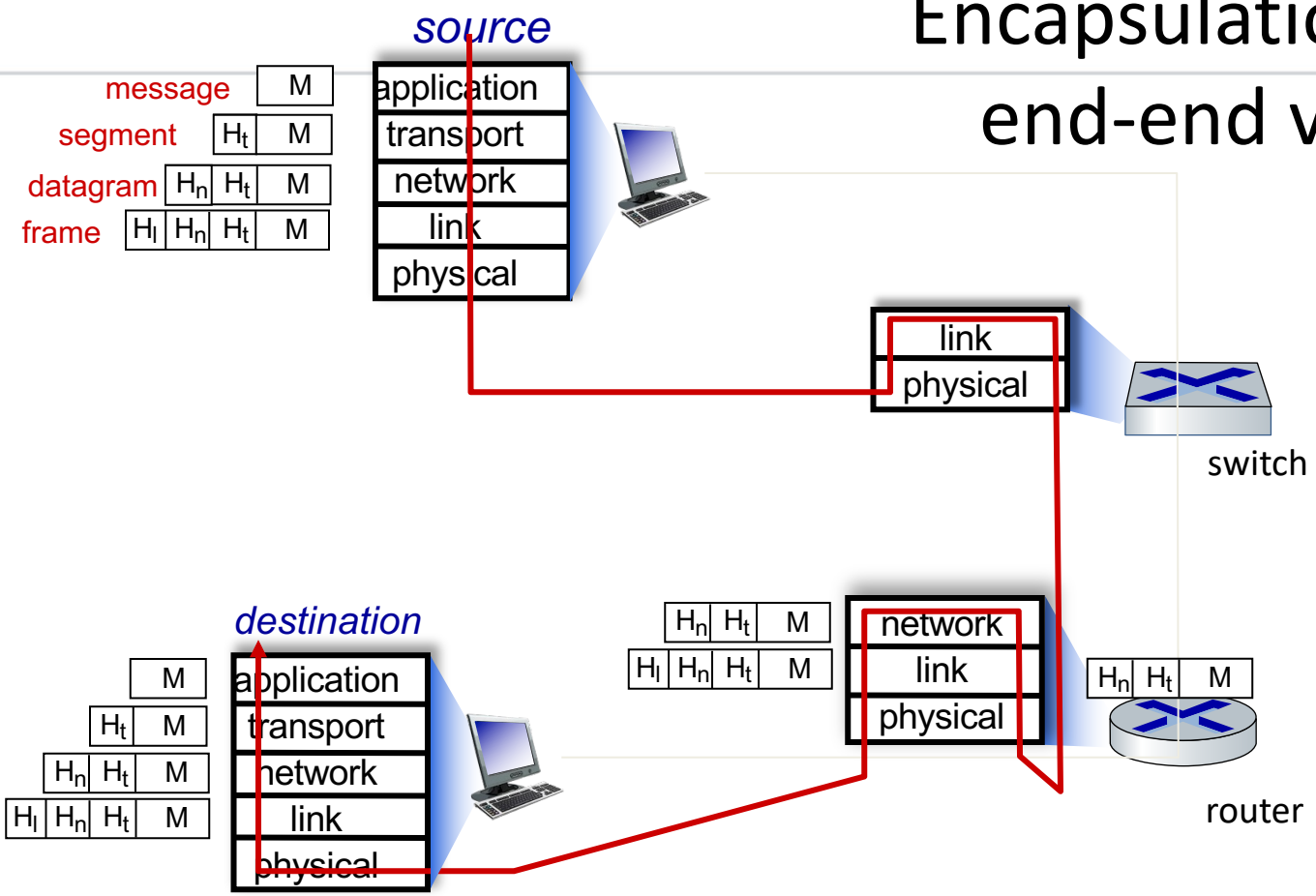
Link-layer protocol transfers datagram $[H_n|[H_t|M]$ from *host* to neighboring host, using network-layer services

- link-layer protocol encapsulates network datagram $[H_n|[H_t|M]$, with link-layer header H_l to create a link-layer frame

Services, Layering and Encapsulation



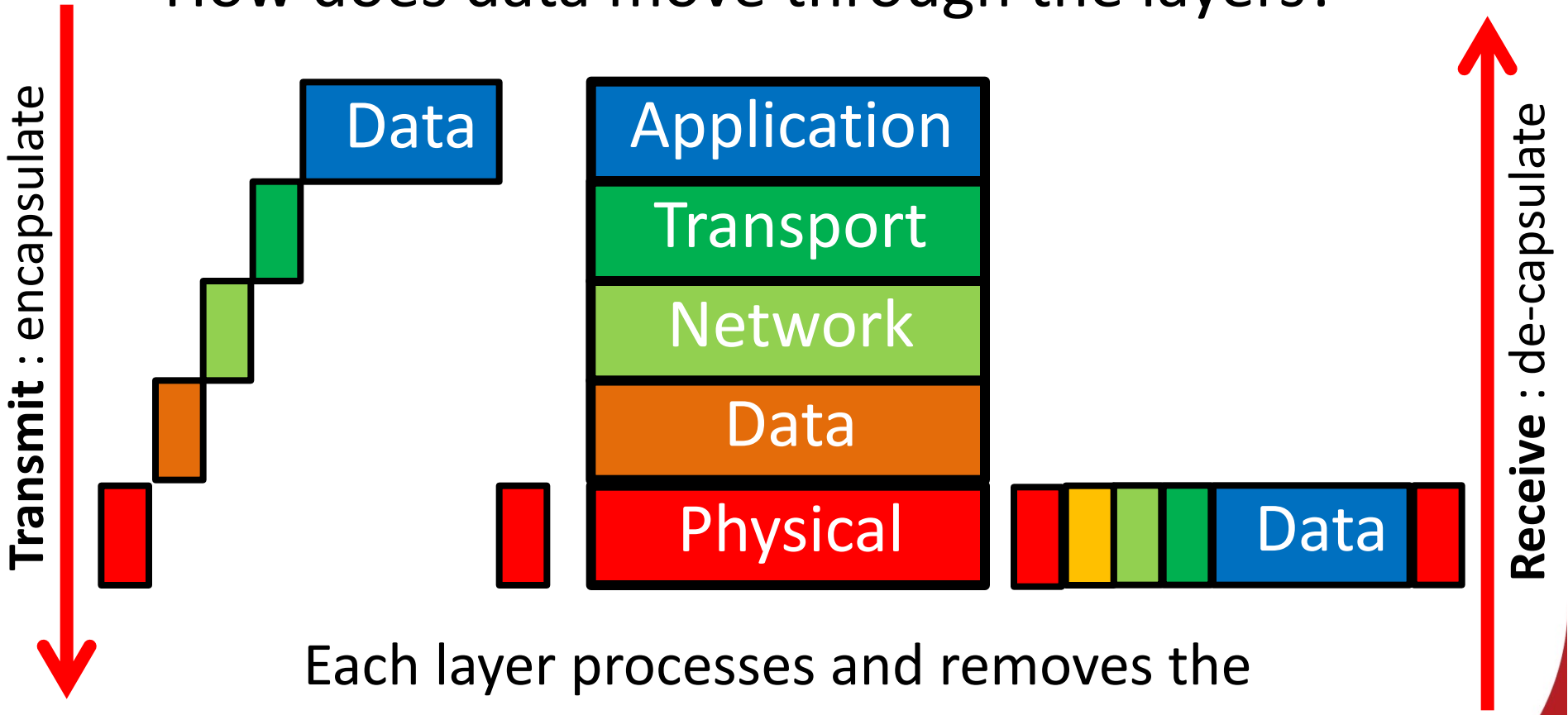
Encapsulation: an end-end view



Each layer processes and removes the corresponding header/trailer

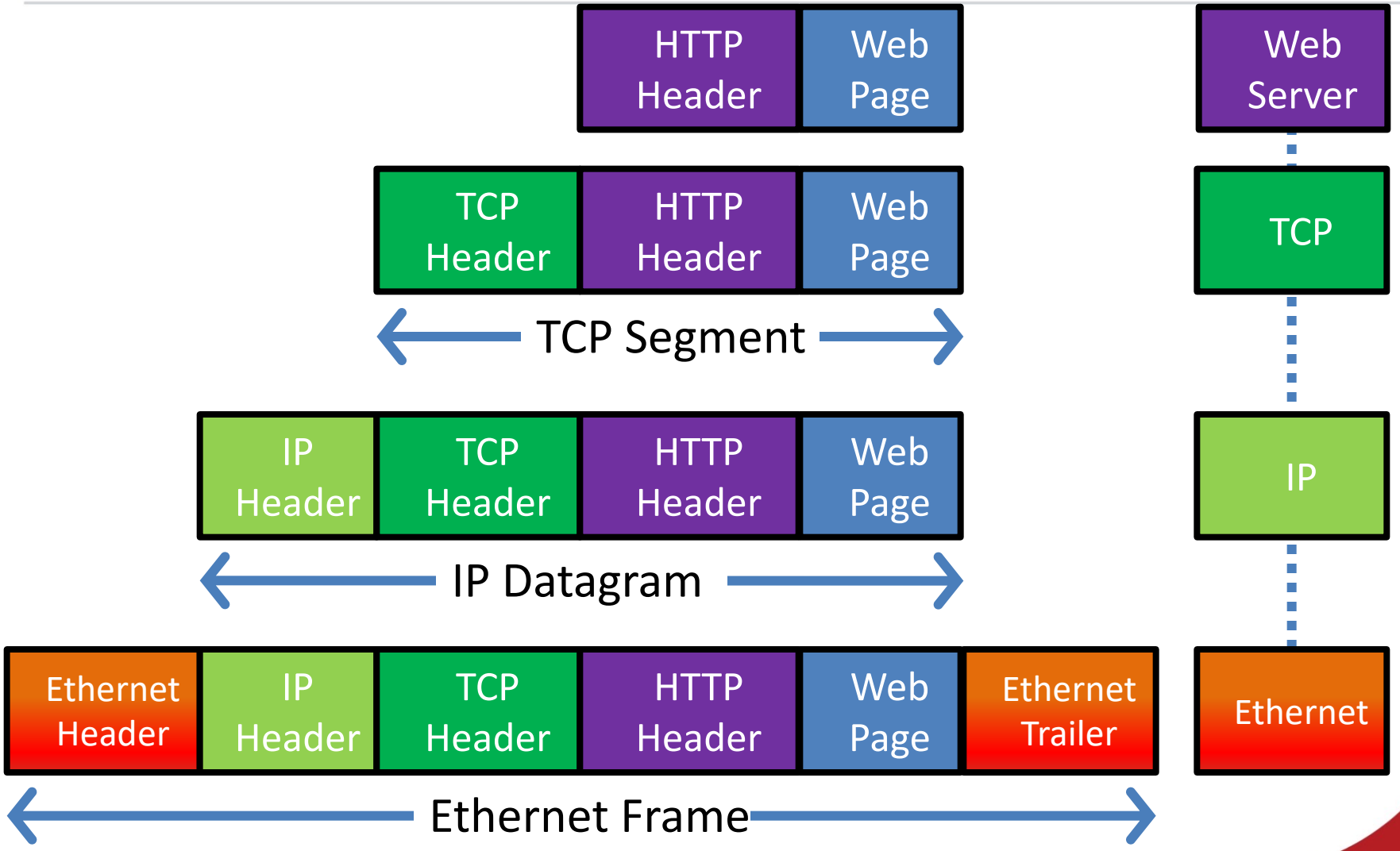
Encapsulation

How does data move through the layers?

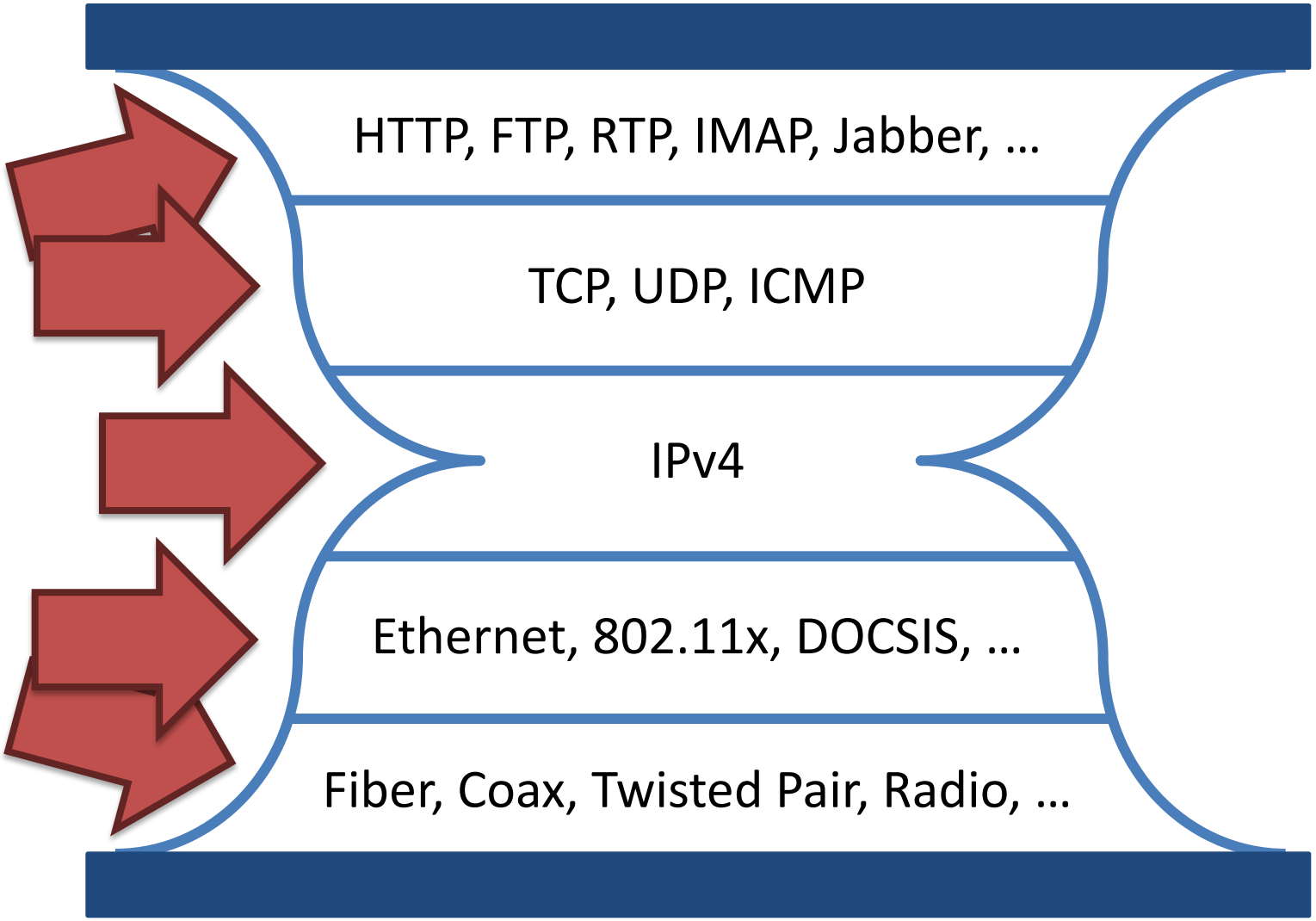


Each layer processes and removes the corresponding header/trailer

Encapsulation in practice



The Internet Hourglass



The Internet Hourglass

Lancaster University

Think about the difficulty of deploying IPv6…

- One Internet layer means all networks interoperate
- All applications function on all networks
- Room for development above and below IP
- But, changing IP is insanely hard

Fiber, Coax, Twisted Pair, Radio, …

Why is the Internet core is "dump" by design

- Minimal functionality in Layer below the Application layer
- The **End-to-End Principle**:

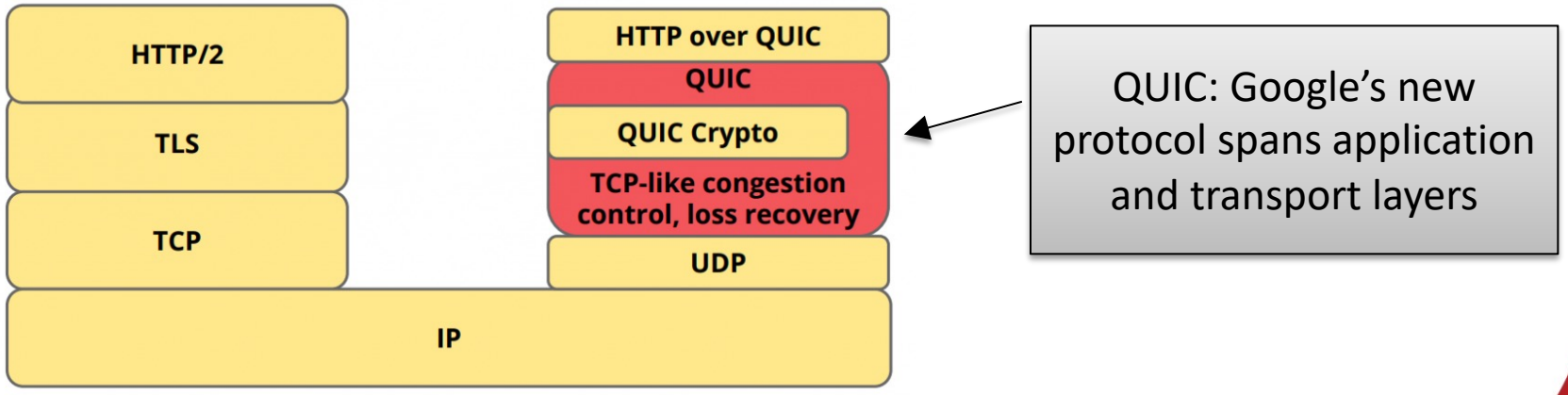
> *Don't implement anything in the network that can be implemented correctly by the hosts*

- Generality, Net Neutrality, Low cost and complexity
- Tradeoffs?

David Clark

Layering sometimes can be harmful

- Lower performance due to data and processing overhead added by protocol headers
- The same functionality may be duplicated by different layers (e.g. error detection)
- Lack of transparency in lower layers hinders optimization



Reality Check

Lancaster University

- Layering regularly violated

Firewalls

Transparent Proxies

NATs

- Conflicting interests:
  - Architectural purity
  - Commercial necessity

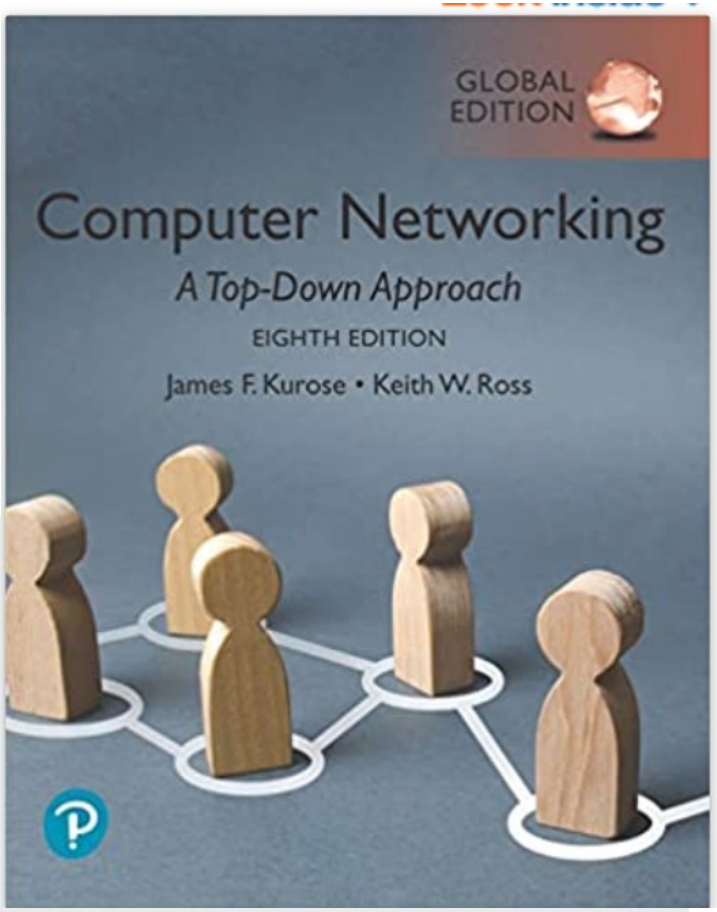
Summary

Lancaster University

- Basic Principles of layered architectures
  - Layering is used to separate communication functions and concerns
  - Each layer is responsible for different tasks
  - A lower layer offers a service to the layer above
- Layering benefits:
  - Manage complexity
  - Modularity
  - Extensibility
- Encapsulation
  - Data Units between the layers
- Layering can be harmful

Required reading

Chapter 1.5 in the main textbook:



Lancaster University

Thanks for listening!

Any questions?