上海交通大学
SHANGHAI JIAO TONG UNIVERSITY

SJTU

Chapter 1 Introduction

Liping Shen 申丽萍

lpshen@sjtu.edu.cn

饮 水 思 源 • 爱 国 荣 校

Chapter 1: Introduction

Our goal:

- get “feel” and terminology
- paint a broad picture to see the forest through the trees
- approach:
  - use Internet as example

Topics:

- What’s Computer Network?
- protocol layers, service models
- basic concepts of data transmission: bandwidth, delay, throughput, multiplexing, switching
- What’s the Internet?
- network edge: hosts, access net, physical media
- network core: packet/circuit switching, Internet structure

Chapter 1: roadmap

- What's Computer Network?
- protocol layers, service models
- basic concepts of data transmission:
  - bandwidth, delay, throughput, multiplexing, switching
- What's the Internet?
- network edge:
  - hosts, access net, physical media
- network core:
  - packet/circuit switching, Internet structure

What is Computer Network

- Collection of autonomous computers interconnected by a single technology

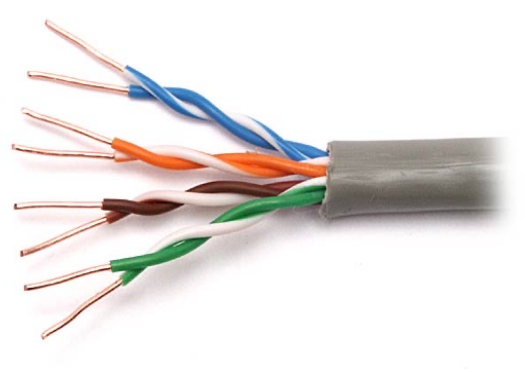
-- From Computer Network by Tanenbaum

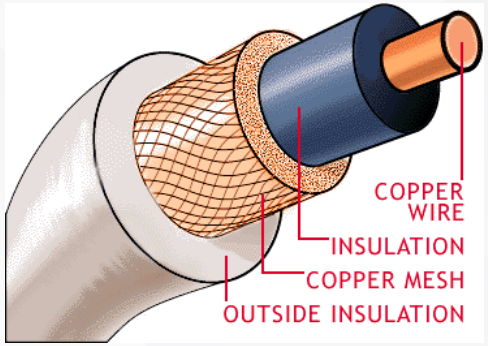
- A collection of computers and devices interconnected by communications channels that facilitate communications among users and allows users to share resources.

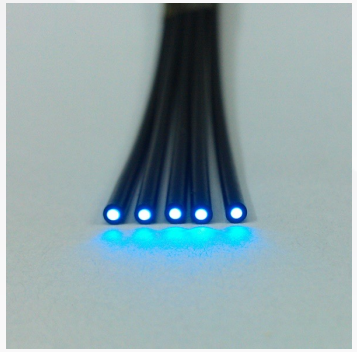
-- From wikipedia

Classification of Network by Media

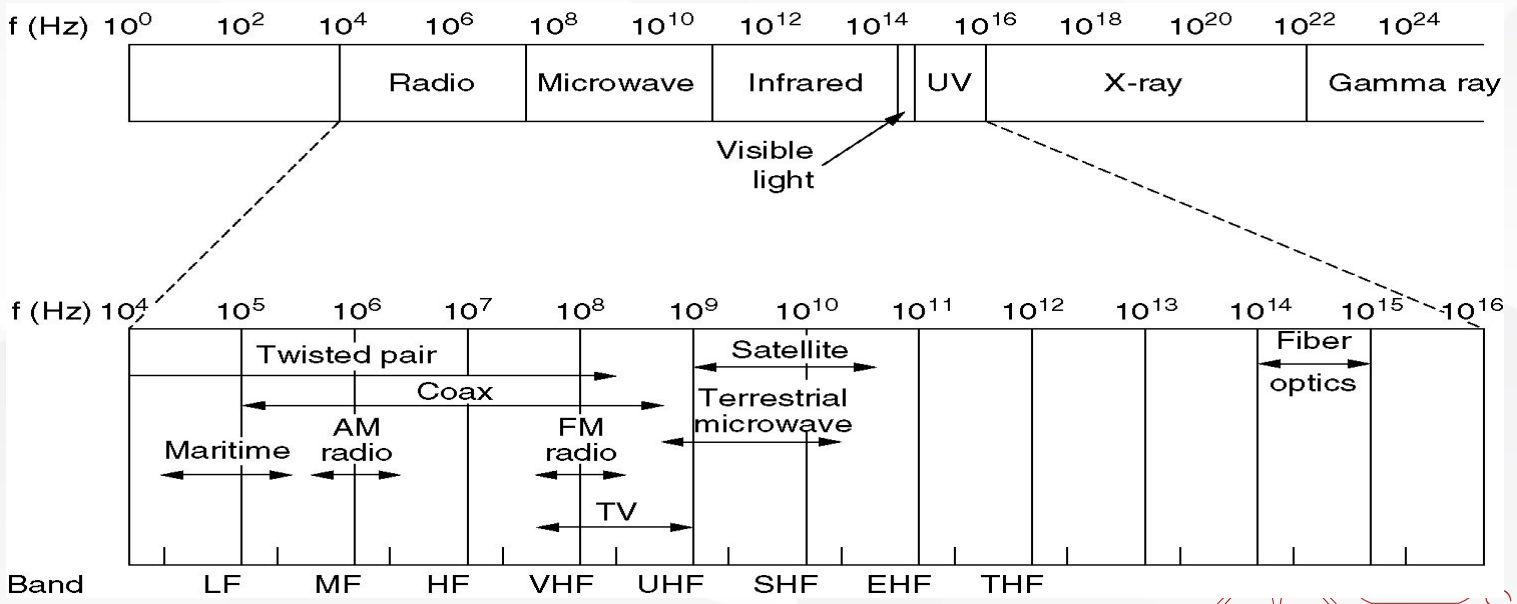
- Wired network:
  - Twisted pair
  - Coaxial cable
  - Optical fiber





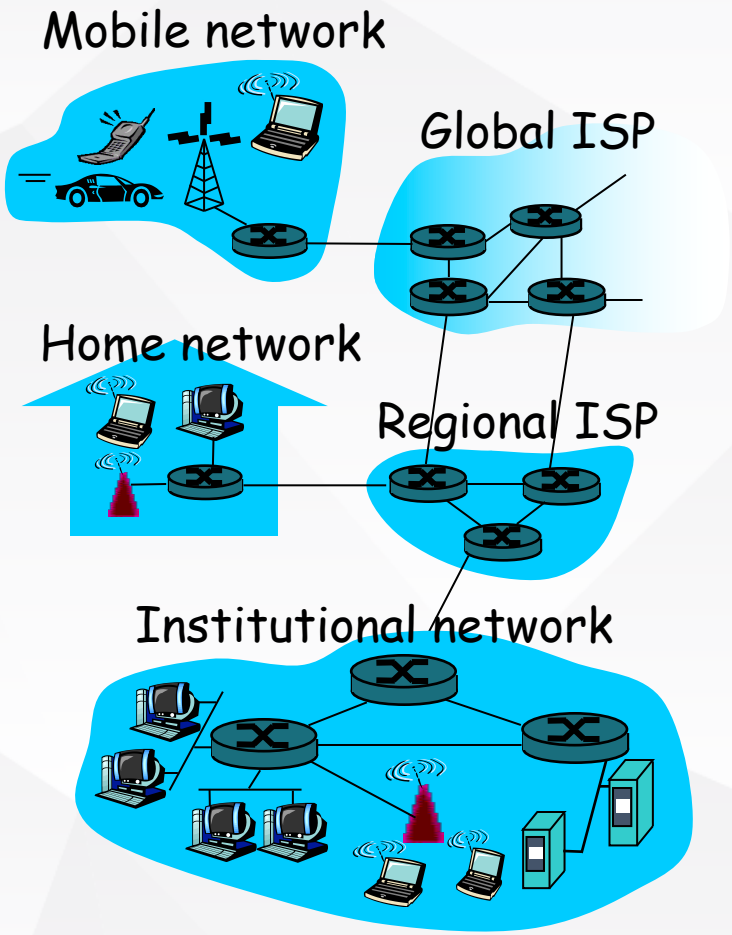


- Wireless network:
  - Mobile ad hoc
  - Wi-Fi
  - Wi-Max
  - Cellular System
  - Satellite



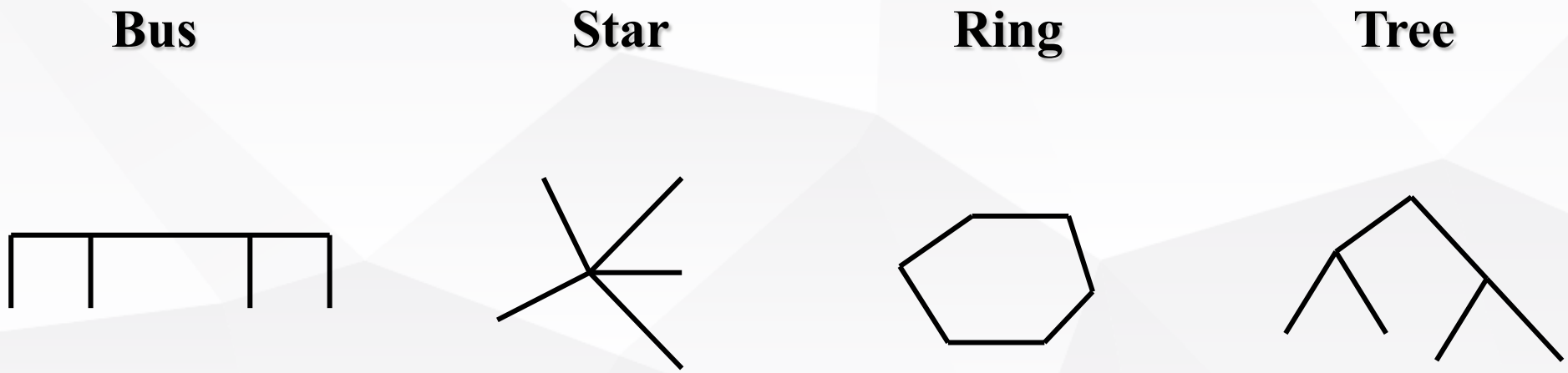
Classification of network by transmission technology

- **Broadcast network:** a single communication channel is shared by all computers=>sending a packet implies that all others receive it.
- **Point-to-point network:** Computers are connected in pairs => sending a packet goes strictly from the sender to the receiver, possibly having to visit intermediate machines (*routing*).



Classification of network by Topology

- **Network topology** is the coordination by which devices in the network are arranged in their logical relations to one another, independent of physical arrangement.

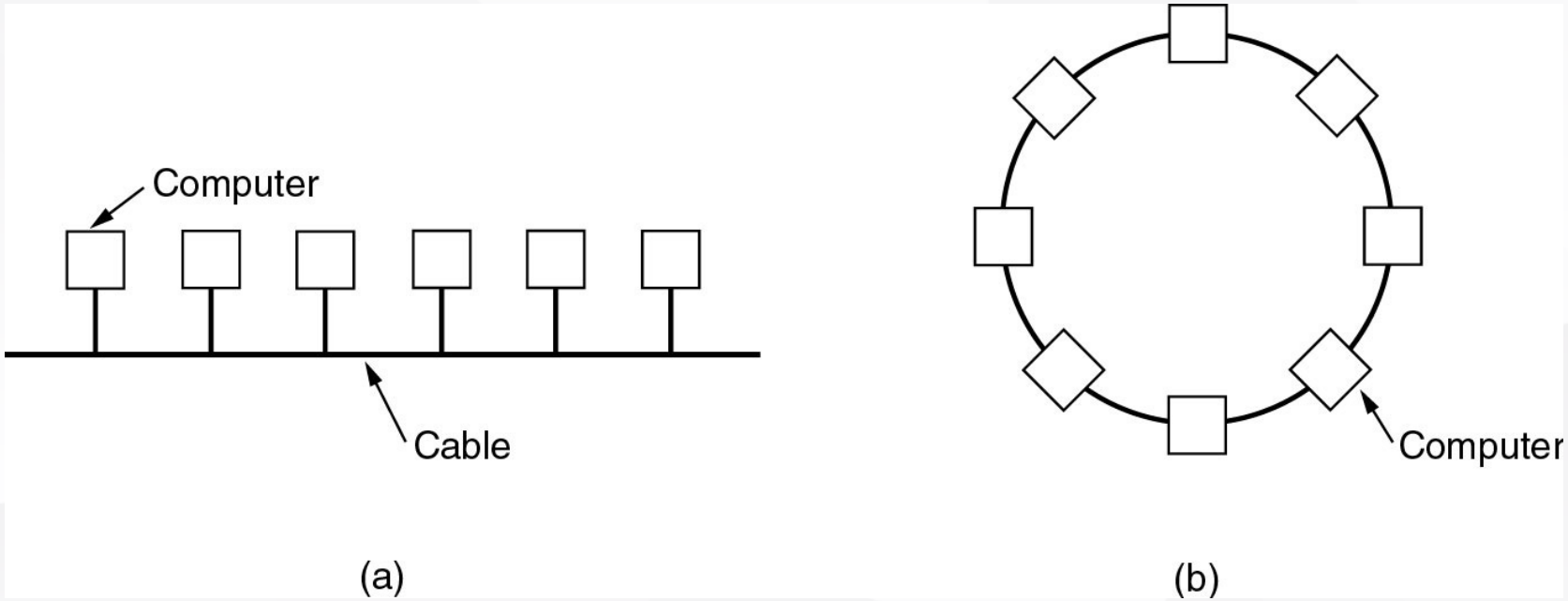


Classification of network by scale

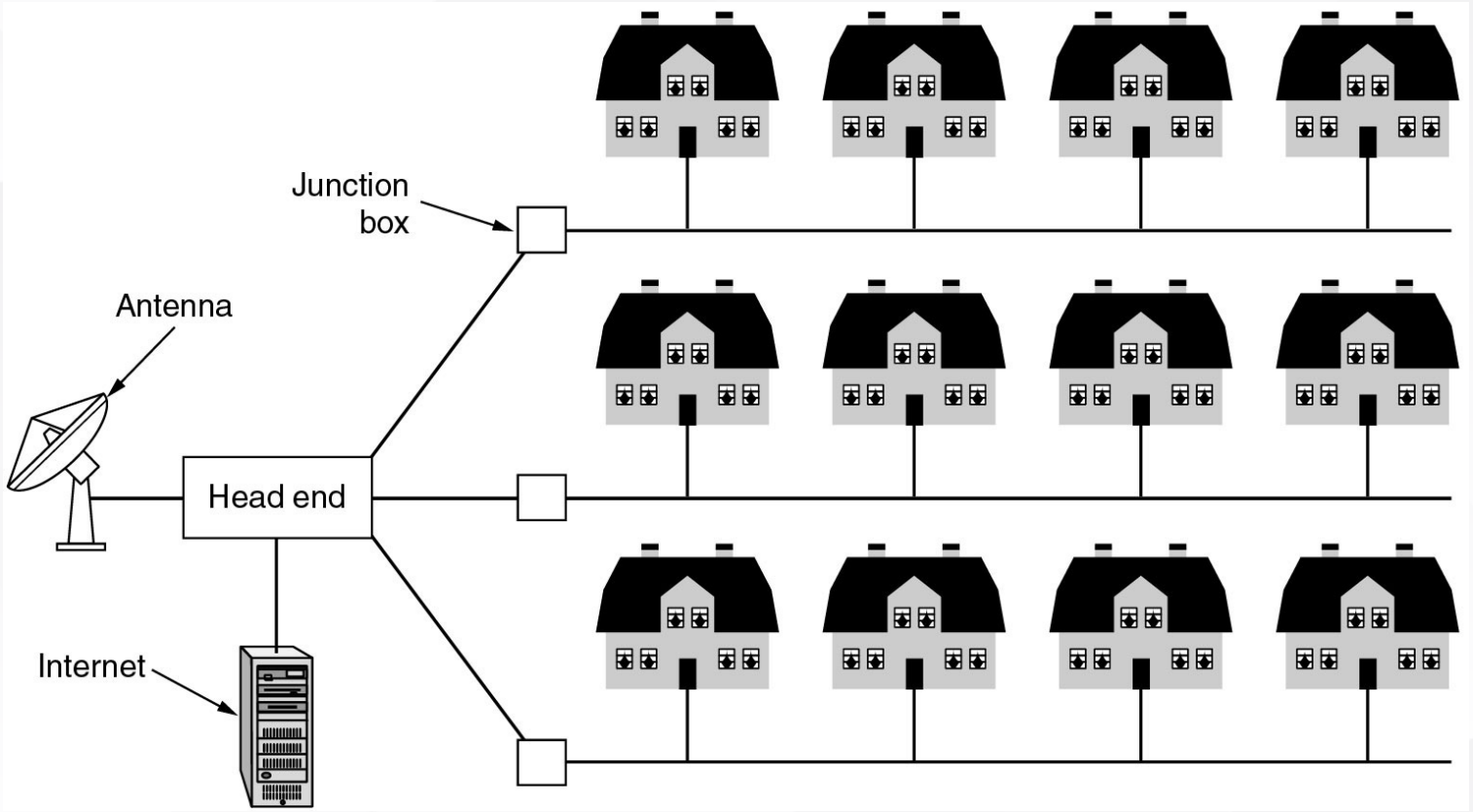
Interprocessor distance	Processors located in same	Example
1 m	Square meter	Personal area network
10 m	Room	Local area network
100 m	Building	
1 km	Campus	
10 km	City	Metropolitan area network
100 km	Country	Wide area network
1000 km	Continent	
10,000 km	Planet	The Internet

Local Area Networks

- Apart from scale, LANs distinguish themselves from other networks by (generally) using **broadcast** technology,
- and having simple **topologies**:

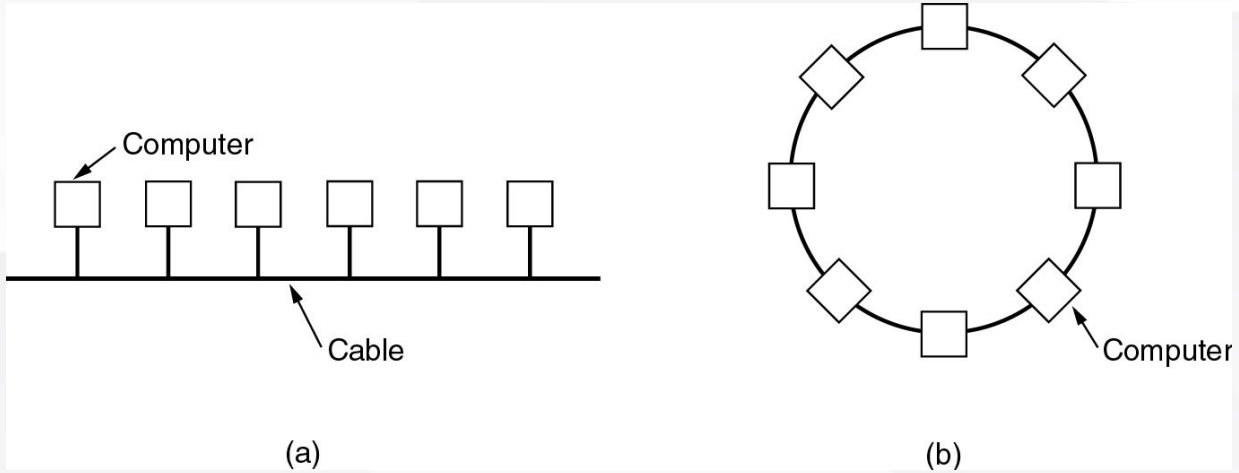


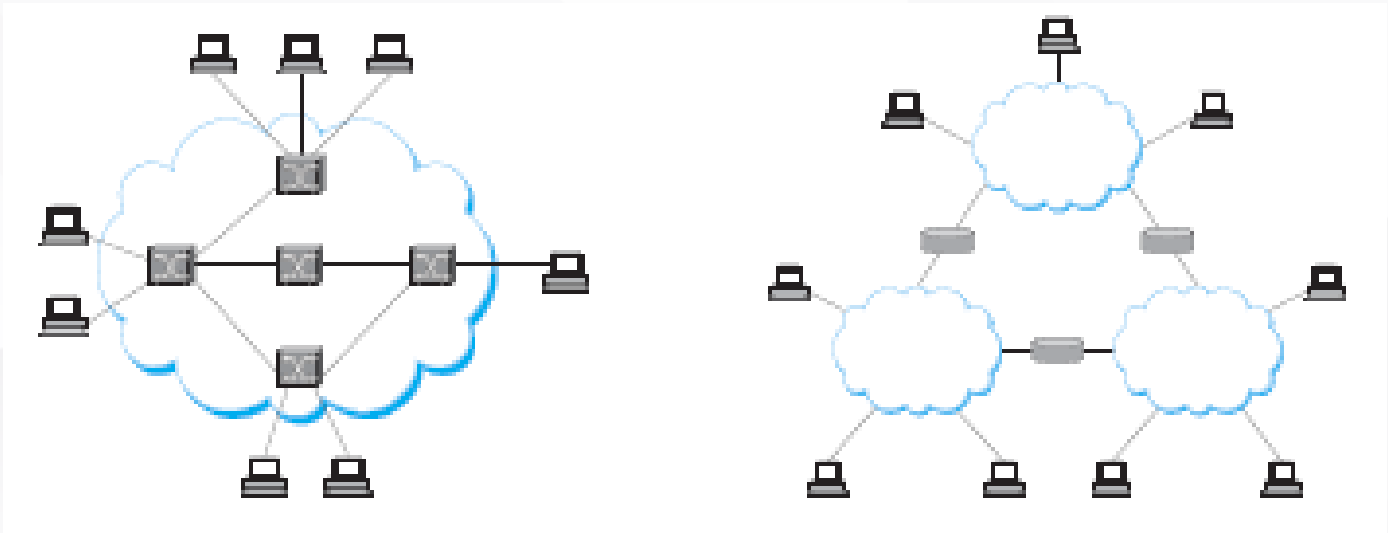
Metropolitan Area Networks



A metropolitan area network based on cable TV.

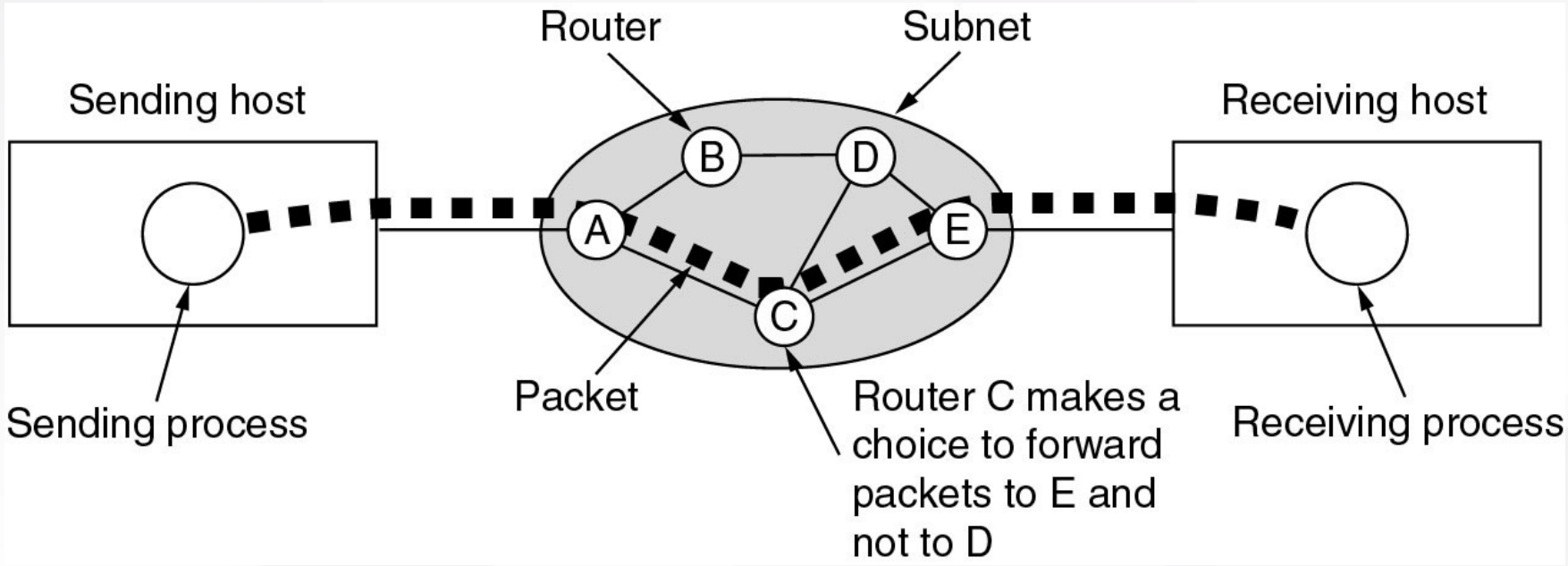
From Links to Networks





From Links to Networks

- How to send packets from sender to receiver that single link couldn't reach?
- through switching nodes (switches, routers…)



Resources Subnet

Network edge

Communication Subnet

Network Core / backbone

Resources Subnet

Network edge

Internetworks

- The assumption so far is that a network is **homogeneous**: there is hardly any variation in hardware and software. In practice, large networks can only be constructed by interconnecting different kinds of/ **heterogeneous** networks=> internet(work).
- internet(work): connecting LANs to each other through a WAN
- Connecting WANs to each other (the Internet).

Internetworks

- The assumption so far is that a network is **homogeneous**: there is hardly any variation in hardware and software. In practice, large networks can only be constructed by interconnecting different kinds of/ **heterogeneous** networks=> internet(work).
- internet(work): connecting LANs to each other through a WAN
- Connecting WANs to each other (the Internet).

Components of Computer Networks

- **Hardware**: how you can connect computers into a network:
  - Network interface cards / Network Adapter
  - Repeaters
  - Bridges
  - Switches
  - Routers
  - Firewalls
- **Software**: This is what actually makes computer networks– not the hardware!
  - Protocols: describe *how* two communicating parties exchange information.
  - Services: describe *what* a network offers to parties that want to communicate.
  - Interfaces: describe how a client can make use of network services, i.e. how the services can be accessed.

Internet As an Example

PC

server

wireless laptop

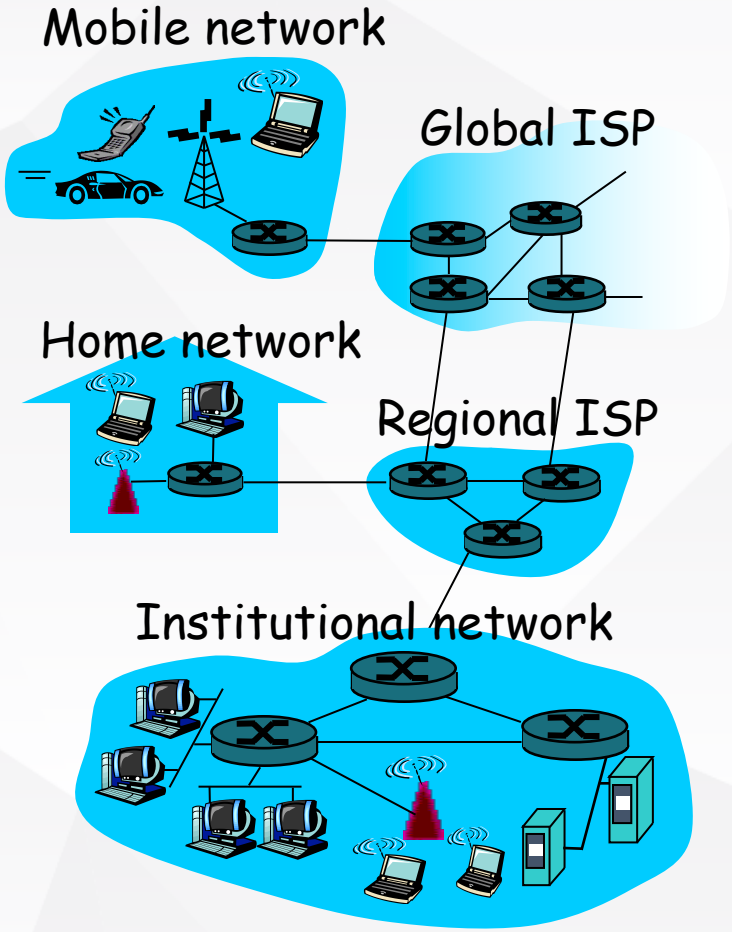
cellular handheld

access points

wired links

router

- *end systems:* millions of connected computing devices
- *communication links*
  - fiber, copper, radio, satellite
- *routers:* forward packets (chunks of data)



software

- *protocols* control sending, receiving of msgs

Chapter 1: roadmap

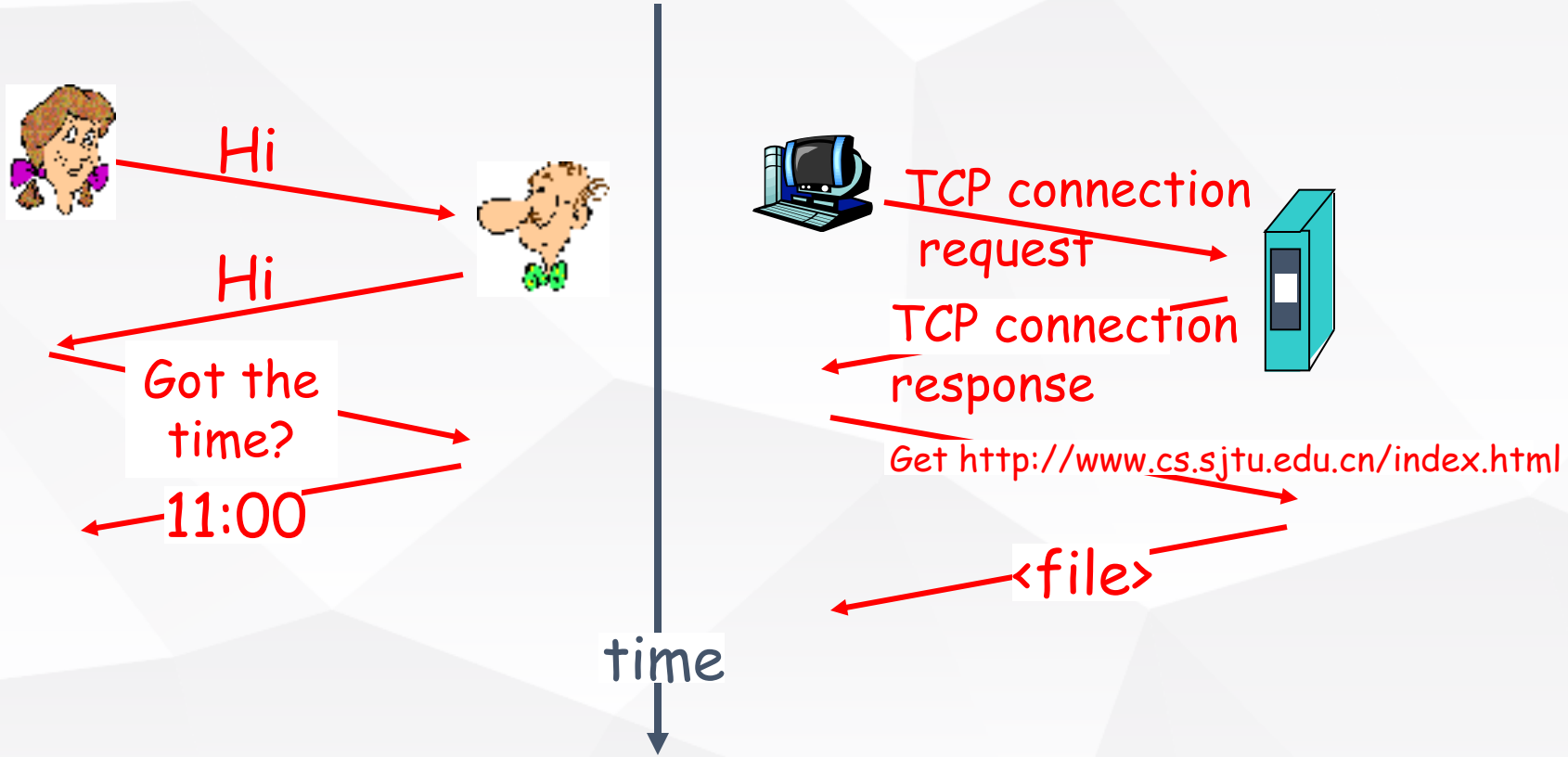
- What's Computer Network?
- Protocol layers, service models
- basic concepts of data transmission:
  - bandwidth, delay, throughput, multiplexing, switching
- What's the Internet?
- network edge:
  - hosts, access net, physical media
- network core:
  - packet/circuit switching, Internet structure

The Need for Protocols

- Basic communication hardware consists of mechanisms that can transfer bits from one point to another. (cumbersome and inconvenient)
- Application programs that use a network don't interact directly with network hardware. Instead, they interact with protocol software that follows the rules of a given protocol.
- An agreement that specifies the format, meaning of and actions taken on messages exchanged is known as a communication protocol, which handles most low-level communication details.

What's a protocol?

- a human protocol vs. a computer network protocol



Protocols are complex

Networks are complex!

- many “pieces”:
  - hosts
  - devices
  - switches
  - routers
  - links of various media
  - applications

Communication are complex!

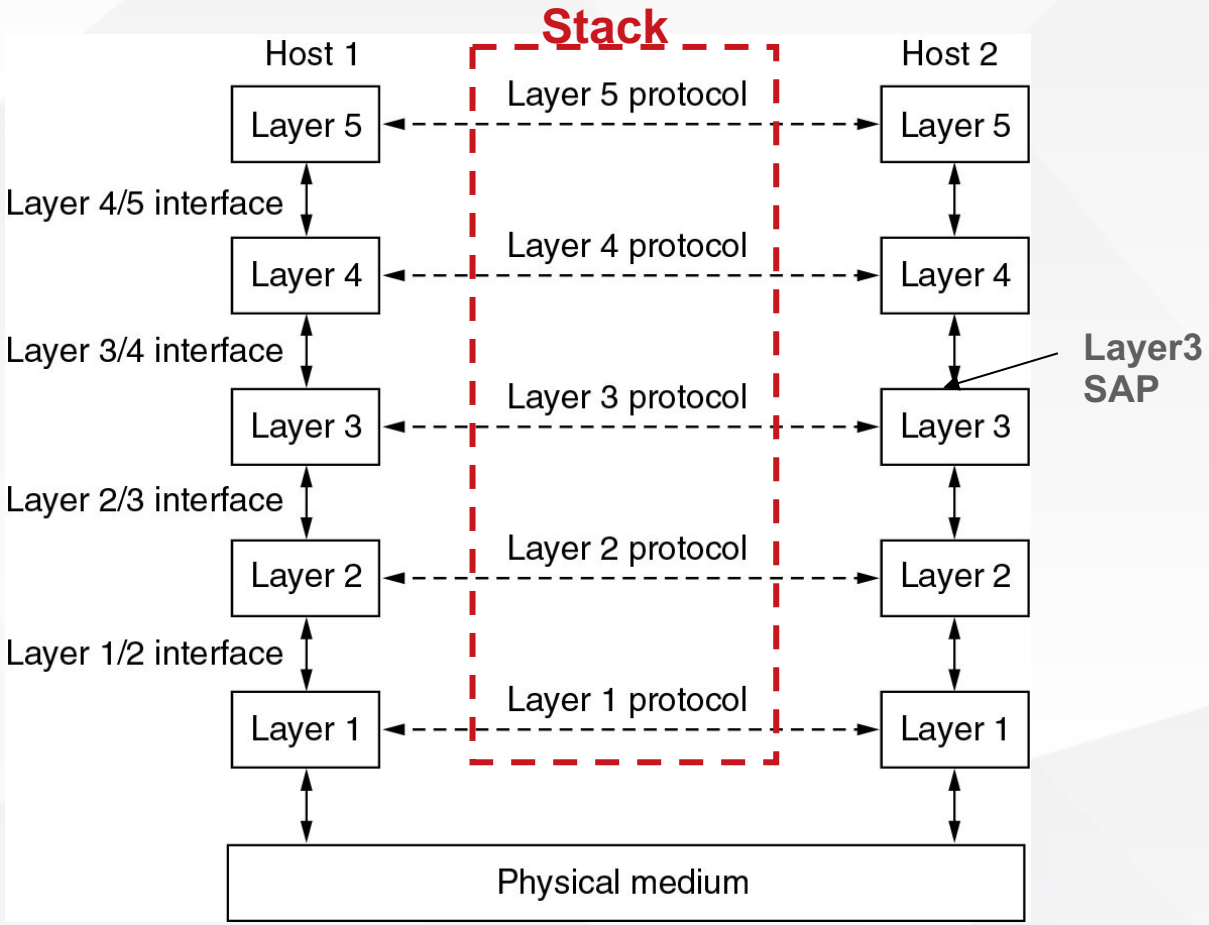
- many “tasks”:
  - data encoding,
  - transportation,
  - addressing,
  - error control,
  - flow control,
  - congestion control,
  - media access control

Why Protocol Layering

- Most network software are organized as a stack of layers or levels, each one built upon the one below it.
  - To reduce design complexity, divide the communication problem into subpieces and to design a separate protocol for each subpiece, making each protocol easier to design, analyze and implement.
  - Independence. Each layer could be designed, maintained and updated independently, as long as keep in mind the services the lower layer provides for it and the services it should provides for the upper layer.
  - Flexibility. Allow subsets of protocols be used as needed and allow any one of the protocols be replaced or updated.

Layering Model

- Layers, peers, protocols, services, interfaces and stack.



Concepts of Layering

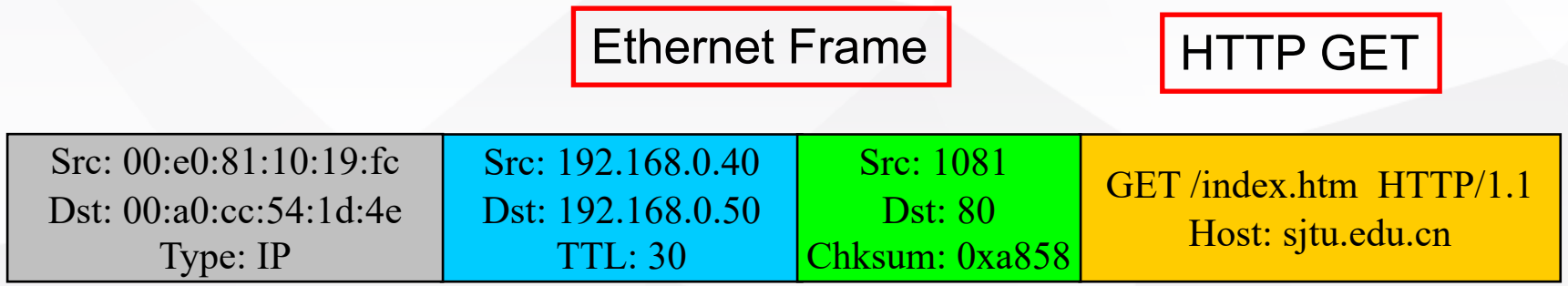
- **Protocol:** two parties at different sites, but at the same level (peers), always agree on how they will exchange information.
- In order for one party to send and receive information, it can only make use of the **communication services** offered by the layer directly underneath it.
- Services offered by a layer are always fully specified in terms of an **interface** that makes those services accessible.

How Layered Software Works

Encapsulation: multiple, nested Headers

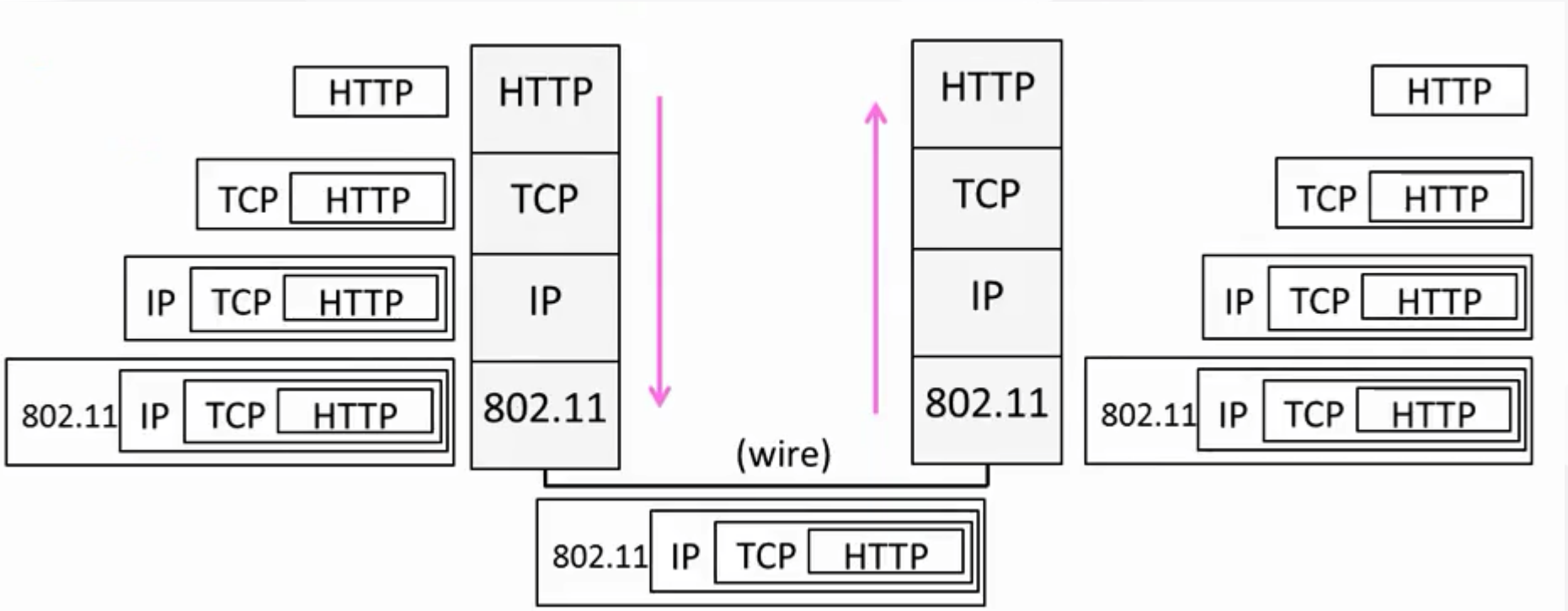
- Protocol software in a given layer on the sending computer adds information (header) to the outgoing data, and software in the same layer on the receiving computer uses the header to process incoming data.
- Outgoing data passes down through each layer, with headers added, and incoming data passes up through each layer, with headers verified and removed.

Demo



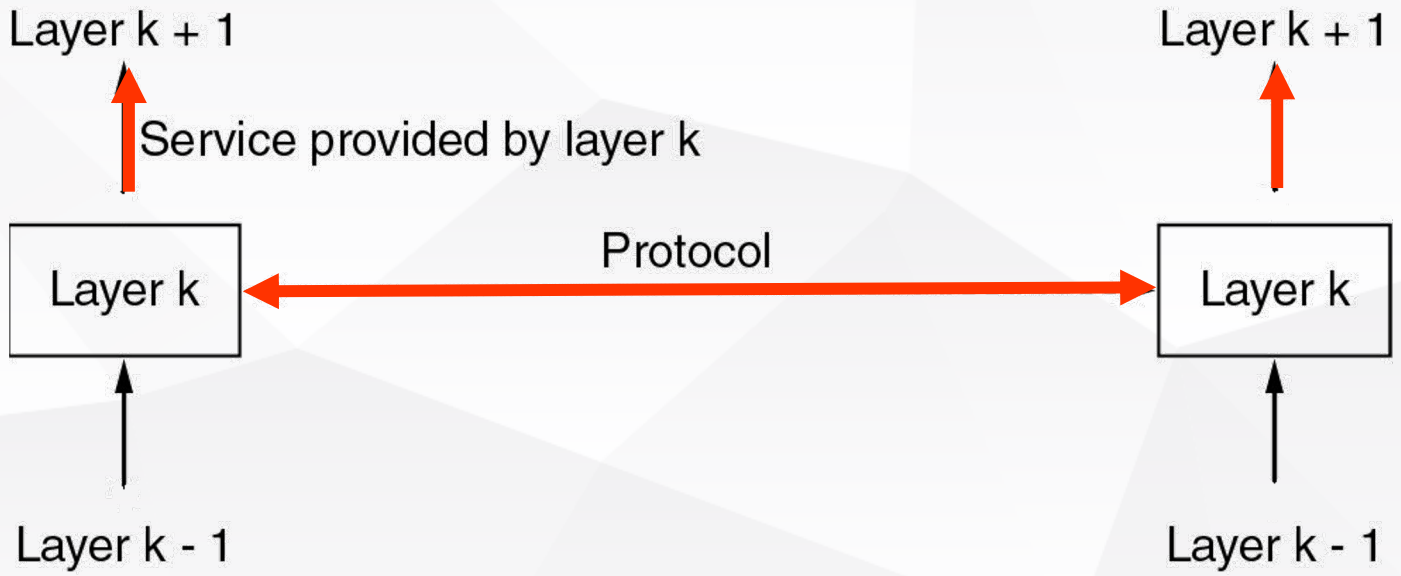
How Layered Software Works

- Encapsulation: multiple, nested Headers



Service Model

- Service are set of primitives that a layer provides to the layer above it (interfaces).
- Protocols are used to implement their service definitions.
- The service and the protocol are completely decoupled



Services: Connection-oriented and Reliable

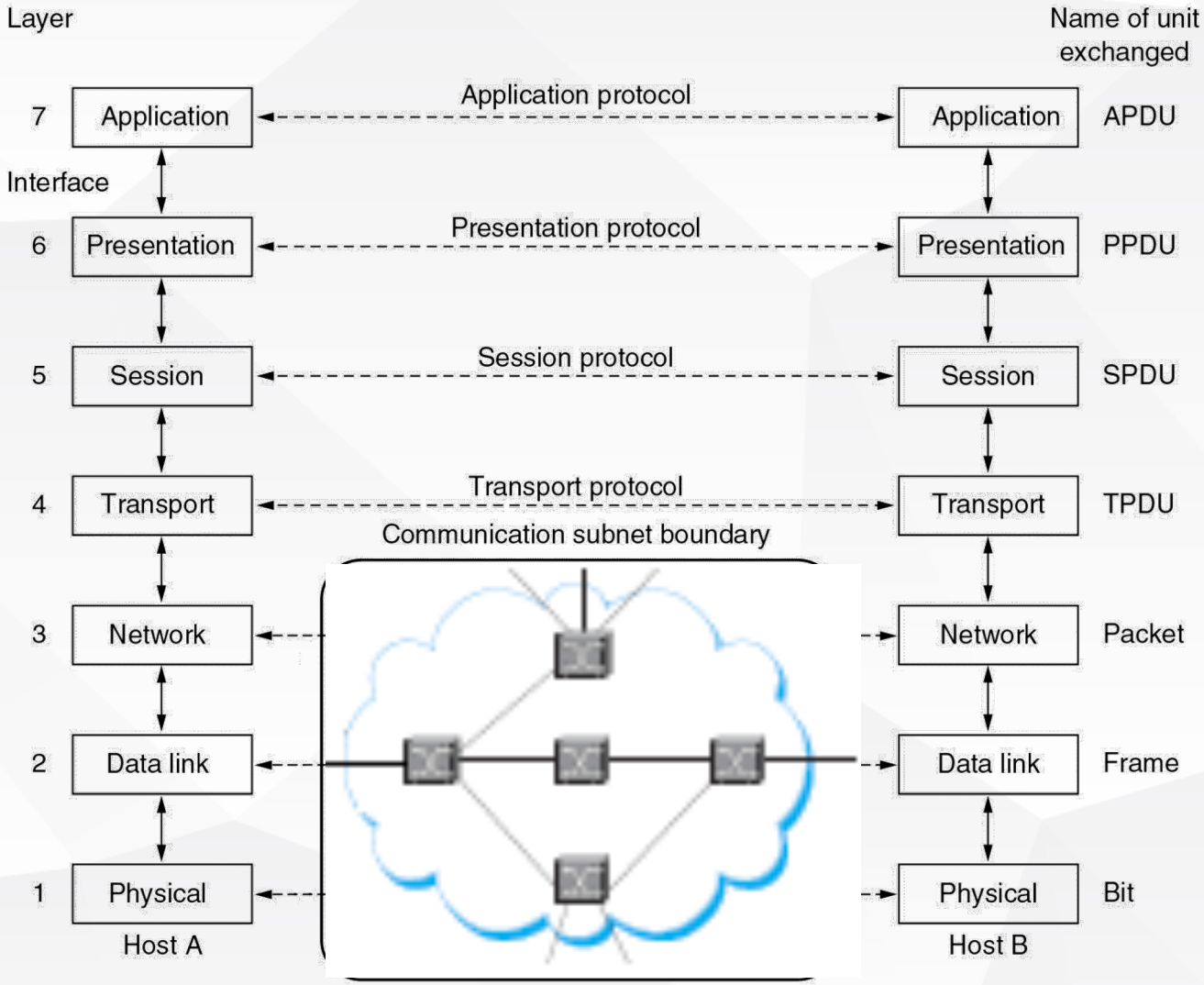
- **Connection-oriented:** This is the telephone model: you first **establish** a connection, then do a lot communication. Postal service is connectionless.
- **Reliable** : no bit error, no data loss, in order. Reliable service is implemented by having the receiver acknowledge the receipt of each message. Performance may degrade.

Service	Example	Connection
Reliable connection	TCP (www, email)	Connection-Oriented
Unreliable connection	Voice over IP	Connection-Oriented
Reliable datagram	Registered mail	Connectionless
Unreliable datagram	IP, UDP (DNS)	Connectionless

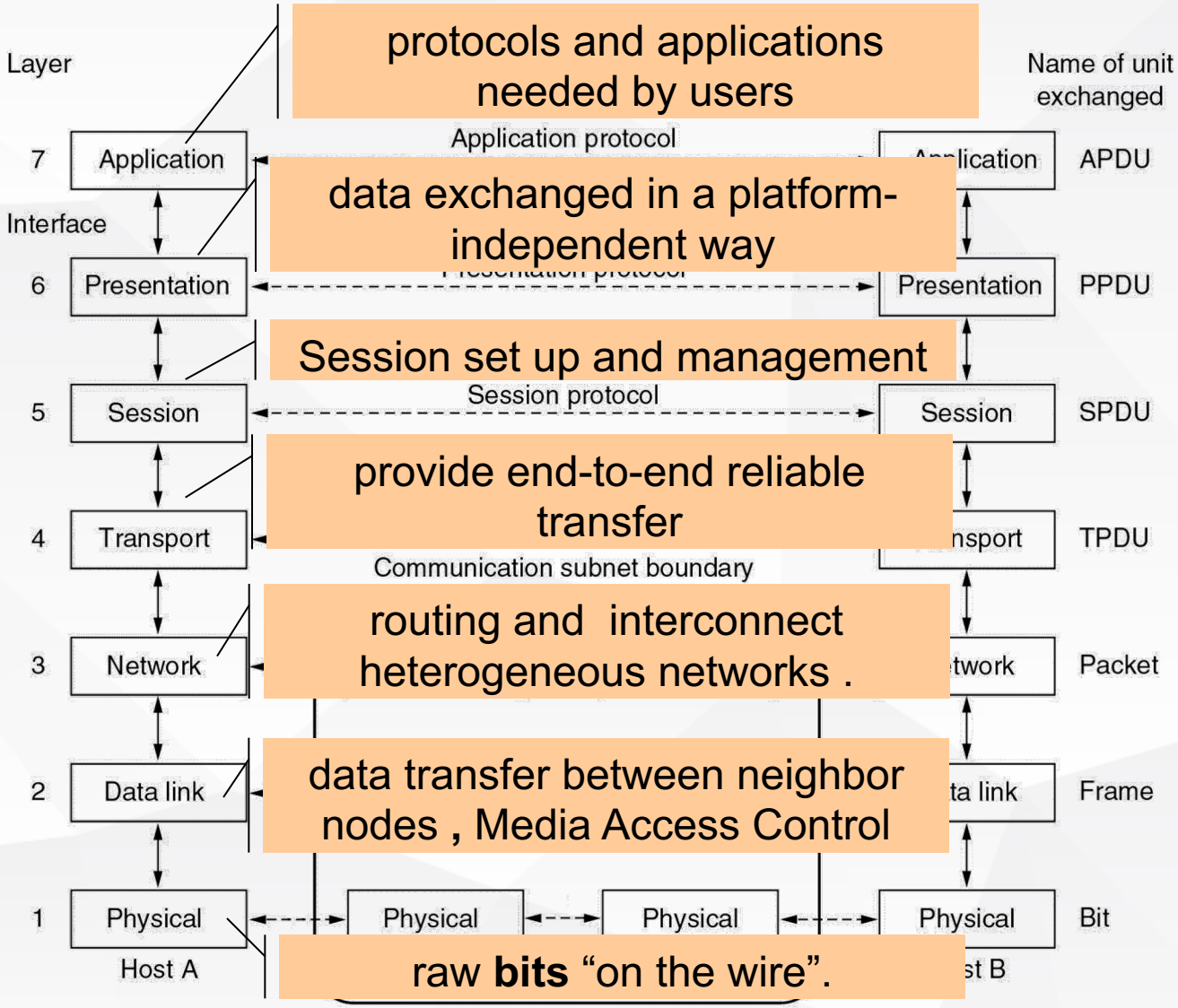
Layering Reference Models

- The OSI Reference Model
- The TCP/IP Reference Model
- A Comparison of OSI and TCP/IP

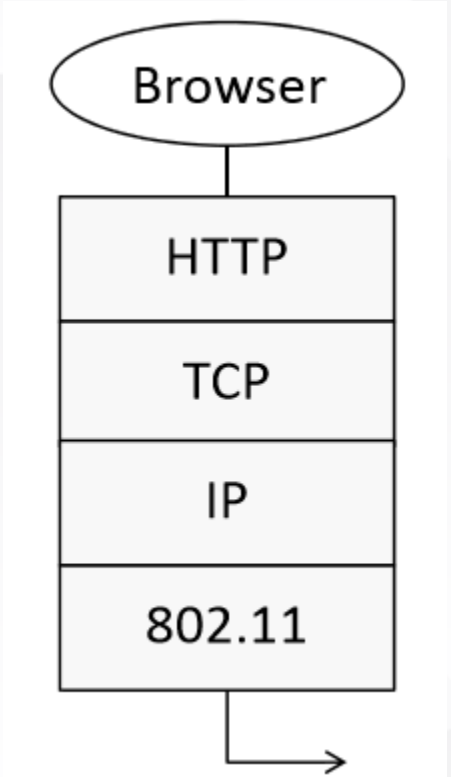
The ISO OSI 7-layer Reference Model

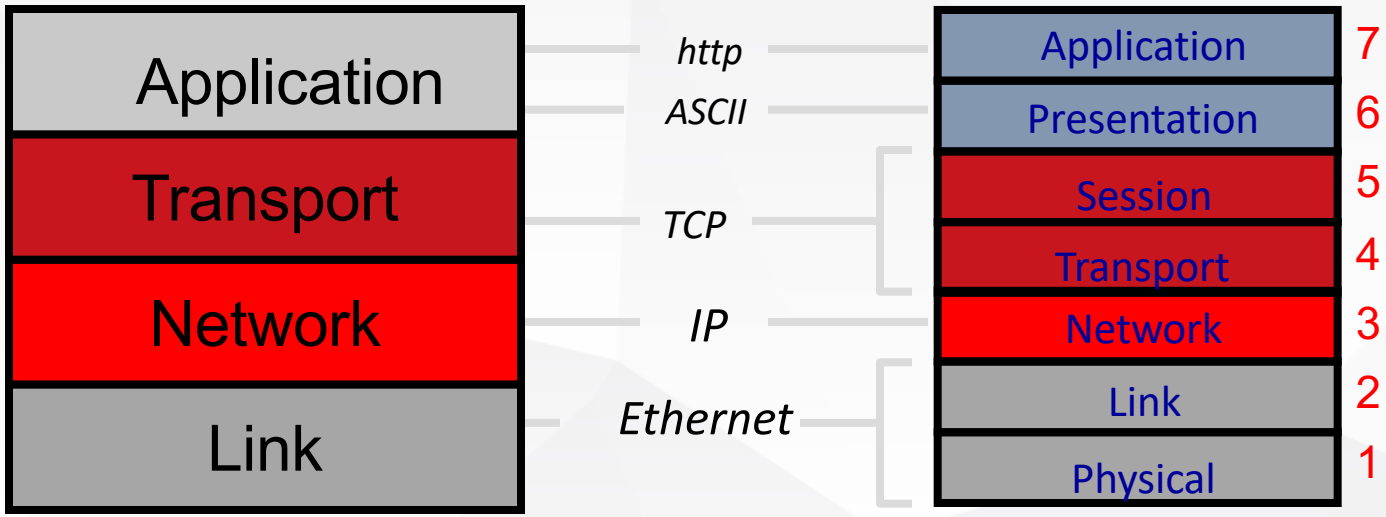


The ISO OSI 7-layer Reference Model



The 4-layer TCP/IP Model





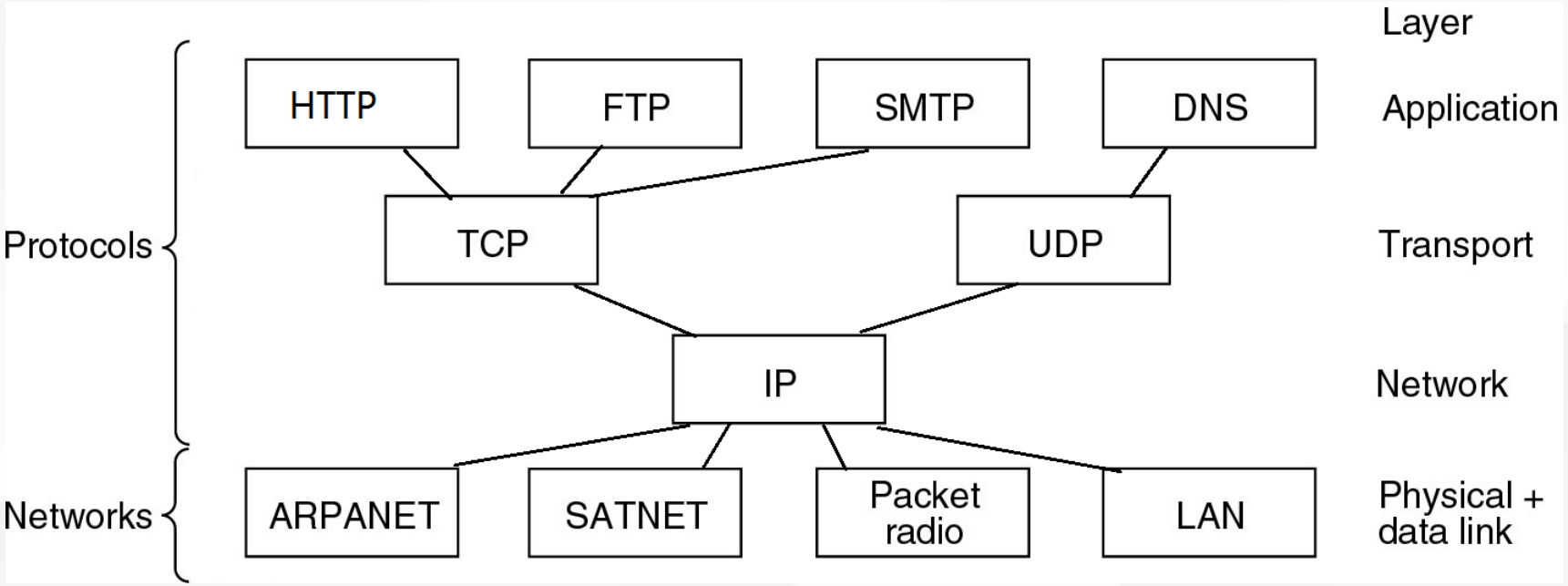
The 7-layer OSI Model

IETF RFCs

ISO standards

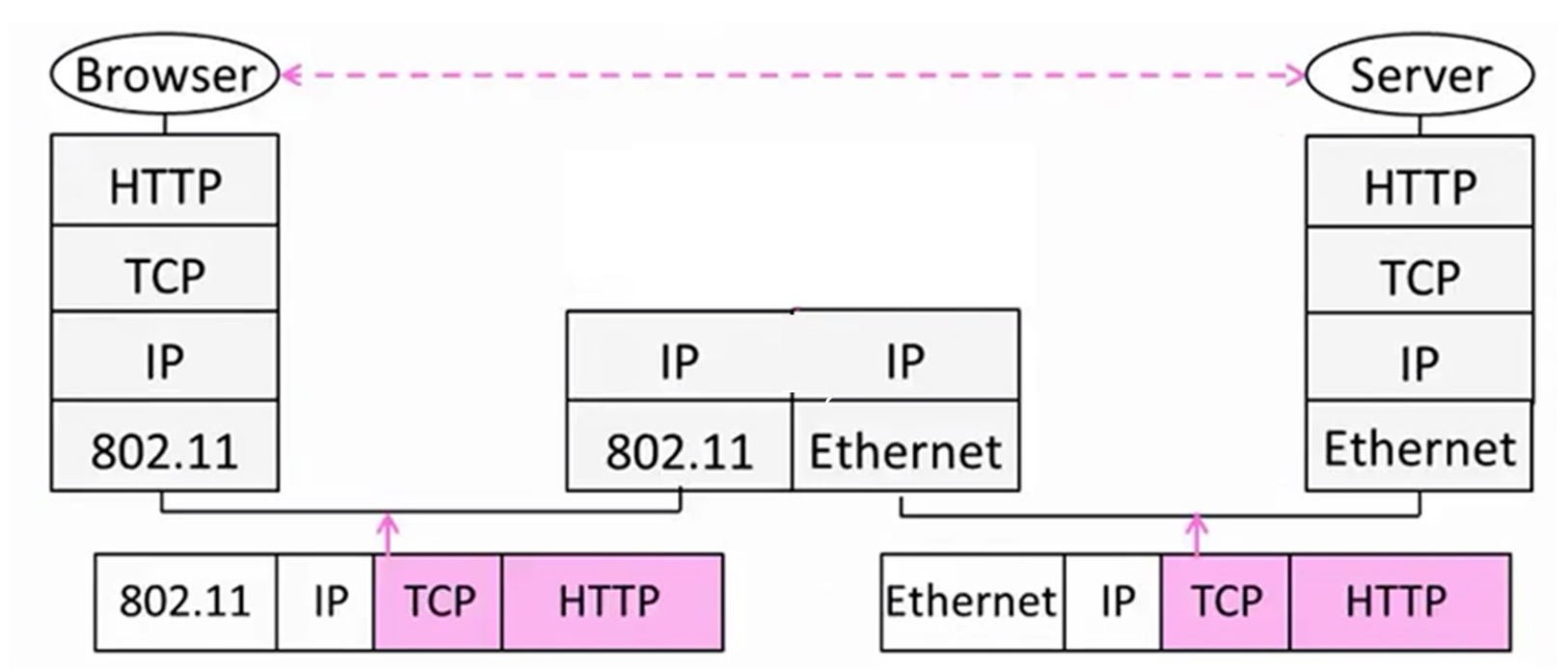
“Hourglass” philosophy of Internet

IP bridges different applications over different networks.



“Hourglass” philosophy of Internet

IP bridges different applications over different networks.



Comparing OSI and TCP/IP Models

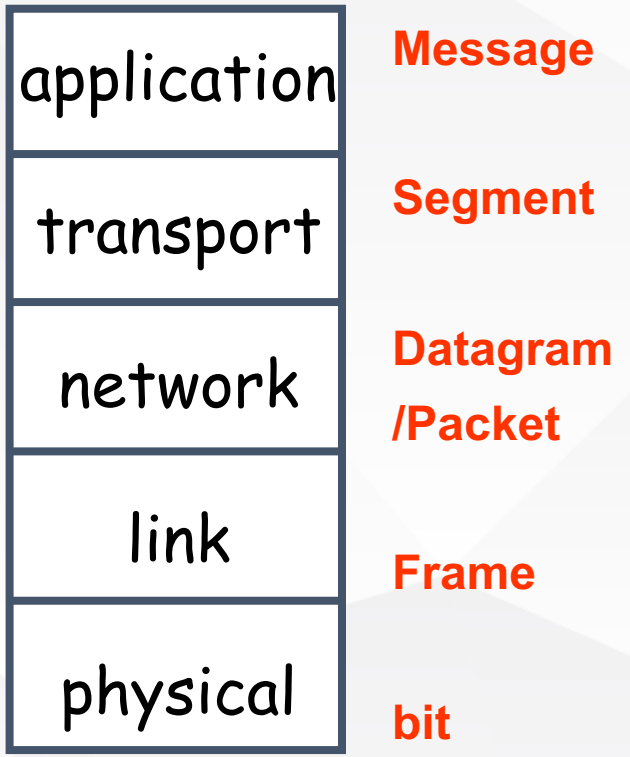
- Much in common:
  - a stack of independent protocols
  - functionality of the layers is roughly similar
- Many differences:
  - Services, interfaces and protocols are central concepts of the OSI model, TCP/IP model doesn't distinguish these concepts and is not a general model.
  - OSI model was devised before the corresponding protocols, with TCP/IP the reverse was true.
  - OSI model/protocols took too much time and are too complex, while TCP/IP is simple and not so comprehensive.
  - Number of layers
- OSI model has proven to be exceptionally useful for discussing computer networks, OSI protocols have not become popular.
- TCP/IP Model is practically nonexistent, but protocols are widely used, deeply entrenched, and thus hard to replace.

Who takes over the world?

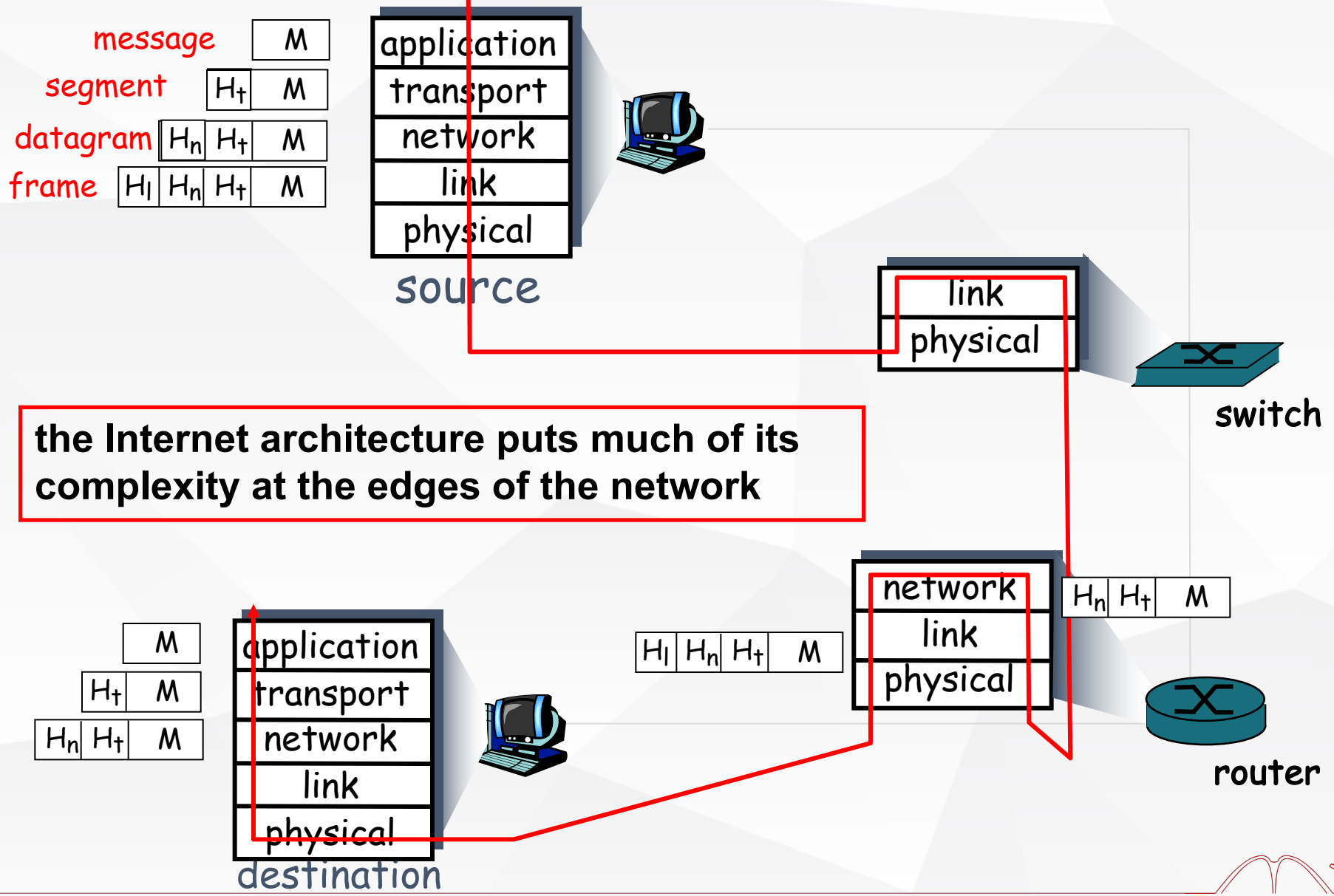
- Why OSI not
  - Bad timing
  - Bad technology
  - Bad implementations
- Why TCP/IP protocol suite is successful
  - it was there when needed
  - freely distributed with the UNIX operating system

Hybrid Model Used in this Course

- application: programs using network services (http, ftp, smtp)
- transport: end-end reliable data transfer (tcp, udp)
- network: send packets over multiple networks (ip, routing algorithms)
- link: data transfer between neighboring network nodes (Ethernet, WiFi, ppp, MAC)
- physical: send bits as signals “on the wire” (media, modulation,encoding)



Protocol Walkthrough



Internet architecture design

- **Layering principle**: simple and flexible
- **Packet switching:** suitable for bursty data, store-and-forward, statistical multiplexing
- **"Hourglass" philosophy**: IP is the glue
- **End-to-End principle**: Whenever possible, put the functionality at the endpoint.
  - Minimal assumption about the network, best-effort service, stateless
  - TCP/IP
  - Exceptions: wireless link layer error control
- References
  - **"END-TO-END ARGUMENTS IN SYSTEM DESIGN",** J.H. Saltzer, D.P. Reed and D.D. Clark, MIT, 1984
  - **"The Design Philosophy of the DARPA Internet Protocols",** D.D. Clark, MIT, SIGCOMM 1988
  - "**Architectural Considerations for a New Generation of Protocols**", D. D. Clark, D. L. Tennenhouse, MIT 1990

Internet architecture goals (Clark88)

- Main goal: multiplexed utilization of existing interconnected networks
- Secondary goals
  - survivability in the face of failure
  - support for multiple types of service
  - support for a variety of networks
  - distributed management of resources
  - cost effectiveness
  - easy addition of new hosts
  - accounting of resource usage