UIC UNIVERSITY OF ILLINOIS AT CHICAGO Department of Computer Science COLLEGE OF ENGINEERING

CS450 – Introduction to Networking

Lecture 3 – Internet layers and Socket Programming

Phu Phung

January 16, 2015

Peer discussion groups

Matthew Borgelt Dustin Kimble Jaime Ramirez	Ashley Riley Sunil Rao Sorenson, Chris	Adrian Campos Ian Swift Maravelias, Joshua	Andriy Klyuka Mikhail Rogozhin Nakul Saunshi
James Angelakos Tianniu Lei Analil Mamucod	Michael Albanese Steven Gliniewicz Nikhil Vellala	Zhengyu Hao Matthew Walsh Zhongkai Wen	George Maratos Matthew Patterson Mateusz Wszolek
Davison, Erik N. Kehinde, Adebayo Schlake, Alexander	Ortega, Carlos Sander, Alex	Hao, Zhengyu Khajanji, Piyush Momeni Milajerdi, Sedegh	Neelakantan, Janani Ravi Anand Prasad, Satish Anand

Quiz: What is the Internet?

A. A protocol to connect computers

B. A standard for computer to connect to a network

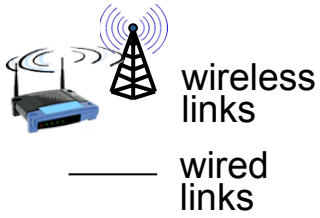
C. World Wide Web

D. Computers that are all linked together in a network

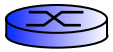
E. All of the above

What's the Internet: "nuts and bolts" view

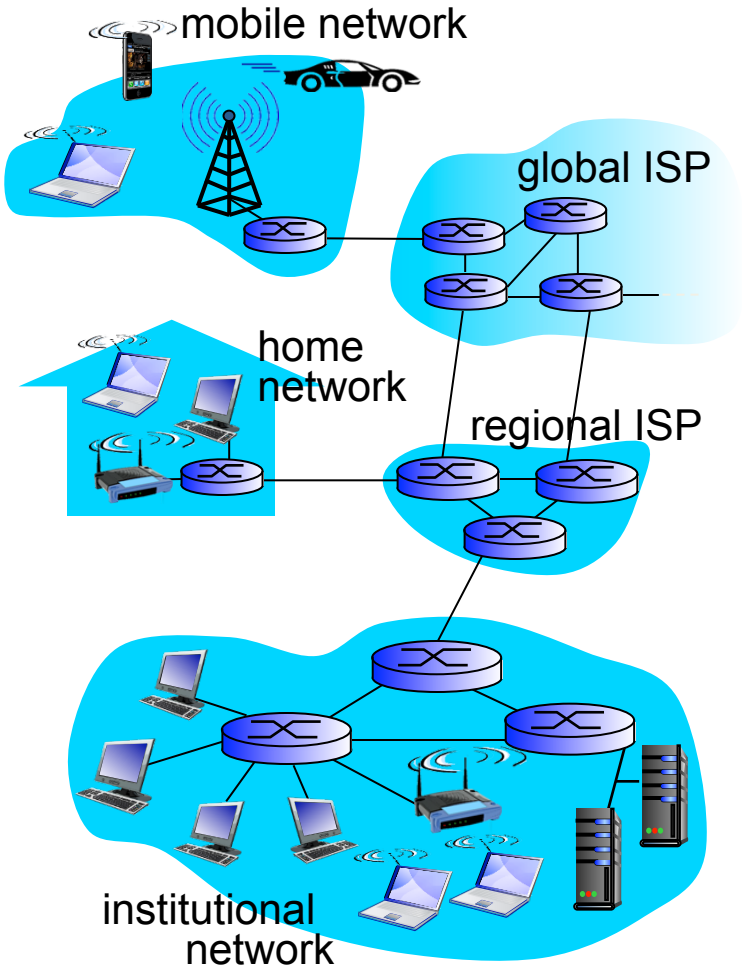
- millions of connected computing devices:
  - *hosts* = *end systems*
  - running *network apps*



- *communication links*
  - fiber, copper, radio, satellite
  - transmission rate: *bandwidth*



- *Packet switches:* forward packets (chunks of data)
  - *routers* and *switches*



“Fun” internet appliances

IP picture frame
http://www.ceiva.com/

Web-enabled toaster +
weather forecaster

Tweet-a-watt:
monitor energy use

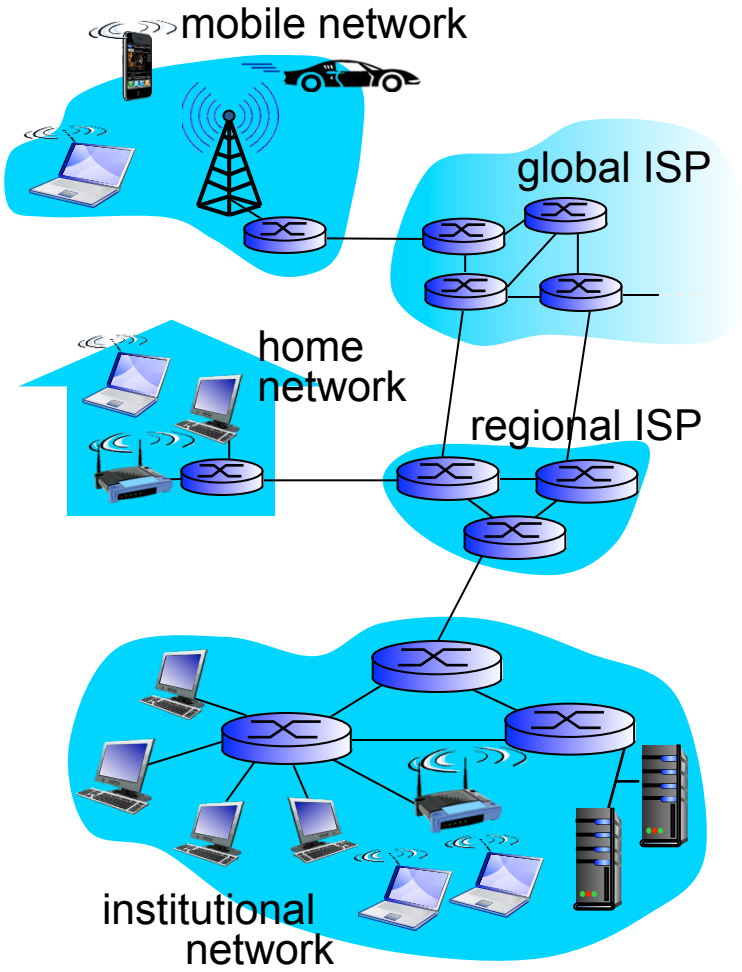
Internet
refrigerator

Slingbox: watch,
control cable TV remotely

Internet phones

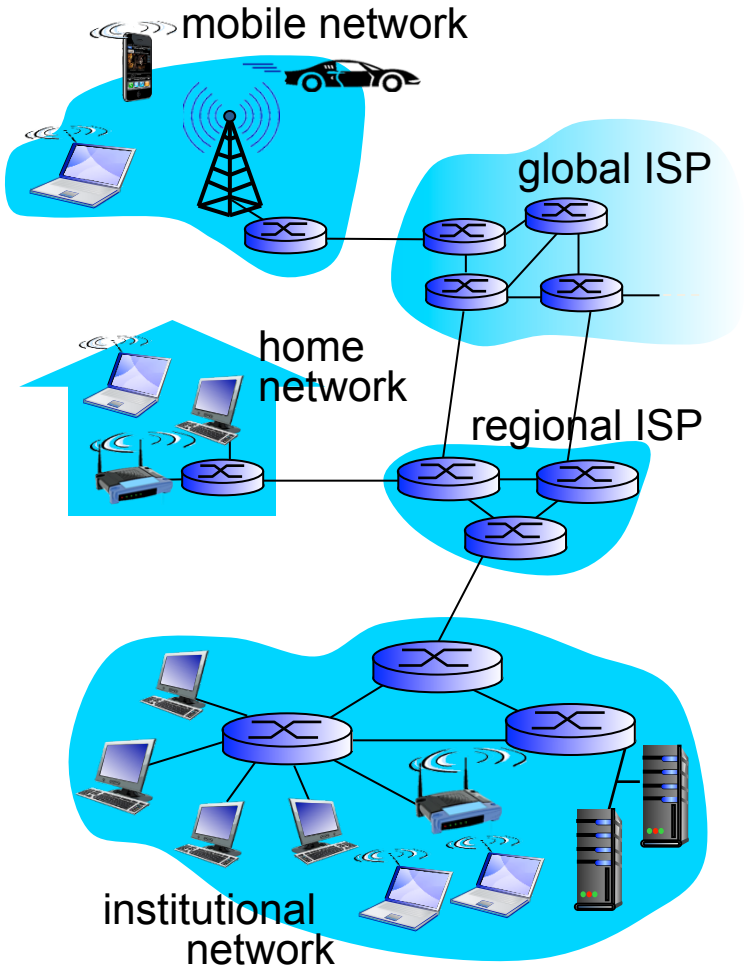
What's the Internet: "nuts and bolts" view

- *Internet:* "network of networks"
  - Interconnected ISPs
- *protocols* control sending, receiving of msgs
  - e.g., TCP, IP, HTTP, Skype, 802.11
- *Internet standards*
  - RFC: Request for comments
  - IETF: Internet Engineering Task Force



What's the Internet: a service view

- *Infrastructure that provides services to applications:*
  - Web, VoIP, email, games, e-commerce, social nets, …
- *provides programming interface to apps*
  - hooks that allow sending and receiving app programs to "connect" to Internet
  - provides service options, analogous to postal service



What's a protocol?

human protocols:

- "what's the time?"
- "I have a question"
- introductions

… specific msgs sent

… specific actions taken when msgs received, or other events

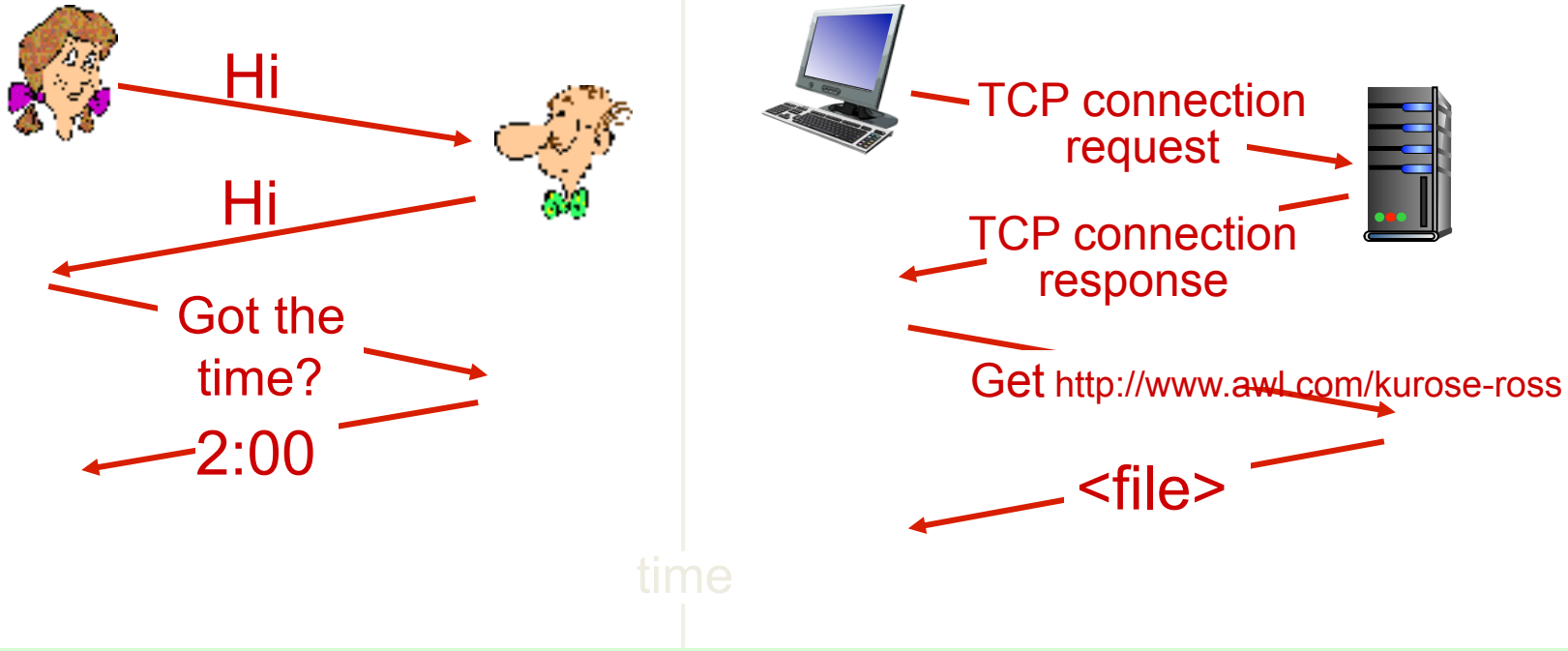
network protocols:

- machines rather than humans
- all communication activity in Internet governed by protocols

protocols define format, order of msgs sent and received among network entities, and actions taken on msg transmission, receipt

What's a protocol?

a human protocol and a computer network protocol:



Protocol: *agreed message format and transfer procedure between sender and receiver*

Protocol "layers"

Networks are complex, with many "pieces":

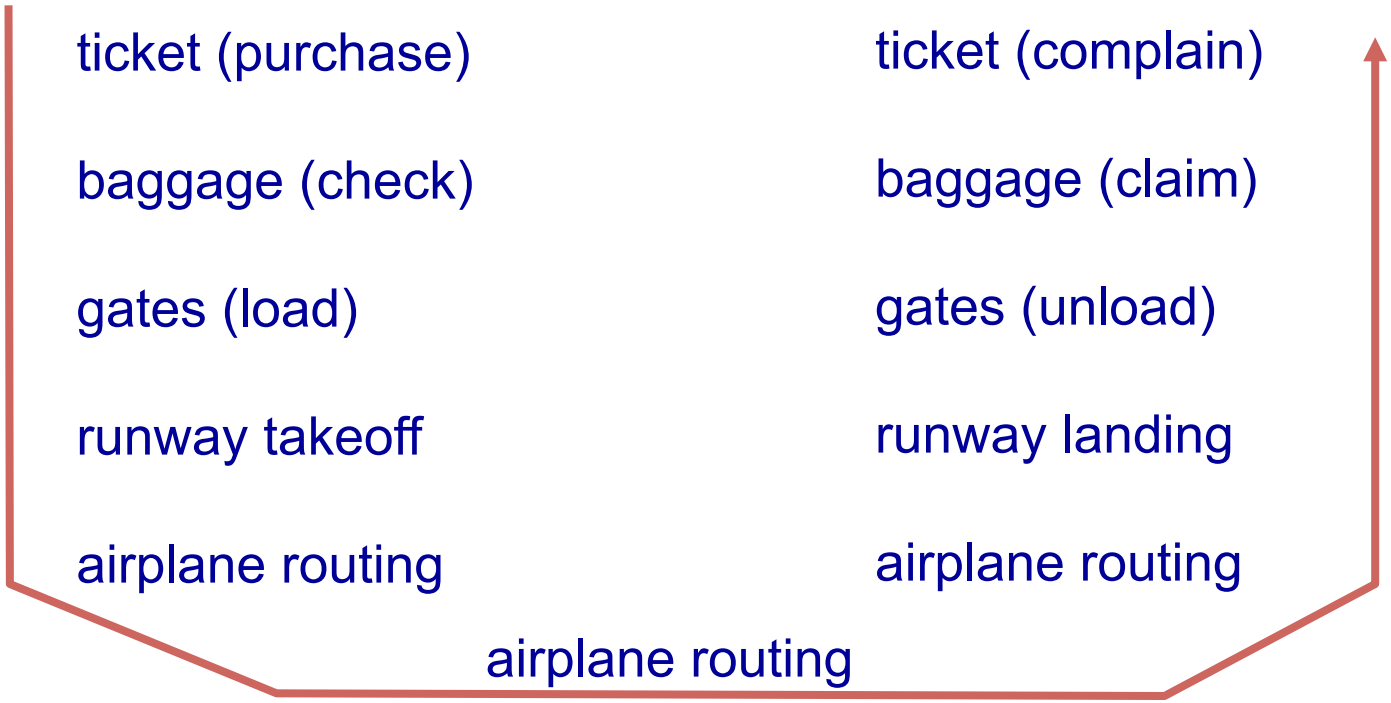
- hosts
- routers
- links of various media
- applications
- protocols
- hardware, software

Question:

is there any hope of *organizing* structure of network?

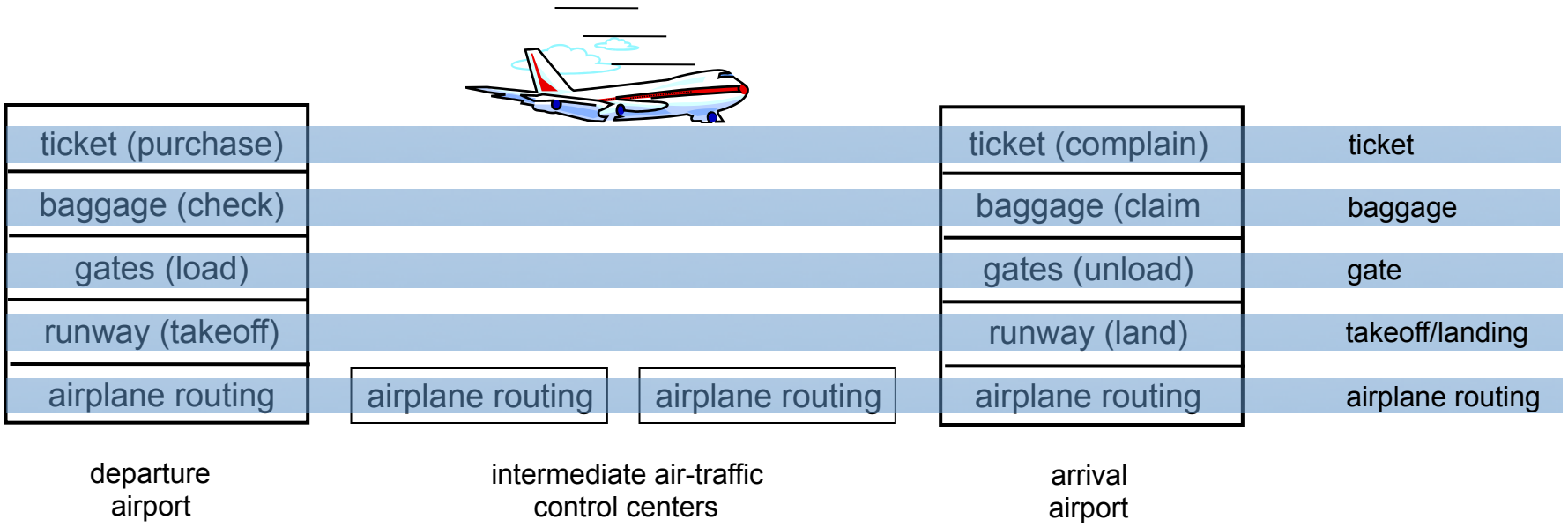
…. or at least our discussion of networks?

Organization of air travel



- a series of steps

Layering of airline functionality



layers: each layer implements a service

- via its own internal-layer actions
- relying on services provided by layer below

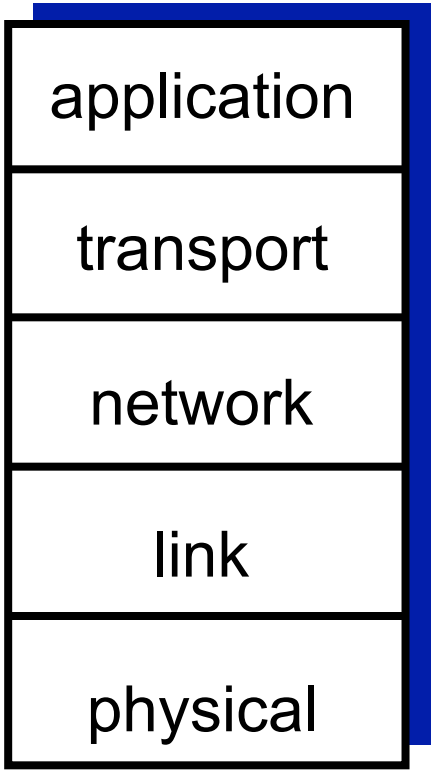
Why layering?

dealing with complex systems:

- explicit structure allows identification, relationship of complex system's pieces
  - layered *reference model* for discussion
- modularization eases maintenance, updating of system
  - change of implementation of layer's service transparent to rest of system
  - e.g., change in gate procedure doesn't affect rest of system

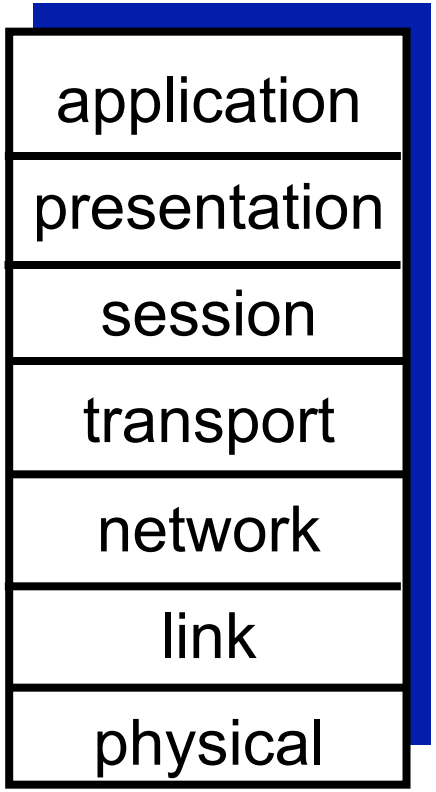
Internet protocol stack

- *application:* supporting network applications
  - FTP, SMTP, HTTP
- *transport:* process-process data transfer
  - TCP, UDP
- *network:* routing of datagrams from source to destination
  - IP, routing protocols
- *link:* data transfer between neighboring network elements
  - Ethernet, 802.111 (WiFi), PPP
- *physical:* bits "on the wire"

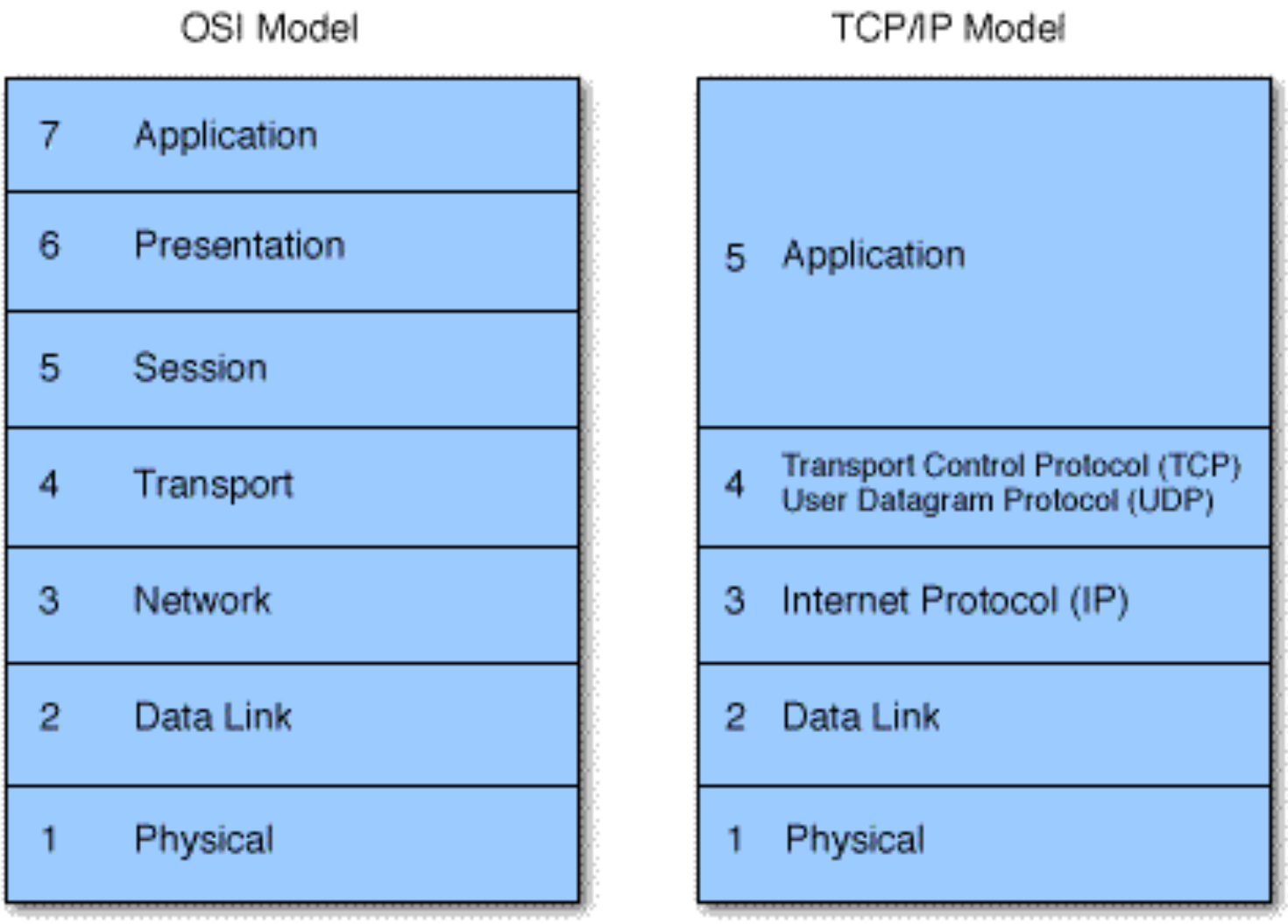


ISO/OSI reference model

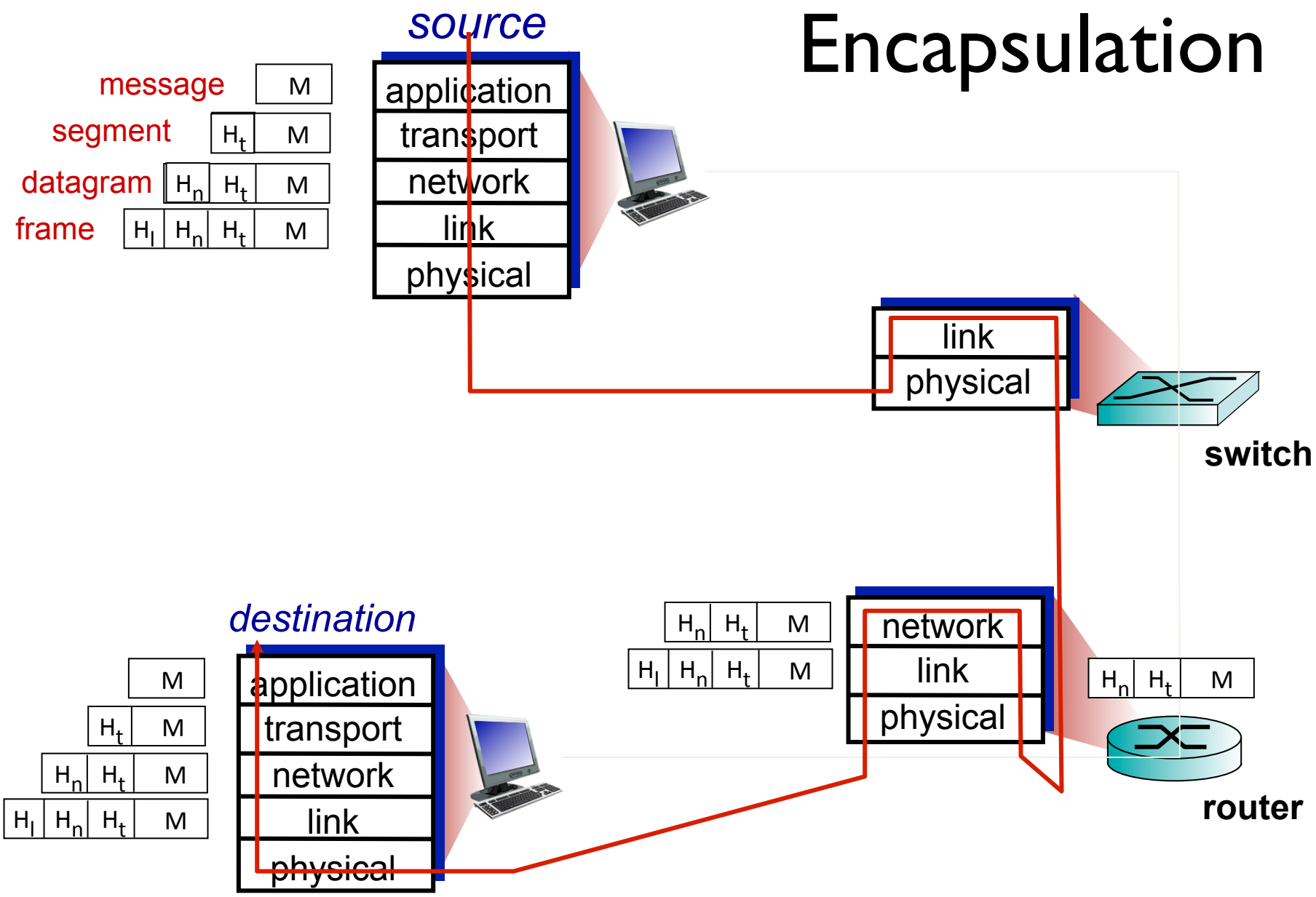
- *presentation:* allow applications to interpret meaning of data, e.g., encryption, compression, machine-specific conventions
- *session:* synchronization, checkpointing, recovery of data exchange
- Internet stack "missing" these layers!
  - these services, *if needed,* must be implemented in application
  - needed?



OSI reference model vs TCP/IP model



Encapsulation



Which Layer in Internet model implementing the format of data e.g., encryption, compression…?

A. Application

B. Presentation

C. Session

D. Network

E. All layers

Which Layers Do Routers Need?

A. Application, Transport

B. Network, Link and Physical

C. Link and Physical

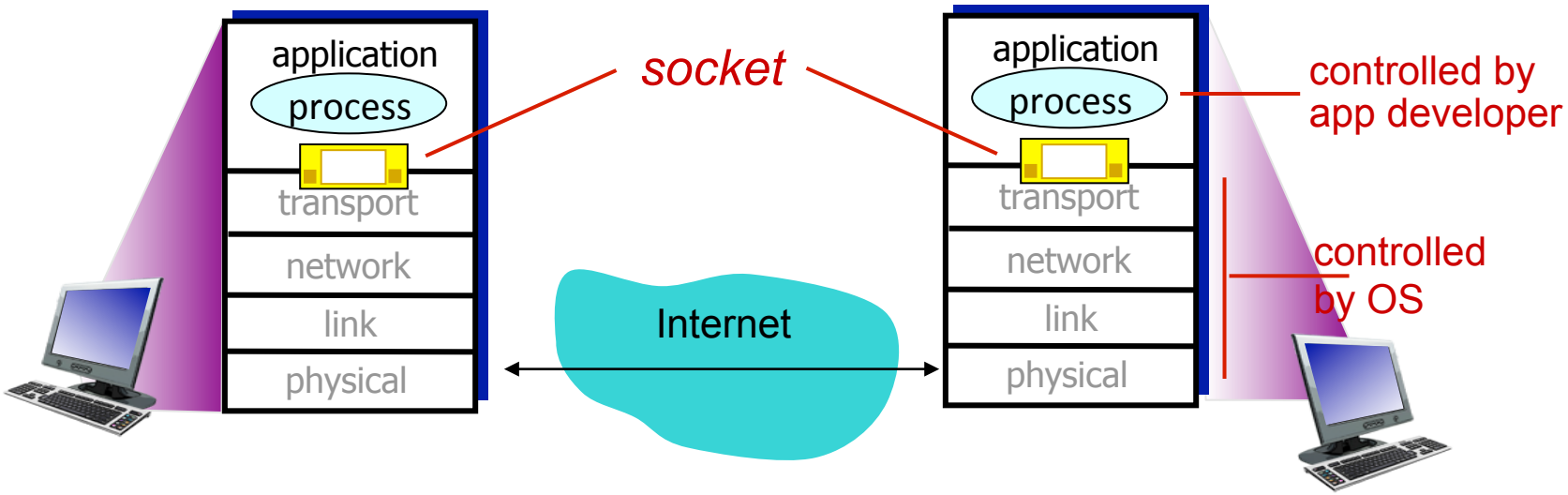
D. Application through Network

E. All layers

Socket programming

goal: learn how to build client/server applications that communicate using sockets

socket: door between application process and end-end-transport protocol



Socket programming

Two socket types for two transport services:

- *UDP:* unreliable datagram
- *TCP:* reliable, byte stream-oriented

Application Example (Assignment 1):

1. Client connects and sends the HTTP request to the server.
2. The server receives the HTTP request and get the resource to construct HTTP response.
3. The server sends the HTTP response to the client.
4. The client receives the data and handles the data.

Socket programming *with TCP*

client must contact server

- server process must first be running
- server must have created socket (door) that welcomes client's contact

client contacts server by:

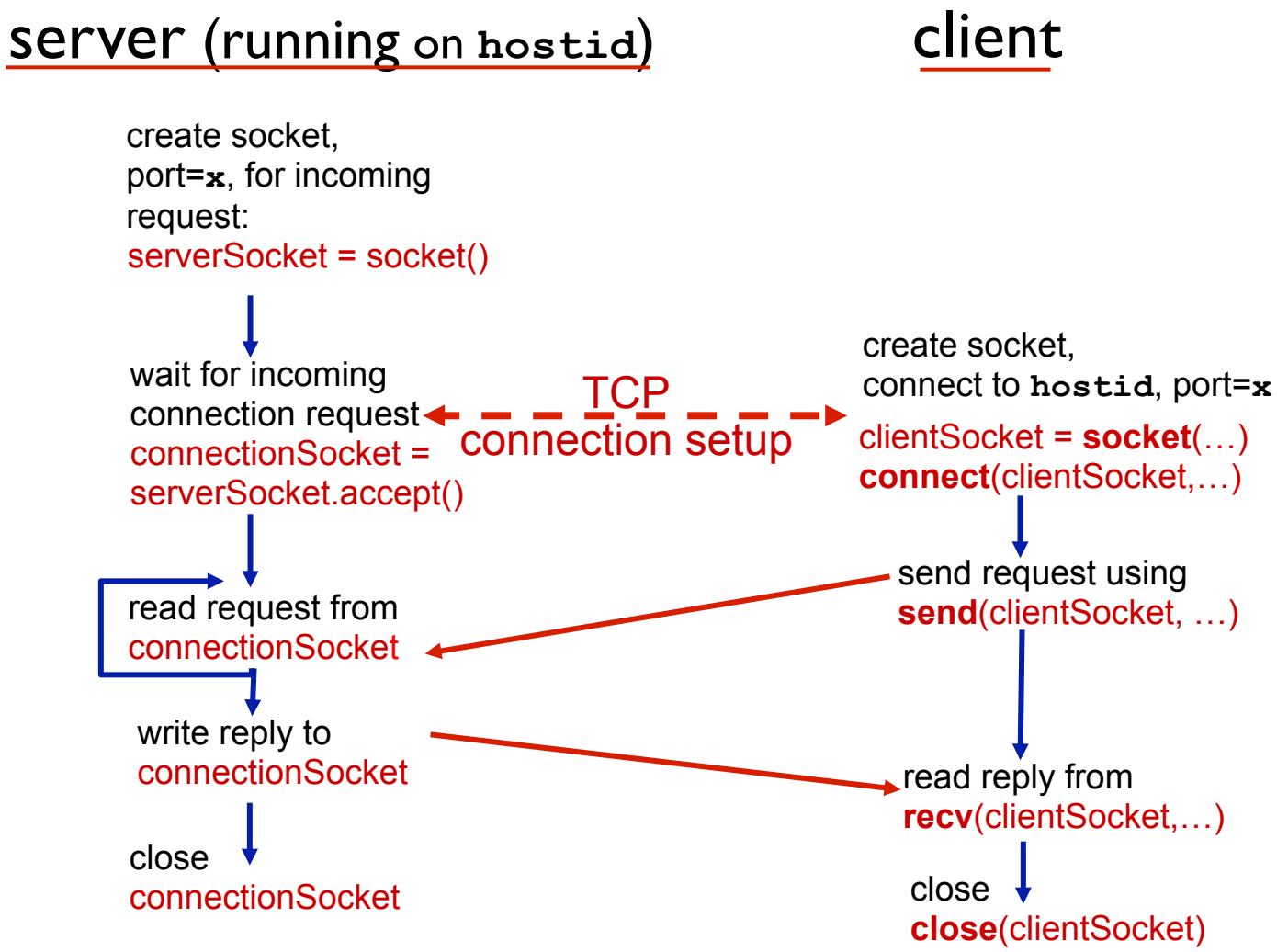
- Creating TCP socket, specifying IP address, port number of server process
- *when client creates socket:* client TCP establishes connection to server TCP

- when contacted by client, *server TCP creates new socket* for server process to communicate with that particular client
  - allows server to talk with multiple clients
  - source port numbers used to distinguish clients (more in Chap 3)

application viewpoint:

TCP provides reliable, in-order byte-stream transfer ("pipe") between client and server

Client/server socket interaction: TCP



Beej's Guide to Network Programming

http://beej.us/guide/bgnet/output/html/singlepage/bgnet.html

Next lecture (Wed Jan 21)

- Application layer and HTTP protocol
  - Required readings 2.2.0 - 2.2.4