

# ICS 451: Data Networks

**Edo Biagioni**

`esb@hawaii.edu`

Department of Information and Computer  
Sciences

# ICS 451: Today's plan

- Introductions
- Course overview
- What is networking?
- Clients and servers
- Internet service and sockets
- Connection-oriented and connectionless

# Introductions

# ICS 451 Course Overview: Management

esb@hawaii.edu

- <http://www2.hawaii.edu/~esb/2015fall.ics451/>
- Lectures are Tue/Thu 3pm in POST 127
- Textbook: Computer Networking, Principles, Protocols, and Practice, by Olivier Bonaventure (available free online)
- Assignments, 2 exams and a final, code reviews
- Office hours: Tue 5-6pm, Thu 2-2:45pm

# ICS 451 Course Overview: Teaching Assistant

- Jon Moroney
- [jmoroney@hawaii.edu](mailto:jmoroney@hawaii.edu)
- The TA grades the assignments
  - the instructor grades the exams
- Office hours: Mon and Wed 12-1, Fri 4-5
- Turn in assignments **on time!!!**
  - 20% per day penalty for late assignments

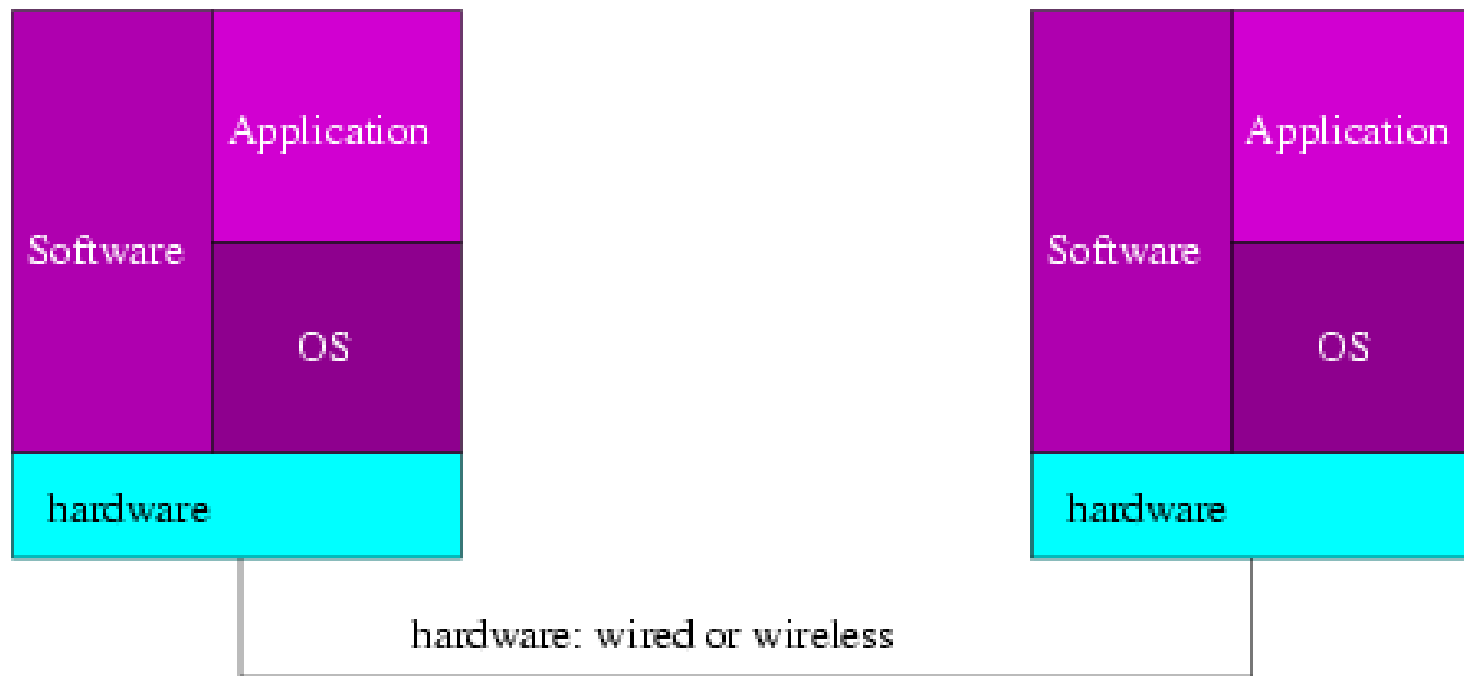
# ICS 451 Course Overview: Contents

- Networking overview
  - Applications: DNS, email, web
  - Internet Protocols: TCP and IP
  - Local Area Networks: Aloha, Ethernet, 802.11
- Programming Assignments:
- Sockets interface
  - Clients
  - Servers
  - Routers
  - Protocols

# What is Networking?

- Engineering, science, and art
- Designing and building
  - Computer networks, and
  - Networking protocols
- A computer network is
  - A hardware and software system that
  - Enables communication among computers

# Networking Hardware and Software





# Why is Networking Interesting?

- Jobs
- Networks are pervasive
  - Useful for science and technology (the Internet)
- Networks are complex systems
  - Distributed
  - Parallel
  - Interoperable
  - fast and efficient

# What is (not) in this course?

- A **network administrator** manages a network, adding and configuring equipment and deciding where to place additional money or equipment
- A **network designer** transforms a budget into a network design that will satisfy given constraints
- This course does teach the *theoretical foundations* that network administrators and network designers need
  - but does not teach network administration or network design

# What is (not) in this course?

- A **network application developer** creates applications distributed across the network
- A **network protocol designer** designs and develops protocols, agreements about how computers and programs communicate
- network protocol design is a part of designing distributed systems, systems that work across multiple computers

# What is in this course?

- This course teaches the basics of network application development and introduces students to network protocol design

# Why might you care?

- Apps for mobile devices
- Java and Javascript embedded in web pages
- Innovative web services
- Understanding what's "under the hood"

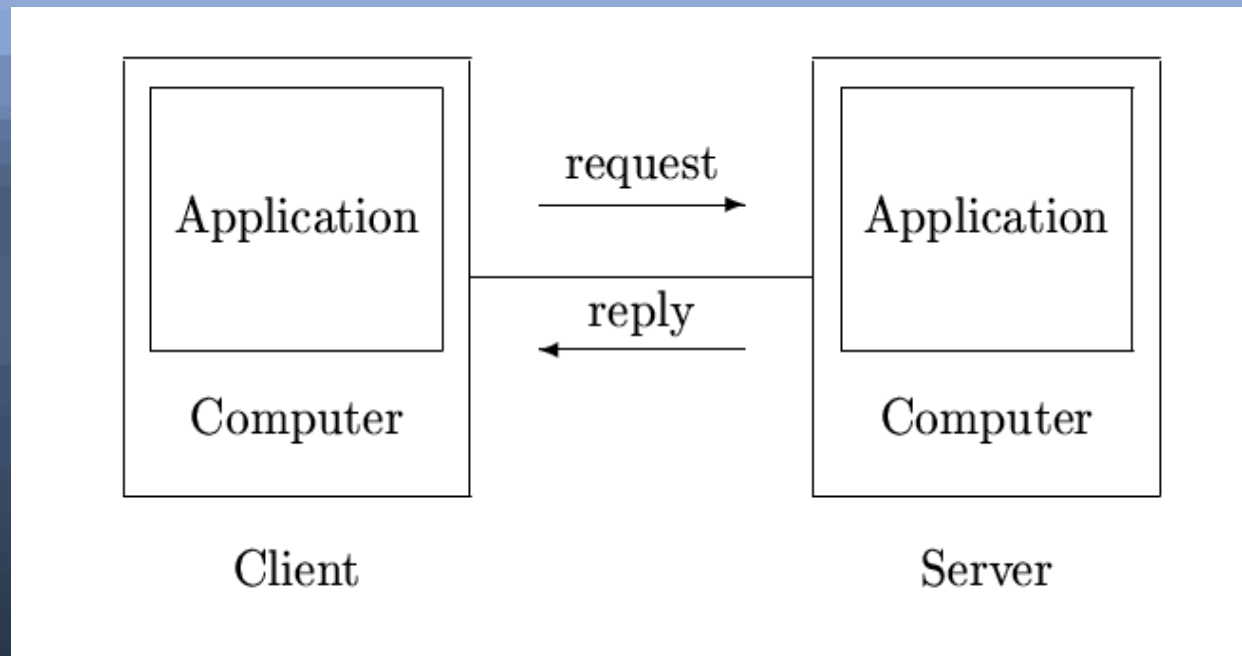
# Network Applications

- applications are programs that run on network-connected computers
- applications communicate by exchanging data
- data is encoded as a collection of bits, with specific meanings assigned to the bits
- applications must have ways of specifying which applications they are communicating with: **addresses, ports, and names**

# Clients and Servers

- A client is a program that requests a service
- A server is a program that provides a service
- clients and servers may or may not run on different machines
- the client-server **model** captures a common way that programs interact
  - Some programs may not match this model

# Client/Server Model

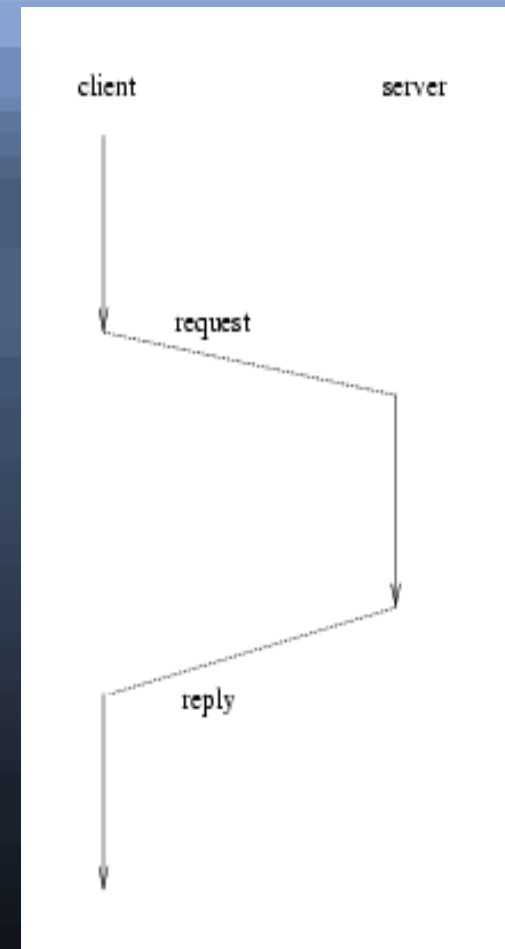


- Clients request services
- Servers reply to requests
- Data flows in both directions



# Client/Server Model

- Servers block waiting for requests
- Clients send requests, then block
- Servers receive the request, send replies
- Once the client receives the reply, the interaction is complete



# Connecting to a Server

- **port numbers** are used in the Internet to identify the service
- **IP numbers** and **domain names** are used in the Internet to identify the machine on which the server is running
- Domain names are familiar, e.g. [www.ietf.org](http://www.ietf.org)
- IP numbers come in two varieties:
  - IPv4, e.g. 4.31.198.44
  - IPv6, e.g. 2001:1900:3001:11::2c

# Connections vs Connectionless

- TCP expects a program to **connect** before sending or receiving any data
- At the end of the communication, the connection should be **shut down (closed)**
- With UDP, connections are not used:
  - each **send** or **receive** operation takes an address as parameter

# Sockets Interface

- Client calls **connect**, then **send/recv**, then **close**, **shutdown**, or both
- Server calls **listen**, then, in a loop
  - **accept**, then **send/recv**, then **close**, **shutdown**, or both
- First, the socket is created by calling **socket**

# Sockaddrs

- An Address consists of  
(Address Family, IP Number, Port Number)
- Address Family is `AF_INET` for IPv4  
`AF_INET6` for IPv6
- IP number can be given manually, or from domain name  
    `getaddrinfo` converts DNs to IP numbers
- Port number you must know somehow  
    usually 80 or 8080 for web servers

# Using Socket Addresses

- Connect requires an address
- Accept returns the client's address
- For connectionless communications:
  - sendto requires an address
  - recvfrom returns the sender's address