

## Computer Networks – Introduction

### Outline:

- General course information.
- Some basic concepts for computer networks.
- Network programming.

9/3/2014

CSC 257/457 - Fall 2014

1

## General Course Information

- Course Web page
  - <http://www.cs.rochester.edu/~kshen/csc257-fall2014>
- Course email address
  - [cs257@cs.rochester.edu](mailto:cs257@cs.rochester.edu)
- Textbook
  - Kurose and Ross, "Computer Networking: A Top-Down Approach Featuring the Internet", 6<sup>th</sup> edition, 2012 (5<sup>th</sup> edition is probably OK as well).
- Accounts in computer science labs
  - If you don't have one, sign up on a sheet AND see Marty

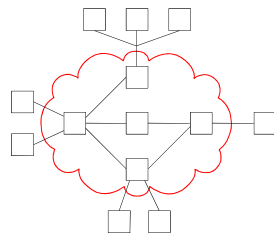
9/3/2014

CSC 257/457 - Fall 2014

2

## Basic Building Blocks for Computer Networks

- Network: two or more hosts connected by links/switches



- Host nodes:
  - PCs, servers, tablets/smart phones, sensors ...
- Links:
  - Twisted pair cable, coaxial cable, phone line, optical fiber, wireless radio channels ...

9/3/2014

CSC 257/457 - Fall 2014

3

## Network Traffic

- **Unicast:** to a single destination node
- **Broadcast:** to all nodes on the network
- **Multicast:** to some subset of nodes on the network
- **Anycast:** to any one in a set of nodes

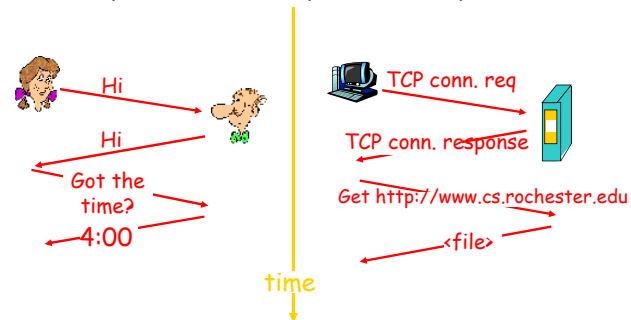
9/3/2014

CSC 257/457 - Fall 2014

4

## Network Protocol: An Example

A human protocol and a computer network protocol:



9/3/2014

CSC 257/457 - Fall 2014

5

## Network Protocol

- A network **protocol** describes how communication entities (sender/receiver) should interact to accomplish a networking task.
  - format, order of messages sent and received between network entities
  - actions taken on message transmission, receipt
- Protocols define communication interface, not implementations on each side
  - allows two sides of a communication task to be independently implemented (Google Chrome browser interacts with Microsoft IIS server) and tested

9/3/2014

CSC 257/457 - Fall 2014

6

## Network Architecture: Layers

- Separation of network functions/features allow independent design and implementation  $\Rightarrow$  thus **distributing the complexity**.
- Layering is a vertical separation of network functions/features.
  - Hiding the complexity**: each layer interface hides complexity in this layer and layers below.
- Disadvantage of layering?
  - lack of full information sharing

Application programs
Host-to-host data transport
Host-to-host connectivity
Communications over link hardware

9/3/2014

CSC 257/457 - Fall 2014

7

## Internet Architecture

Bottom-up:

- physical**: electromagnetic signals "on the wire"
- link**: digital signals (data) transfer on one link (between neighboring network elements)
  - encoding, framing, error correction, access control for shared links
- network**: host-to-host multi-hop connectivity
  - routing, addressing
- transport**: host-to-host data transport
  - reliable data transport, congestion control, flow control
- application**: anything you want to do on computer networks

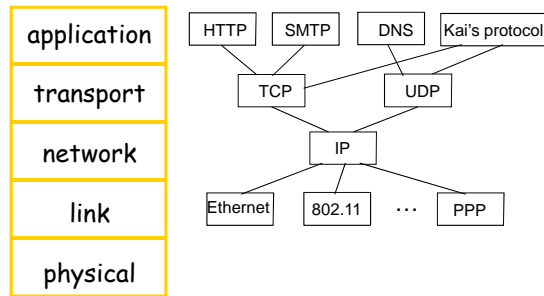
application
transport
network
link
physical

9/3/2014

CSC 257/457 - Fall 2014

8

## Protocols for the Internet Architecture



- Hourglass design.
- Understand layers, protocols, protocol implementations.

9/3/2014

CSC 257/457 - Fall 2014

9

## History/Dominance of the Internet

- Internet (TCP/IP) evolved from a U.S. Defense Department-supported project called ARPANET.
- Internet/ARPANET wasn't the first/only electronic network:
  - Telephone network, X.25
- The Internet becomes dominant today because:
  - Simple network core (packet switching) with efficiency and extensibility.
  - A good TCP/IP implementation was bundled with a popular UNIX operating system (Berkeley UNIX).

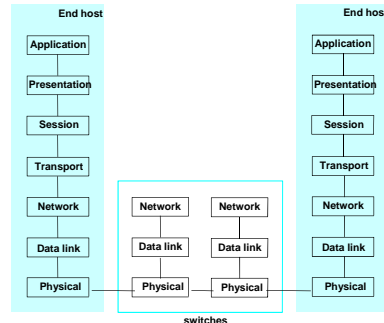
9/3/2014

CSC 257/457 - Fall 2014

10

## OSI Model

- **Session layer:** tie up different transport streams belonging to the same application
- **Presentation layer:** defining uniform format for data representation



9/3/2014

CSC 257/457 - Fall 2014

11

## Course Outline

- Traditional materials about computer networks
  - Data links: Ethernet etc.
  - Network: IP addressing and routing.
  - Transport: TCP/UDP.
  - Application: HTTP, FTP, SMTP, and DNS.
- Advanced topics in computer networks
  - Multimedia networking (quality of service), computer security, wireless (mobile/mesh) networks, overlay networks.
- Case studies of network systems/technologies
  - HTTP load balancing.
  - Content distribution and network caching.
  - Peer-to-peer systems.

9/3/2014

CSC 257/457 - Fall 2014

12

## What we will learn?

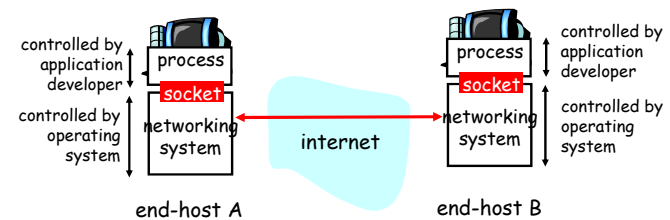
- Practices
  - How do things work in practice?
  - In the case of Internet, the Web, p2p overlay, etc.
- Principles
  - Why things work in ways they do now?
  - Fundamental principles, design rationale (and sometimes a lack of it).

9/3/2014

CSC 257/457 - Fall 2014

13

## Socket Programming



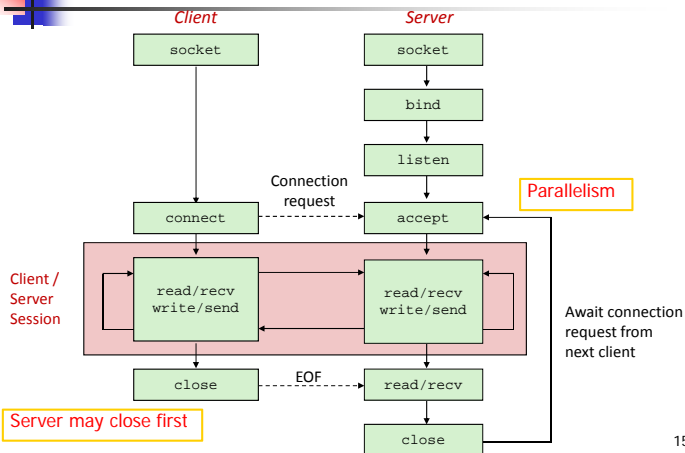
**Socket:** a programming interface on which application processes can communicate with remote application process.

9/3/2014

CSC 257/457 - Fall 2014

14

## Overview of TCP Sockets



15


## Port Numbers

- Multiple sockets might exist in each host.
- A **port number** identifies each such socket in each host.
- Each port number is a 16-bit number, ranging from 0 to 65535.
- Port numbers ranging from 0 to 1023 are restricted for system-level services (you can't use unless you have administrative privilege on the machine).

9/3/2014

CSC 257/457 - Fall 2014


16



## Socket Programming Notes

- `gethostbyname()`
  - Query name server for IP to name translation.


9/3/2014 CSC 257/457 - Fall 2014 17



## Socket Programming Notes

- `connfd = accept(listenfd, ...)`
  - Server waiting for a connection request.
- Notes
  - Returns when the connection with a new client is established and ready for data transfers
  - Data communications are done through the connection socket, not the listening socket
  - One listening socket for the whole server, one connection socket per active connection with a client


9/3/2014 CSC 257/457 - Fall 2014 18



## Socket Programming Notes

- `read/recv, write/send`
  - Data transfer calls.
- Note
  - TCP communications are stream-oriented, with no clear sense of message boundaries
  - when one sends 2000 bytes, the other side may receive twice (with 1460 bytes the first time, and 540 bytes the next time)
- How do you know when you are done with receiving?
  - Your protocol semantics should tell, three typical ways (next slide)

9/3/2014 CSC 257/457 - Fall 2014 19



## Socket Programming Notes

Protocol on the end of message and how to write your receiver code?

- Fixed-size messages defined by protocol
  - `read/recv` for a certain size?
  - Must retry in case you don't get all the first time
  - Likely fine with one try if the size is small
- Fixed-size header which contains a field that indicates the size of body
  - Read the header
  - The rest is similar to above
- Always ends with a special pattern (e.g. an empty line "`\n\n`")
  - Be careful that the ending pattern may straddle over two packets

9/3/2014 CSC 257/457 - Fall 2014 20



## Socket Programming Notes

- Important!
  - Check the return code of every socket call
  - Do error processing, warning messages for all cases
  - See UNIX manual pages for error codes and possible meaning

9/3/2014

CSC 257/457 - Fall 2014

21



## Learn Network Programming

- Learn to use UNIX manual pages
- Find information on the web
- Reference book
  - Network programming in C: Stevens, "Unix Network Programming, Volume 1", 2<sup>nd</sup> edition, 1998.
- Help each other
  - Help with basic questions on network programming, **NOT** writing code for each other's assignment
- C versus Java
- No use of a scripting language until the very last assignment of the class

9/3/2014

CSC 257/457 - Fall 2014

22



## Assignment #0

- Part I: warmup on socket programming, not graded
- Part II: network measurement
  - measure transmission delays for varying TCP traffic
  - report local as well as wide-area Internet measurements
  - report stable results

9/3/2014

CSC 257/457 - Fall 2014

23



## Disclaimer

- Parts of the lecture slides contain original work of James Kurose, Larry Peterson, and Keith Ross. The slides are intended for the sole purpose of instruction of computer networks at the University of Rochester. All copyrighted materials belong to their original owner(s).

9/3/2014

CSC 257/457 - Fall 2014

24