Computer Networks – Introduction

Outline:

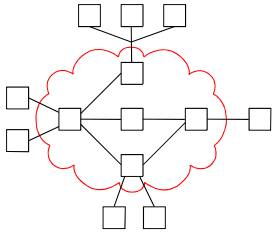
- General course information.
- Some basic concepts for computer networks.
- Network programming.

General Course Information

- Course Web page
  - http://www.cs.rochester.edu/~kshen/csc257-fall2009
- Course email address
  - cs257@cs.rochester.edu
- Textbooks
  - Kurose and Ross, "Computer Networking: A Top-Down Approach Featuring the Internet", 4th edition, 2007; 5th edition, 2009.
  - Peterson and Davie, "Computer Networks: A Systems Approach", 4th edition, 2007.
- Accounts in computer science labs

Basic Building Blocks for Computer Networks

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



- Host nodes:
  - PCs, servers, hand-held devices, sensors …
- Links:
  - Twisted pair, coaxial cable, phone line, optical fiber, wireless radio channels …

Addressing and Routing

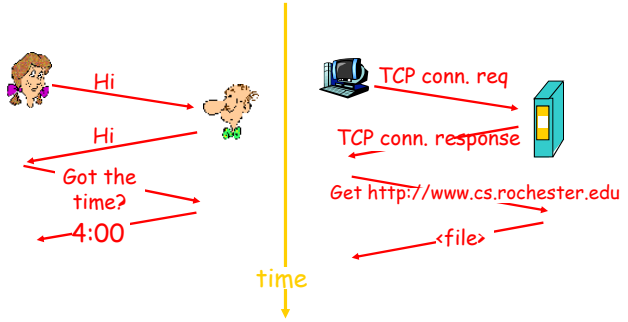
- Address: node identifier in a network.
  - often a unique byte string.
- Routing: finding a route (path) to the destination node based on its address.
- Types of traffic/addressing
  - unicast: to a single destination node;
  - broadcast: to all nodes on the network;
  - multicast: to some subset of nodes on the network;
  - anycast: to any one in a set of nodes.

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 node-local implementations
  - allow two sides of a communication task to be independently implemented.

An Example

A human protocol and a computer network protocol:



Network Layers

- Separation of network functions/features allow independent design and implementation ⇒ 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

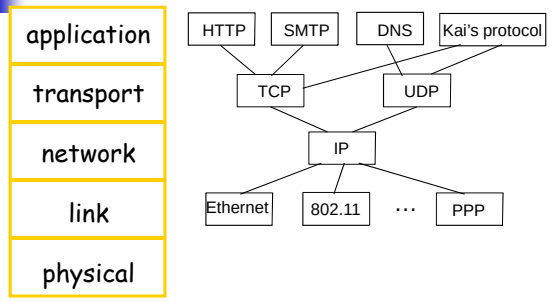
Internet Architecture

Bottom-up:

- physical: electromagnetic signals "on the wire"
- link: data transfer between neighboring network elements
  - encoding, framing, error correction, access control for shared links
- network: host-to-host 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

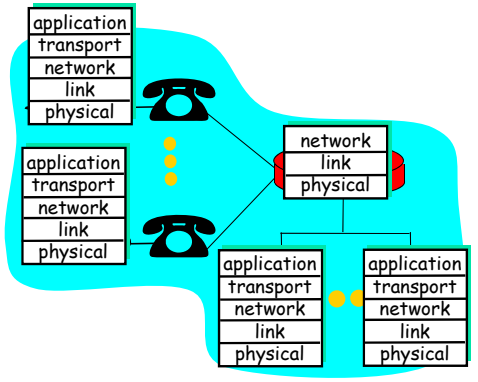
application
transport
network
link
physical

Protocols for the Internet Architecture



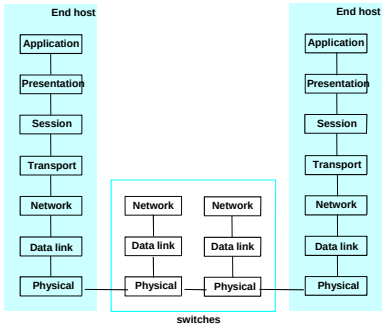
- Hourglass design.
- Understand layers, protocols, protocol implementations.

Internet Layering on Hosts and Switches



OSI Model

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



History/Dominance of the Internet

- Internet (TCP/IP) evolved from a Defense Department-supported project called ARPARNET.
- The Internet becomes dominant today because:
  - Simple core network layer (packet switching) with extensibility.
  - A good TCP/IP implementation was bundled with a popular UNIX operating system (Berkeley UNIX).

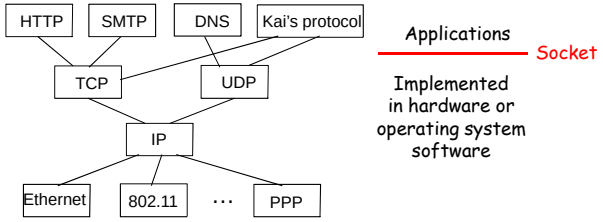
Course Outline

- 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.

What we will learn?

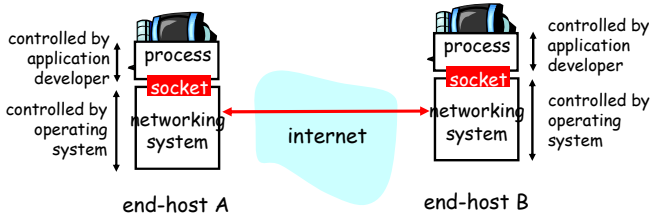
- Practices
  - How do things work in practice?
  - In the case of Internet, the Web, p2p overlay, etc.
- Principles
  - Why things work in ways they do?
  - Fundamental principles, design rationale (and sometimes a lack of it).

Socket Programming



Socket: the interface between application processes and underlying networking systems

Socket Programming (cont.)



Socket: an OS interface (a "door") into which application processes can communicate with remote application process.

TCP/UDP Sockets

Connection-oriented byte streams (TCP):

- Client/server need to establish connection before communications;
- Communications flow in streams of ordered bytes.

Connectionless datagram (UDP):

- No connection is needed before communications;
- Clear boundaries between groups of bytes (messages), but no ordering between messages.

Port Numbers

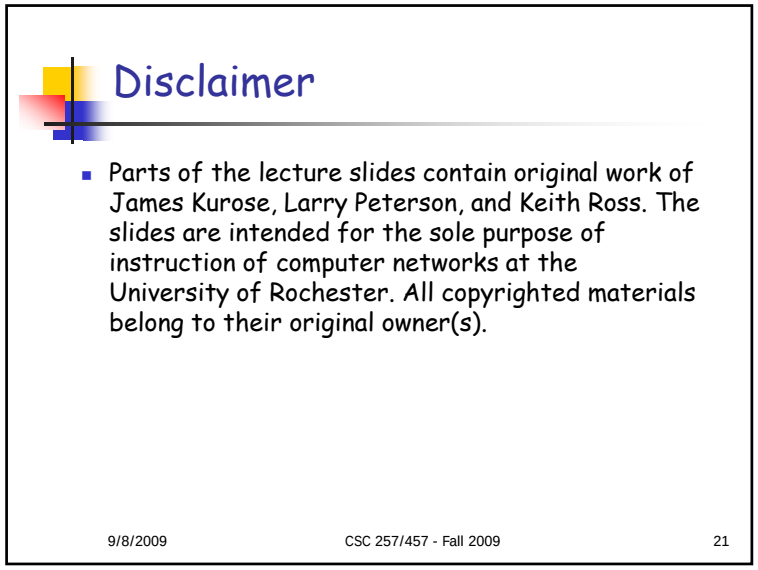
- Multiple sockets might exist in each host.
- A port number identifies each such socket in each host.
- Each port number is a 16-bit number, ranging from 0 to 65535.
- Port numbers ranging from 0 to 1023 are restricted for system-level services.

Learn Network Programming

- Reference books
  - Network programming in C: Stevens, "Unix Network Programming, Volume I", 2nd edition, 1998.
- Learn to use UNIX manual pages
- Find information on the web
- C versus Java

Assignment #0

- Part I: little warmup on socket programming, will not be graded
- Part II: network measurement
  - measuring transmission delays for varying TCP/UDP traffic
  - use high resolution timer
  - report stable results


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).
9/8/2009
CSC 257/457 - Fall 2009
21