WEEK 2 LECTURES CS372

CS 372 Lecture #5

* Overview of Networking:
  + more about access networks
  + physical media
  + Note: Many of the lecture slides are based on presentations that accompany
  + Computer Networking: A Top Down Approach, 6th edition, by Jim Kurose & Keith Ross, Addison-Wesley, 2013.
* Access networks: Home
  + Typical Home Network Components
    - cable/dsl modem
    - router/firewall/NAT
    - Ethernet
    - wireless access point
    - cabled/wireless devices
* Access networks: Local Area Networks
  + Components
    - Insititutional Link to ISP
    - Institutional router
    - Institutional Mail, Web Servers
    - Switches
  + Typically used in companies, universities etc
  + Example: Ethernet
    - local servers, public servers
    - shared or dedicated link connects end systems to edge router
    - 10 Mbs, 100 Mbps, 1 Gbps, 10 Gbps
  + Much more later about LANs
* Access networks: Wireless
  + Shared wireless access network connects end system to router via base station (access point)
  + wireless LANs:
    - within building (100 ft)
    - 802.11b/g (WiFi): 11 or 54
    - Mbps transmission rate
  + wide-area wireless access
    - provided by phone company
    - (cellular) 10 – 30 km
    - 1 to 10 Mbps transmission rate
    - 3G, 4G, LTE
  + Discussion question: What to Internet are 3G, 4G, and LTE?
* Physical media
  + Physical Media
    - provides the required link between sender & receiver
    - propagates bits between sender/receiver pairs
  + Two types of physical media:
    - guided media: signals propagate in solid media (cables and such)
    - unguided media: signals propagate freely, e.g., wireless radio
* Guided Media: twisted‐pair copper wire
  + two insulated copper wires
    - Category 3: 10Mbps Ethernet
    - Category 5: 100Mbps, 1Gbps Ethernet
    - Category 6: 1Gbps, 10Gbps Ethernet
  + still used for high‐speed LAN; e.g., ADSL
  + rate depends on thickness and distance
  + make up 99% of wired connections
  + may pick up interference (“noise”)
  + Discussion question: Why are the pairs twisted?
* Twisted pair connectors (RJ‐45)
  + connector: same on both ends of the cable; used with hubs, switches
  + crossover: one of each; direct connection of 1 computer with another
    - Pins 1, 2: transmit
    - Pins 3, 6: receive
    - Pins 4, 5, 7, 8: not used for network data
    - NOTE: other standards exist (not compatible, consistency essential)
* Guided Media: coaxial cable
  + two concentric copper conductors
  + baseband:
    - More reliable, can handle greater range of frequency
    - single channel on cable
    - legacy Ethernet
  + broadband:
    - multiple channels on cable
    - hybrid fiber‐coax cable (HFC)
  + Cable TV
  + rate depends on thickness and distance
  + less interference than twisted pair
* Baseband and broadband
  + **Baseband** uses a small part of the wave spectrum and sends only one signal at a time
  + B**roadband** uses a larger part of the wave spectrum and uses frequency division multiplexing to send multiple signals simultaneously
* Frequency division multiplexing (FDM)
  + Achieves multiplexing by using different carrier frequencies
  + Frequencies must be separated to avoid interference
  + Receiver can "tune" to specific frequency and extract modulation for that one channel
  + Useful only in media that can carry multiple signals with different frequencies
    - high‐bandwidth required
* Guided Media: fiber optic cable
  + glass fiber carrying light pulses, each pulse represents one bit
  + high-speed operation:
    - high-speed point-to-point transmission (e.g., 10’s-100’s Gbps) l
  + low error rate:
    - immune to electromagnetic noise and other interference
* Wave‐length division
  + Frequency division multiplexing can be applied to optical transmission
    - known as wave‐length division multiplexing (WDM)
    - When many wavelengths are used, the term becomes Dense Wave Division Multiplexing (DWDM)
  + Informally, color division multiplexing
  + Receiver separates frequencies using a prism
    - red, orange, etc
      * More information at <http://www.thefoa.org/PPT/index.html>
      * Source: http://www.thefoa.org
* Unguided media: wireless radio
  + signal carried in electromagnetic spectrum
    - in the “air”
  + no physical wire
  + effects of **propagation environment:**
    - reflection
    - obstruction by objects
    - interference
* Radio link types
  + terrestrial microwave (directional)
    - up to 45 Mbps channels
  + LAN (e.g., Wifi)
    - 11Mbps, 54 Mbps
  + wide‐area (e.g., cellular)
    - 3G, 4G, LTE: hundreds of Kbps
  + satellite
    - 1 Kbps to 45Mbps channel (or multiple smaller channels)
      * Geostationary (36000 km) minimum 280 ms propagation delay
      * Low‐earth orbit (usually 300 – 600 km)
      * Propagation delays depend on configuration
        + Earth 🡨🡪 Satellite 🡨🡪 Earth
        + Earth 🡨🡪 Satellite 🡨🡪 Satellite 🡨🡪 … 🡨🡪 Satellite 🡨🡪 Earth
* **Summary Lecture #5**
  + **Definitions:**
    - **baseband, broadband**
  + **multiplexing**
  + **Physical media**
    - **guided, unguided**

CS 372 Lecture #6 Overview of Networking

* Introduction to Protocols
  + Networks are complicated!
    - hosts
    - routers
    - links to various media types
    - applications
    - protocols
    - other hardware, software
* Layer of functionality
  + Post Office Example
    - Sender (Source)
      * Write letter
      * put into envelope
      * address envelope
      * Take to PO
      * PO routes to recipient
    - Delivery Routing
    - Recipient (Destination)
      * Envelope arrives at destination PO
      * Destination PO accepts envelope
      * Destination PO delivers to address
      * Remove from envelope
      * Read Letter
  + Layers: Each layer implements a service
    - via its own inter-layer actions
    - relying on services provided by the layer above or below
* 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
      * PPP, Ethernet
  + physical
    - carries actual signals between devices
      * Cable, Wireless
* Internet Stack end-to-end transport
  + Message travels from Source to Switch to Router to Destination
  + Message is **encapsulated** at each destination
  + Source
    - Application
    - Transport
    - Network
    - Link
    - Physical
  + Switch
    - link
    - physical
  + Router
    - Network
    - Link
    - Physical
  + Destination
    - Application
    - Transport
    - Network
    - Link
    - Physical
* Why layering?
  + Dealing with Complex Systems
    - Explicit structure allows identification and relationship of complex system’s pieces
      * layered **reference model** for discussion
    - Modularization eases maintenance and updating of complex system
      * change of implementation of layer’s service is transparent to rest of system
    - Change in “deliver to address” procedure doesn’t affect the rest of the system
* Constraints
  + 1. The software of each layer depends only on the services of the software provided by neighboring layers (**well-defined interfaces**)
  + 2. The software at layer *n* at the destination receives exactly the same protocol message sent by layer *n* at the sender (**consistency**)
    - Info at similar layers must have the same meaning
  + 3. These constraints mean that protocols within a protocol stack can be
    - Tested independently
    - Modified/replaced independently
* 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 the application layer of the internet protocol stack
  + ISO Reference Model
    - application
    - ***presentation***- allows apps to interpret meaning
    - ***session-***handles synch
    - transport
    - network
    - link
    - physical
* ISO Model End-to-End Transport
  + diagram of sending and receiving computer
* **Summary Lecture #6**
  + **Definitions**
    - **protocol**
    - **encapsulation**
  + **Internet protocol stack**
  + **ISO layering model**

CS372 Lecture #7

* Overview of Networking: Intro to Network Security
  + Network Security
    - The field of network security is about:
      * How computer networks can be attacked intentionally
      * How computer networks can be “attacked” unintentionally
      * How we can defend networks against attacks
      * How to design architectures that are immune to attacks
    - The internet was not originally designed with security in mind
      * Original vision: “a group of mutually trusting users attached to a transparent network”
      * Internet Protocol designers playing “catch-up”
      * Security consideration in all layers
  + Attackers can put malware into hosts via the internet
    - Malware can get into a host from:
      * **Virus**: self-replicating infection by receiving/executing object (eg e-mail attachment), usually corrupt files on a host
      * **Worm**: self-replicating infection that executes itself as it travels **around a network**
    - **spyware malware**: can record keystrokes, web sites visited and upload info to a collection site
    - Infected host can be enrolled in **botnet**
      * used for spam, distributed denial of service (DDOS) attacks
  + Attackers can attack server, network infrastructure
    - Denial of Service (DDOS)
      * attackers make resources (server, bandwidth) unavailable to legitimate traffic by overwhelming resource with bogus traffic
        + 1. Select Target
        + 2. Break into hosts around the network (see botnet)
        + 3. Send packets to target form compromised hosts

builds server traffic to an extent so that it cannot respond to legit requests

* + Attackers can sniff packets
    - packet “sniffing”:
      * broadcast media (shared Ethernet, wireless)
      * promiscuous network interface reads/records all packets (including passwords!) passing by
    - Wireshark
      * checks contents of packets we send ourself
  + Attackers can use fake addresses
    - IP spoofing: send packet with false source address
  + SUMMARY LECTURE #7
    - Types of security threats
      * malware, spyware
      * denial of service
      * packet sniffing
      * address spoofing

CS 372 Lecture #8

* The application Layer
  + Network applications
  + Application architectures
  + Process communication
* Application Layer
  + Objectives
    - Understand concepts and implementation aspects of network application protocols
      * Transport-layer service models
      * Client-server paradigm
      * peer-to-peer paradigm
    - Understand protocols, by examining popular application layer protocols
      * HTTP - hypertext
      * FTP – file transfer
      * SMTP/POP3/IMAP - email
      * DNS – domain name service
    - Program network applications
* Example Network Applications
  + email
  + web
  + im
  + remote login
  + p2p file sharing
  + multi-user network games
  + streaming stored video clips
  + VOIP
  + real-time video conferencing
  + grid computing
  + Question: Why are these called “applications”?
    - Answer: they communication the data generated by user programs (messages, queries responses, etc)
* Creating a Network App
  + Write programs that can
    - run on (different) *end systems* (hosts) at the network edge
    - communicate via network
      * e.g., web server software communicates with browser software
  + Not much software is written for devices in network core
    - network core devices do not run user applications
    - most operations implemented in hardware
      * certain chips and such
      * because they need to be fast
* Application Architectures
  + 3 types
    - Client-Server
    - Peer 2 Peer
    - Hybrid of client- server and P2P
* Client-Server Architecture
  + Server
    - always-on
    - fixed/known IP address
    - server “farms” and multi-threading for scaling
      * serves many clients simultaneously
  + Clients
    - communicate with server only
    - intermittently connected
    - may have dynamic IP addresses
    - do not communicate directly with each other
  + Examples
    - get services form Google, Amazon, MySpace, YouTube etc.
* Pure P2P Architecture
  + There is no always-on server
    - arbitrary end systems communicate directly
    - peers and intermittently connected and might change IP addresses
    - example: BitTorrent (file distribution through P2P)
  + Pros
    - Scalable
    - distributive
  + Cons
    - difficult to manage
    - not secure
* Hybrid of client-server and P2P
  + Skype
    - Voice of IP P2P application
    - centralized server: finding address of remote party
    - client-client connection: direct (not through server)
  + Instant Messaging
    - chatting between two users is P2P
    - Centralized service: client presence/location
      * user registers IP address with central server when it comes online
      * user contacts central server to find IP addresses of buddies
* Processes communicating – **Process to Process Communication**
  + Process: a program running within a host
    - processes inside a single host communicate using **inter-process communication** (managed by OS)
    - processes in two different hosts communicate by exchanging **messages** (managed by protocols)
  + Client Process- process that initiates communication
  + Server Process- process that waits to be contacted
  + Discussion Question:
    - Applications with P2P architectures have both client processes and server processes
      * -Why?
* Addressing Processes
  + To receive messages, process must have an identifier
  + Receiving host device has unique IP address
  + Q: is IP address of host (on which process runs) sufficient to identify the process?
  + A: No, many processes can be running on same host
* Addressing Processes
  + Identifier consists of
    - IP Address (host)
    - port number (process)
  + This identifier is called a socket or endpoint
  + Example port numbers
    - HTTP server: 80
    - Mail server: 25
  + To send HTTP message to oregonstate.edu web server:
    - IP address: 128.193.4.112
    - Port Number: 80
    - Try it! (http:128.193.4.112.80)
  + Discussion question: Can a server process at a specific port number serve more than one client concurrently? If so ….how?
* Sockets and Connections
  + Process sends/receives messages to/from its socket
  + A connection is a socket pair (4-tuple consisting of the client IP address, client port number, server IP address, and server port number) that uniquely identifies the two endpoints
* Some “well-known” logical port numbers
  + List
* **Summary Lecture #8**
  + **Application Layer**
    - **Examples of network applications**
    - **Application architectures**
      * **client-server, peer-to-peer p2p, etc.**
    - **How processes communicate**
    - **Definitions**
      * **client, server**
      * **port number**
      * **socket, connection**

CS 372 Lecture #9

* The Application Layer:
  + Application layer protocols
  + Transport Layer services required by application layer protocols
* Application Layer Protocols
  + An application layer protocol defines
    - Types of messages exchanged,
      * e.g., request, response
    - Message syntax:
      * what fields are included in a messages
      * how fields are delineated
    - Message semantics
      * Meaning of information in fields
    - Rules for when and how processes send & respond to messages
* Application Layer Protocols
  + Public-domain protocols:
    - defined in RFCs
    - allow for interoperability
      * e.g., HTTP, SMTP, etc.
  + Proprietary protocols
    - defined by the application
      * e.g., Skype
      * e.g., the app you will create for project #1
* First project is an application layer protocol message
* “Lower” layer services required by *Application Layer* protocols
  + Data Integrity
    - some apps (e.g., audio) can tolerate some loss
    - other apps (e.g., file transfer, telnet) require 100% reliable data transfer
  + Timing
    - some apps (e.g., Internet telephony, interactive games) require low delay to be “effective”
  + Bandwidth
    - some apps (e.g., multimedia) require a certain amount of bandwidth to be “effective”
    - other apps (“elastic apps”) make use of whatever bandwidth they get
  + Security
    - How big a problem is this?
* Lower layer services required by common applications
  + chart with fields:
    - application
    - allowed data loss
    - bandwidth
    - time senstitive
* Recall (Lecture #2) what services do internet transport protocols provide?
  + TCP Service
    - connection oriented: setup required between client and server processes
    - reliable transport: between sending and receiving processes
    - flow control: sender won’t overwhelm receiver
    - congestion control: throttle sender when network overloaded
    - does not provide: timing, minimum bandwidth guarantees
  + UDP service
    - “Unreliable” (best effort) data transfer between sending and receiving process
    - does not provide: connection setup, reliability, flow control, congestion control, timing or bandwidth
  + Discussion question: Why is there UDP?
    - when speed is more important than reliability
* Internet Applications: Application and Transport Protocols
  + Fields:
    - Application
    - Application Layer Protocol
    - Underlying Transport Protocol
* **Summary Lecture #9**
  + **Application Layer Protocols**
    - **Responsibilities**
    - **Examples**
  + **Transport Layer Services required by application layer protocols**
    - **Reliability, timing, security, etc.**
    - **Examples**