Agenda

- **Last time**
  - Properties of Dis sys, why dis sys?, challenges
- **This time**
  - More system models (chpt 1 + 2)
  - Networks (chpt 3)
- **Next time (Tues)**
  - *No class Tuesday Jan 30 (Marty at conference)*
    - *Will be made up Thurs Feb 8 / Fri Feb 9*
- **Real next time (Thurs, Feb 1) – only one class**
  - Interprocess Communication (chpt 4)
  - 1st HW/PA out.

Before we start

- **How do we make-up classes? Thurs night AND/OR Friday afternoon?**
  - Thurs night 5-6:15 AND Fri 3-4:15 ??
- **Office hours:**
  - How about Mon 2-4pm

Chapter 2: Distributed System Models

- **Architectural models**
  - How do components of the system interact?
  - How are these components map onto the underlying network of computers?
- **Fundamental models**
  - Formal description of system properties common in all architectural models
  - Interaction, failure, security models
  - Appear throughout the course, but not discussed in detail

Architectural Models

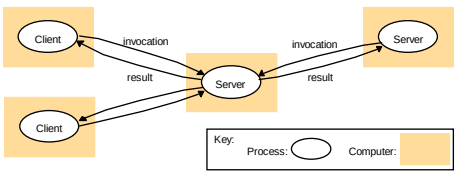
- Architecture: structure of separately specified components
- Overall goal: Structure should meet present and future demands on reliability, manageability, adaptability, and cost-effectiveness
- Functions of individual components not interesting
  - Abstracted away
- **Consider instead:**
  - **Placement (across network)**
  - **Patterns for data/workload distribution**
  - **Interrelationships (Functional roles, communication patterns)**

Architectural Models

- **Client/server vs. peer processes**
- **Architectural models can be used to determine placement**
- **More dynamic systems can be built as variations on the client-server model**
  - Moving code
  - Adding/removing nodes
    - Discovery and advertisement of services

System Architectures: Client-Server Model

- **The most widely used**



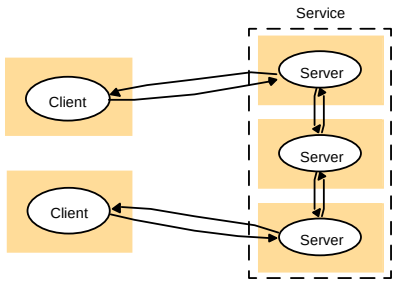
System Architectures: Client-Server Model

- Client: Process wishing to access data, use resources or perform operations on a different computer
- Server: Process managing data and all other shared resources amongst servers and clients, allows clients access to resource and performs computation
- Interaction: invocation / result message pairs
- Example
  - http server: client (browser) requests page, server delivers page

- **Passive (server) vs. active (client)**
- **Stateless vs. stateful**

- **Three-tier architecture: client, app server, and DB server ("business logic")**

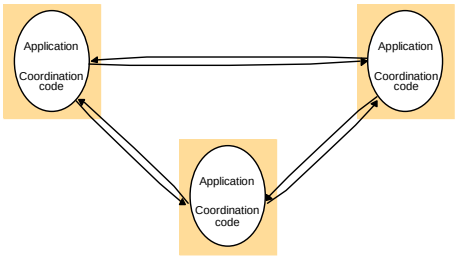
System Architectures: Multiple Servers Model



System Architectures: Multiple Servers Model

- **Services may be provided by multiple servers**
- **Partitioned or replicated service-related objects**
- **Replication provides**
  - Increased performance, availability and fault-tolerance
- **But requires replica coordination / consistency preservation**
- **E.g. high availability web servers (portals, download centers), information services**
- **Servers maintain either replicated or distributed database**

System Architectures: Peer-to-Peer Model



System Architectures: Peer-to-Peer Model

- **Peer processes: processes that play similar roles**
  - No absolute distinction between client/server
  - May still assume client/server roles from time to time
- **Reduces inter-process communication delay for local object access**
- **Increased fault-tolerance and scalability**
- **Coordination difficult**
- **E.g. distributed search, routing, distributed computing, news servers (and of course MP3s)**

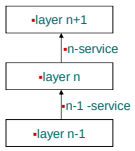
Client/Server vs. P2P

- **Client/Server**
  - More secure

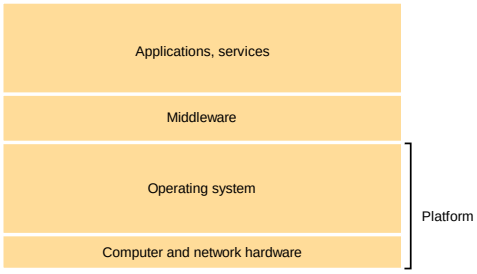
- **P2P**
  - No single point of failure or bottleneck

Architectural Models: Layers

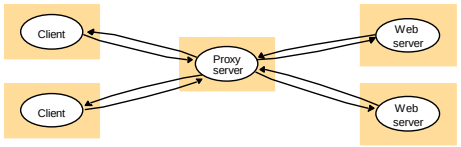
- **Basic idea: break up the complexity of systems by designing them through layers and services**
  - layer: group of closely related and highly coherent functionalities
  - service: functionality provided to a superior layer
- **Examples of layered architectures**
  - operating systems (kernel, other services)
  - computer network protocol architectures
- **Problem with layered approach: not always possible to implement service at layer N using ONLY layer N-1 services**



Layers



System Architectures: Proxy Server Model



- **Cache: a close store of recently used data**
  - Considerably increases performance in many applications
  - But requires cache coherence protocols
- **Proxy server: a shared cache of resources**
  - Renders replication/distribution transparent
  - Most commonly used for web access
  - Frequently used in WWW (e.g., Akamai)

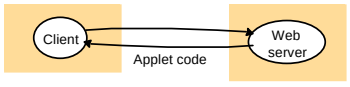
Variations on the Client-Server Model

- **Variations inside the client-server model from following factors:**
  - server initiated communication (e.g., "call-backs")
  - the use of mobile code and mobile agents
  - lightweight clients based on users' need for low-cost computers and easy management

Variations on the Client-Server Model

- **Mobile code: Code that is sent to a client process to carry out a specific task**

a) client request results in the downloading of applet code



b) client interacts with the applet

Variations on the Client-Server Model

- **Mobile agents**
  - Executing program (code + data) that migrates between processes in a network
    - Carries out an autonomous task usually on some other process' behalf
    - Has internal knowledge, beliefs and goals
  - Advantage: local access everywhere
    - Savings in communication costs
  - Potential security threat
    - Limited applicability
  - E.g., information collection from multiple sources, installation of programs, worm programs (e-mail)

Variations on the Client-Server Model

- **Network computer**
  - All files stored remotely
  - Minimum of local software
  - Any local disk used mainly as a cache
- **Thin client**
  - Does not even run its own applications
  - Executes windows-based user interface on a local computer
  - Programs are run by a powerful compute server

Chapter 3: Networks - Basics

- **A network consists of:**
  - Transmission media (wire, cable, …).
  - Hardware devices (routers, switches, …).
  - Software components (protocol stacks, drivers, …).
- **Terminology:**
  - Host: Computers and other devices that use a network.
  - Node: Any computer or switching device attached to a network.
  - Subnet: Set of interconnected nodes.
- **Design issues: Performance, scalability, reliability, security, mobility, quality of service.**

Networks - types

	Range	*Bandwidth (Mbps)*	*Latency (ms)*
LAN	1-2 kms	10-1000	1-10
WAN	worldwide	0.010-600	100-500
MAN	2-50 kms	1-150	10
Wireless LAN	0.15-1.5 km	2-11	5-20
Wireless WAN	worldwide	0.010-2	100-500
Internet	worldwide	0.010-2	100-500

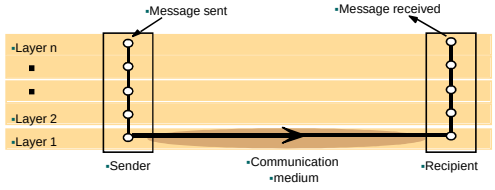
Networks - packet transmission

- **Message: Sequence of data items of arbitrary length.**
- **Messages subdivided into packets.**
- **Switching schemes:**
  - Broadcast.
  - Circuit switching.
  - Packet switching.
  - Frame relay.

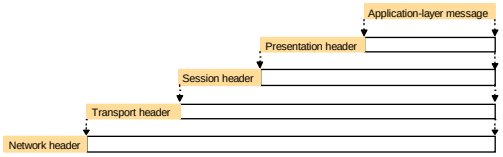
Protocols - basics

- **Protocol: Set of rules and formats to be used for communication between processes in order to perform a given task.**
- **Should include specification of:**
  - Sequence of messages that must be exchanged.
  - Format of the data in the messages.
- **Implemented by a pair of software modules in the sending and receiving computers.**

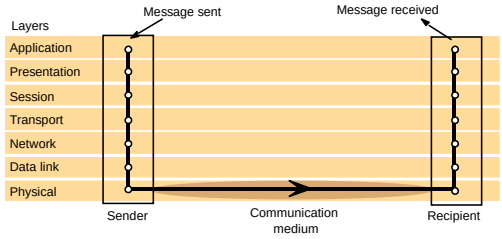
Protocol layers - conceptual layering of protocol software



Protocol layers - encapsulation and headers



Protocol layers - the ISO Open Systems Interconnection (OSI) model



Protocol layers - OSI protocol summary

Layer	*Description*	*Examples*
Application	Protocols that are designed to meet the communication requirements of specific applications, often defining the interface to a service.	HTTP, FTP, SMTP, CORBA IIOP
Presentation	Protocols at this level transmit data in a network representation that is independent of the representations used in individual computers, which may differ. Encryption is also performed in this layer, if required.	Secure Sockets (SSL), CORBA Data Rep.
Session	At this level reliability and adaptation are performed, such as detection of failures and automatic recovery.	
Transport	This is the lowest level at which messages (rather than packets) are handled. Messages are addressed to communication ports attached to processes, Protocols in this layer may be connection-oriented or connectionless.	TCP, UDP
Network	Transfers data packets between computers in a specific network. In a WAN or an internetwork this involves the generation of a route passing through routers. In a single LAN no routing is required.	IP, ATM virtual circuits
Data link	Responsible for transmission of packets between nodes that are directly connected by a physical link. In a WAN transmission is between pairs of routers or between routers and hosts. In a LAN it is between any pair of hosts.	Ethernet MAC, ATM cell transfer, PPP
Physical	The circuits and hardware that drive the network. It transmits sequences of binary data by analogue signalling, using amplitude or frequency modulation of electrical signals (on cable circuits), light signals (on fibre optic circuits) or other electromagnetic signals (on radio and microwave circuits).	Ethernet base-band signalling, ISDN

Protocol layers - internetwork layers