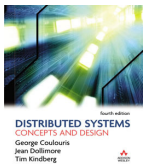
Characterization of Distributed Systems



From **Coulouris, Dollimore and Kindberg**
Distributed Systems:
Concepts and Design
Edition 4, © Pearson Education 2005

Networking and Parallel Computing

- Computer networking
  - Hardware that connects computers
  - Software that sends/receives messages from one computer to another, which might be on different networks (end to end delivery)
  - Goal is to transmit messages reliably and efficiently
- Parallel Computing
  - Multiple homogeneous processors in "one" computer
  - Shared or distributed memory
  - Goal is to execute a program faster by division of labor

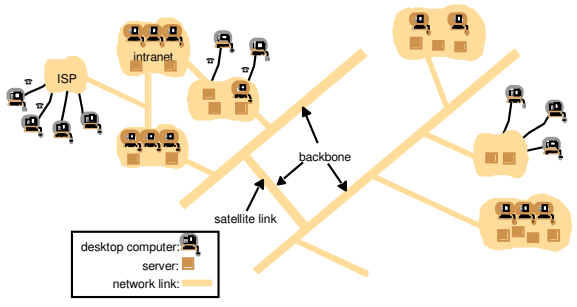
Distributed Computing

- Networked computers that could be far apart
  - rely on computer networking
- Communicate and coordinate by sending messages
- Goal is to share (access/provide) *distributed* resources
- Issues:
  - Concurrent execution of processes
  - No global clock for coordination
  - More components, more independent failures

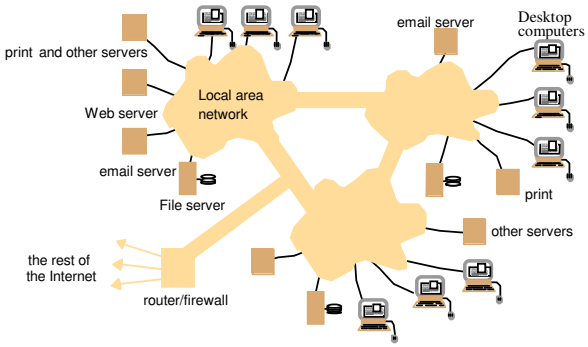
Examples of Distributed Systems

- Global Internet
- Organizational Intranets--behind router/firewall
- Mobile Computing -- computers move
- Ubiquitous Computing -- computers embedded everywhere
- Issues:
  - discovery of resources in different host environments
  - dynamic reconfiguration
  - limited connectivity
  - privacy and security guarantees to the user and the host environment

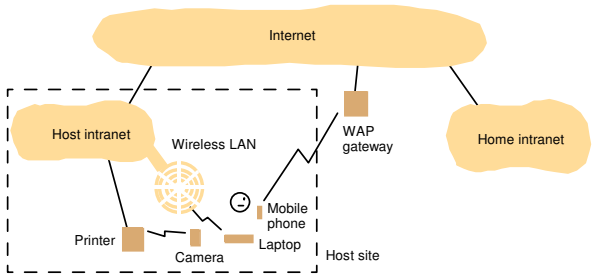
A Typical Portion of the Internet



A Typical Intranet



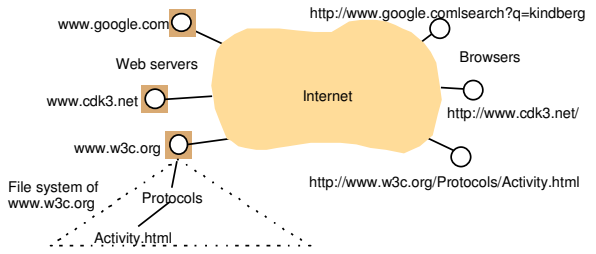
Portable and handheld devices



Resources Sharing and the Web

- HTML, Hyper Text Markup Language
- URL, Uniform Resource Locator
  - http://servername[:port] [/pathname] [?arguments]
- HTTP, HyperText Transfer Protocol
  - request-reply protocol (client-server)
  - content types--MIME types, multipurpose internet mail extensions
  - one resource per request
  - simple access control (mostly public)

Web Servers and Web Browsers



Other Web Technologies

- web forms
- CGI programs, common gateway interface, run on the server
- applets, run on the client
- RDF, resource description framework, vocabulary for meta-data
- XML, extensible markup language, allow meta-data information to be included

Computers in the Internet

Date	*Computers*	*Web servers*
1979, Dec.	188	0
1989, July	130,000	0
1999, July	56,218,000	5,560,866
2003, Jan.	171,638,297	35,424,956

Computers vs. Web servers in the Internet

Date	*Computers*	*Web servers*	*Percentage*
1993, July	1,776,000	130	0.008
1995, July	6,642,000	23,500	0.4
1997, July	19,540,000	1,203,096	6
1999, July	56,218,000	6,598,697	12
2001, July	125,888,197	31,299,592	25
		42,298,371	

Challenges and Issues (1)

- Heterogeneity
  - networks, hardware, os, languages...
  - middleware—corba
  - mobile code, virtual machines
- Openness
  - extended and re-implemented in various ways
  - standard published interfaces
  - RFC, request for comments
- Security
  - confidentiality
  - integrity
  - availability

Challenges and Issues (2)

- Scalability
  - effective with significant increase in resources
  - cost
  - performance
- Failure handling
  - detecting
  - masking—hide, less severe (retransmit)
  - tolerating--ignore, timeout
  - recovery--logs, rollback
  - Redundancy
- Concurrency

Challenges and Issues (3)

- Transparency
  - *Access transparency*: enables local and remote resources to be accessed using identical operations.
  - *Location transparency*: enables resources to be accessed without knowledge of their physical or network location (for example, which building or IP address).
  - *Concurrency transparency*: enables several processes to operate concurrently using shared resources without interference between them.
  - *Replication transparency*: enables multiple instances of resources to be used to increase reliability and performance without knowledge of the replicas by users or application programmers.

Challenges and Issues (4)

- Transparency
  - *Failure transparency*: enables the concealment of faults, allowing users and application programs to complete their tasks despite the failure of hardware or software components.
  - *Mobility transparency*: allows the movement of resources and clients within a system without affecting the operation of users or programs.
  - *Performance transparency*: allows the system to be reconfigured to improve performance as loads vary.
  - *Scaling transparency*: allows the system and applications to expand in scale without change to the system structure or the application algorithms.