

## CS 5523 Lecture 1: Overview of Distributed Systems

- Course administration and web page
- Quick review of operating systems (uniprocessor)
- Definition of distributed systems
- Implications
- Examples
- Challenges

## Course administration and web page

- Syllabus
- Course web page: <http://vip.cs.utsa.edu/classes/cs5523s2003>
- Prerequisites
- Textbooks (CDK, Stevens I and Stevens II, HC I and HC II)
- Programming requirements
- Topics and schedule
- Access list forms

## Quick review of operating systems

- Process control
- Memory management
- Device management

## What is a distributed system?

CDK's Definition:

*A system in which hardware and software components located on networked computers communicate and coordinate their actions only by passing messages.*

- Why have distributed systems?
- How does this compare with a traditional operating system such as Unix?
- What implications does this definition have on implementation?

Figure 1.1 (CDK)  
A typical portion of the Internet

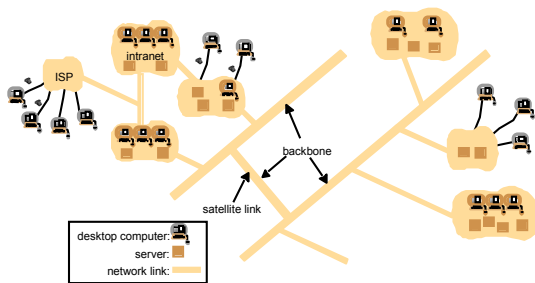


Figure 1.2 (CDK)  
A typical intranet

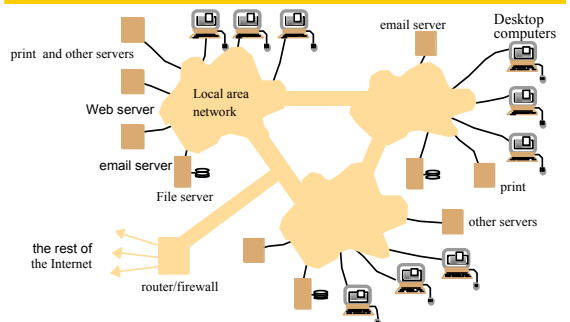
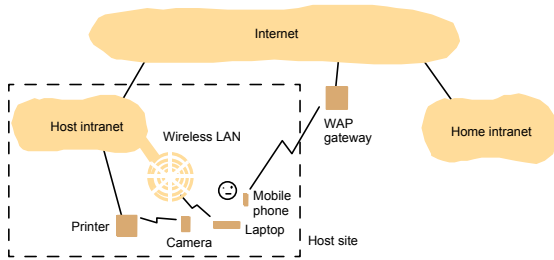


Figure 1.3 (CDK)  
Portable and handheld devices in a distributed system



Instructor's Guide for Cookson, Dullman and Kniberg. Distributed Systems: Concepts and Design. Eds. 3  
© Addison-Wesley Publishers 2000

## World Wide Web

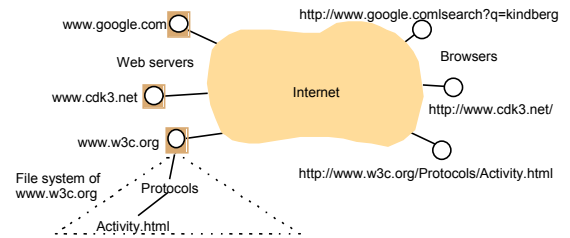
- Ultimate distributed system.
- Developed in 1989 in CERN Switzerland for document exchange among physicists.
- Documents have a hypertext structure.
- New resources are located by following these hypertext links.
- Uses a client-server model.
- A browser (client) on a user machine makes requests and handles the display. New document formats can be handled by plug-ins or helpers without changing the browser software.
- The web server only delivers documents and is not concerned with user interfaces or document format.

Figures 1.5 and 1.6 (CDK)  
Computers vs. Web servers in the Internet

Date	Computers	Web servers	Percentage
1979, Dec.	188	0	0
1989, July	130,000	0	0
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

Instructor's Guide for Cookson, Dullman and Kniberg. Distributed Systems: Concepts and Design. Eds. 3  
© Addison-Wesley Publishers 2000

Figure 1.4 (CDK)  
Web servers and web browsers



Instructor's Guide for Cookson, Dullman and Kniberg. Distributed Systems: Concepts and Design. Eds. 3  
© Addison-Wesley Publishers 2000

## World Wide Web Components

- **HTML (HyperText Markup Language)** - specifies the format for the documents delivered by the server
- **URL (Uniform Resource Locator)** - specifies a resource that is accessible via the web.
- **HTTP (HyperText Transfer Protocol)** - specifies the interaction between browsers and web servers

## HTML format for web documents:

- Text format so can be created using Text editor or HTML tool.
- Uses tags to specify content:
 

```
<P> A paragraph </P>
<IMG SRC="Book.gif">
<A HREF="http://www.utsa.edu">UTSA</A>
```
- The `<A HREF...> X</A>` specifies a link identified by X in the text.

## URLS:

---

### Format

*scheme: scheme-specific-location*

### Format of scheme-specific-location when scheme is HTTP:

*http://servername[:port]/[pathOnServer]/[arguments]*

### Examples

*http://www.utsa.edu*

*http://vip.cs.utsa.edu/classes/cs5523s2001*

*http://www.google.com/search?q=browsers*

## HTTP:

---

Assumes reliable delivery (in practice TCP sockets).

Request-reply protocol:

*Client initiates with a request (GET)*

*Server responds with requested document or an error*

Content types identify document types for browser.

HTTP 1.0 - need a separate request for each resource

HTTP 1.1 - requests pipelined and served by a single connection.

Executables:

*CGI (Common Gateway Interface) executes on server*

*Java Applets execute on browser*

## Challenges for distributed systems

---

- Heterogeneity
- Openness
- Security
- Scalability
- Failure handling
- Concurrency
- Transparency

### Figure 1.7 (CDK) Transparencies

---

*Access transparency:* enables local and remote resources to be accessed using identical operations.

*Location transparency:* enables resources to be accessed without knowledge of their location.

*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.

*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.

Instructor's Guide for Cookson, Dillmann and Knudsen: Distributed Systems: Concepts and Design, Edn. 3  
© Addison-Wesley Publishers 2000

## For next time:

---

- Read CDK Chapter 1
- Look at Exercises 1.6, 1.7, 1.9, 1.12
- BE prepared to discuss assigned questions from Chapter 1
- Start reading CDK Chapter 2
- Look over Laboratory 1