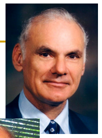
University of Illinois at Urbana-Champaign
Dept. of Electrical and Computer Engineering

ECE 101: Computing Technologies and the Internet of Things

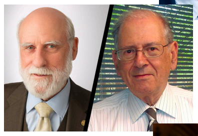
Client-Server: Providing Services on the Internet

Part 1

What Good is the Internet?



1969 - Larry Roberts created ARPANET (first packet network, the the precursor to the Internet).



Mid 1970s - Vint Cerf and Bob Kahn built TCP/IP

1989 - **Tim Berners-Lee** invents World Wide Web

1993 - NCSA releases Mosaic (first widely available web browser)

Researchers used the Internet
- for more than a decade
- before UIUC made it important
- to the other 99.9% of the world

A Server Provides Some Sort of Service

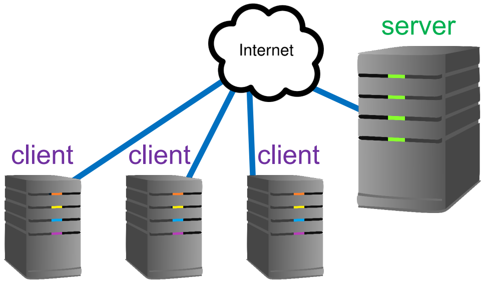
Some computer may **provide a certain service**, such as
- providing copies of published IRS tax documents,
- accepting paper submissions to a research conference, or
- computing turbulence in fluid flow around a structure,

We call that computer a **server**.

A Server's Clients Make Use of that Service

Other computers **contact that computer, and use the service**.

These computers are the **clients** for that server.

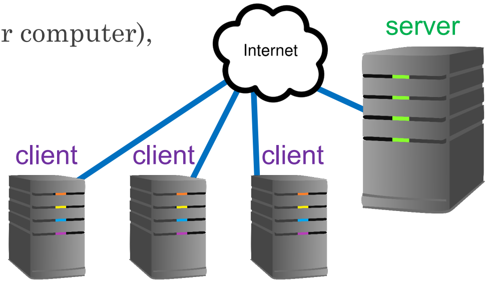


Any Computer Can Provide or Use a Service

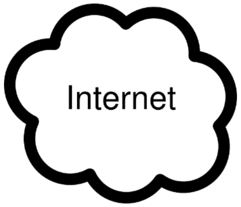
Note that a **server**

- **may also be a client**
- to another server (another computer),
- and vice-versa.

Client and server are just roles for a given service.



Internet Itself can be Considered a Service



Is the Internet itself a service?

One can certainly view it that way.

What is the service?

Try to **deliver a packet** of bits (information) to a specified address (IP address).

What are the clients?

Any computer that wants to communicate.

What are the server(s)?

Routers (and Internet Service Providers, and …)

Internet Service Analogous to Postal Delivery

In that sense,
the **Internet** is **like the post office**.

To send bits,

- put your bits into envelopes (packets),
- write the (IP) address on the front, and
- drop them into the nearest router.

Delivering a Letter Requires Many Forwarding Steps

What happens when you send a physical letter?

You drop the letter into a mailbox.

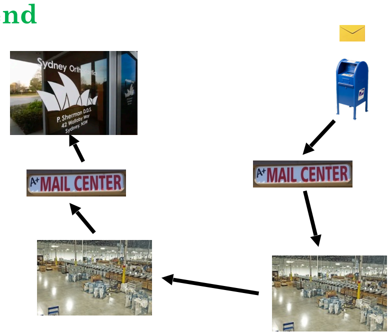
Your letter is taken to a local center.

And forwarded to a regional center.

And to another regional center.

And to a local center near recipient.

And finally to the recipient!

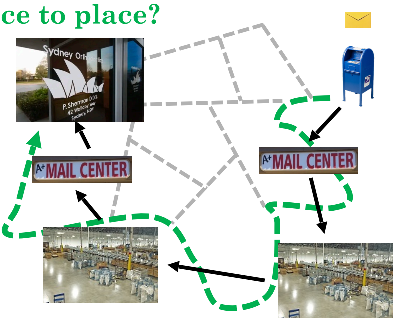


Mail Travels over Some of the Road Network

How do the servers (the employees of the post office) move your letter from place to place?

Using the roads (and sometimes the airways)!

But there are **a lot more roads** that are **not used** by the post office!

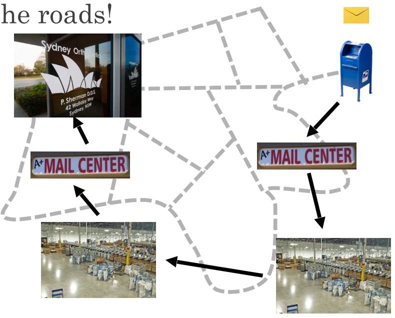


Services Also Define Virtual Networks Over the Internet

The **post office uses** its own **virtual road network** (**black arrows**) on top of the roads!

Can a service define a virtual network over the Internet?

Absolutely! E-mail, for example, **does** just that. Let's see how.

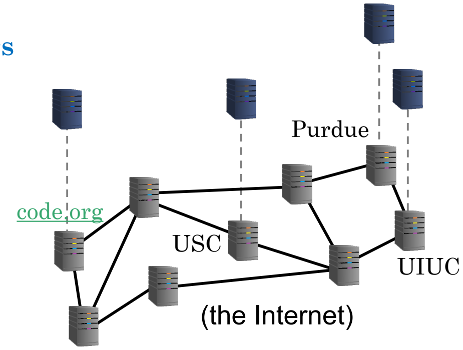


E-Mail Services Provided by Some Computers

With e-mail, some computers provided mailbox services:

- servers at UIUC and other major universities, and
- servers at some companies.

But **not every computer** was an email server!



E-Mail Servers Connected through the Internet

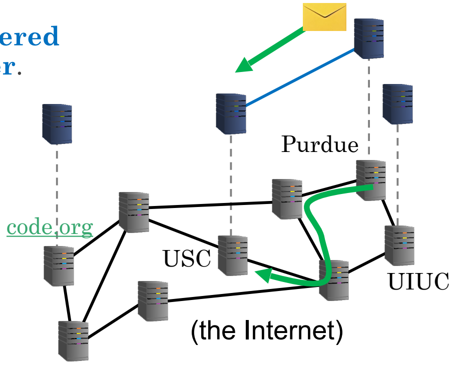
To send an e-mail, a **user delivered** a **message to any** e-mail **server**.

For example:

- from jan@purdue.edu
- to pat@code.org

The e-mail server **forwarded** the message to another e-mail server …

… **through the Internet**.

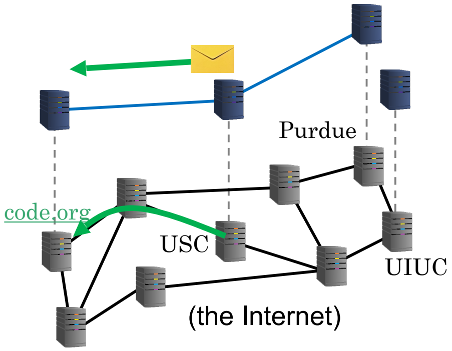


Messages Move from E-Mail Server to E-Mail Server

That server might forward the message to another server…

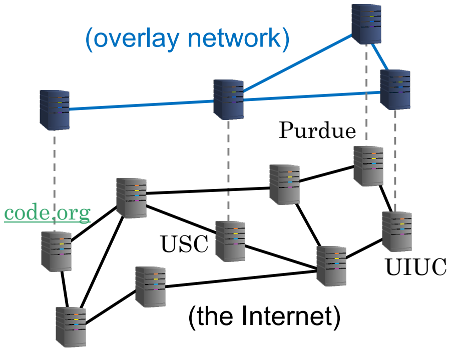
…again, through the Internet…

…until the message reached the destination mailbox.



Together, E-Mail Servers Form an Overlay Network

The “connections” between e-mail servers
- form an **overlay network**:
- **a subset of nodes**
- **connected by virtual links**
- (also called a virtual network).



Next: the World Wide Web Service

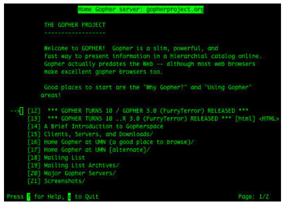
Let’s look at more examples of services.

How about the **World Wide Web**…

(which some people may call “the Internet”)?

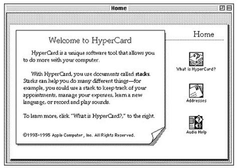
Gopher and HyperCard: Examples of Early Tech

Early Internet services such as **Gopher**
- allowed clients to **explore text documents**
- spread **across multiple servers**
- such as guidance on the requirements for undergraduate curriculum at UIUC.



```
                 Home Gopher server: gopherproject.org

           THE GOPHER PROJECT
           ------------------

           Welcome to GOPHER!  Gopher is a slim, powerful, and
           fast way to present information in a hierarchial catalog online.
           Gopher actually predates the Web -- although most web browsers
           make excellent gopher browsers too.

           Good places to start are the "Why Gopher?" and "Using Gopher"
          areas!

 -->[] [12]  *** GOPHER TURNS 10 / GOPHER 3.0 (FurryTerror) RELEASED ***
       [13]  *** GOPHER TURNS 10 ..R 3.0 (FurryTerror) RELEASED *** [html] <HTML>
       [14] A Brief Introduction to Gopherspace
       [15] Clients, Servers, and Downloads/
       [16] Home Gopher at UMN (a good place to browse)/
       [17] Home Gopher at UMN [alternate]/
       [18] Mailing List
       [19] Mailing List Archives/
       [20] Major Gopher Servers/
       [21] Screenshots/

Press ? for Help, q to Quit                                      Page: 1/2
```

Apple’s **HyperCard**
- enabled users to **move from page to page**
- **by clicking on** a **keyword** or an icon.
- **Sound familiar?**



Welcome to HyperCard

HyperCard is a unique software tool that allows you to do more with your computer.

With HyperCard, you use documents called **stacks**. Stacks can help you do many different things—for example, you could use a stack to keep track of your appointments, manage your expenses, learn a new language, or record and play sounds.

To learn more, click “What is HyperCard?,” to the right.

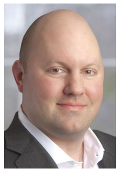
©1993-1995 Apple Computer, Inc. All Rights Reserved.

What Most People View as the Internet Arrived in 1993

These ideas were **combined into a single protocol** (HyperText Transfer Protocol, **HTTP**) in 1989 by Tim Berners-Lee (at CERN).

The **first web browser** (integrating images with text), **Mosaic**, was **developed by Marc Andreessen** and others (at UIUC) in 1992, and made public in 1993.

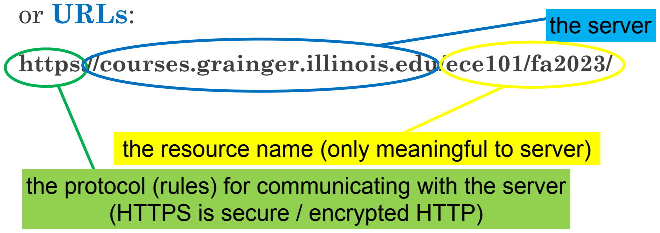
The browser made the Internet interesting to the rest of humanity.



HTTP Protocol Perhaps Familiar to You?

A **web server** is an example of an **Internet service**.

Online resources are named using “Universal Resource Locators”, or **URLs**:



the server

https://courses.grainger.illinois.edu/ece101/fa2023/

the resource name (only meaningful to server)

the protocol (rules) for communicating with the server
(HTTPS is secure / encrypted HTTP)

Web Browser is a Client to a Web Server

A **web browser**
- is **client software**
- that enables a human
- **to** make use of **web servers**.

Last week, we talked about how a web browser communicates with a server.

The URL is what a human (or another web page) provides to identify which server to contact.

In a couple of weeks,
- we’ll look at web search,
- another Internet service
- that allows one to find interesting URLs.

Clients and Servers Must Interact Correctly

Each **Internet service** is unique:
- **defines what it provides,**
- defines the **rules for clients** to make requests for services, **and**
- defines the **form of answers** and how they are returned to clients.

Clients must know these things—generally, **every service has distinct client software**!

Another Simple Service: Use That Computer?

**What if I just want to
let someone else use my computer
(over the Internet)?**

Before we can answer that question, we need to understand the role of the operating system —we'll call it the OS.

Most Modern Operating Systems Based on Unix

**Can you name some
operating systems (OSs)?**

Most operating systems today are
- **based on** Bell Labs' **Unix** (1970s)
- (Windows, MacOS, Linux, Android, IOS, and so forth).

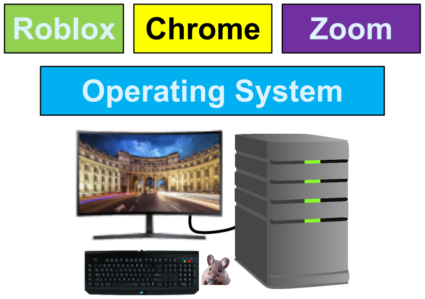
Operating System Interfaces Applications to Hardware

What exactly is an OS?

Software that
- sits **between applications**
- **and the hardware** (computer, monitor, keyboard, mouse)
- and **provides** …

…**services**!

(We like that word a lot.)



Roblox

Chrome

Zoom

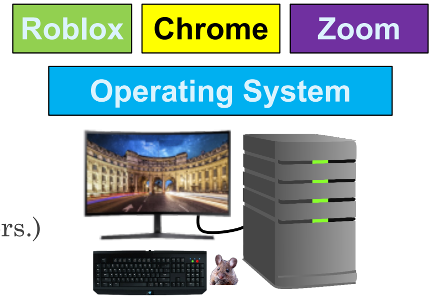
Operating System

Operating System Simplifies Use of Devices

What does the OS do?

Abstracts – **simplifies** the **use of input and output devices**: keyboard, mouse, display, network, printer

(Hardware devices often define vendor-specific, complicated protocols and do not tolerate errors.)



Roblox

Chrome

Zoom

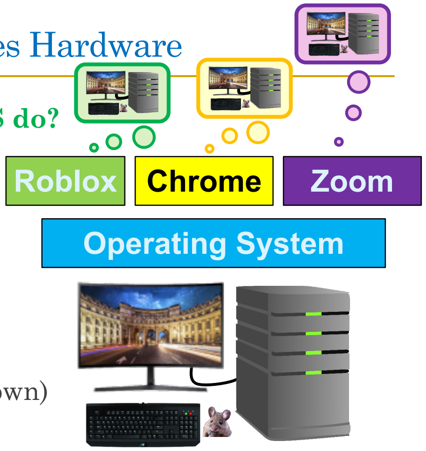
Operating System

Operating System Virtualizes Hardware

What does the OS do?

Virtualizes – **makes** small number of physical **resources available** to many programs **at the same time**, as if each program had private hardware

(includes memory, which isn't shown)



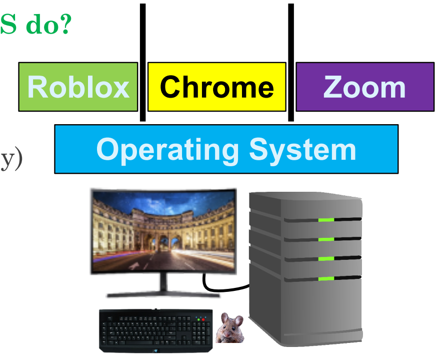
Roblox Chrome Zoom

Operating System

Operating System Protects Programs (and Hardware)

What does the OS do?

Protects—stops programs from interfering with each other (deliberately or accidentally) or with the hardware



Roblox Chrome Zoom

Operating System

Virtual Machine: Software that Mimics a Computer

- In the 1960s IBM created a "**virtual machine**", a program that **simulates a real computer**
  - A computer is hardware.
  - Write software to do the same thing as hardware - so it can pretend to be the hardware.

Cheap Computers Eliminated Need for Virtual Machines

Then the PC (personal computer) appeared, and
- computers became cheap, and
- the idea went away.

Mostly.

Until the 1990s.

(This kind of coming and going happens a lot in technology!)

Time for a Poll

How many of you have a calculator?

A computer?

A top-of-the-line graphics card (graphics processing unit, or GPU)?

A supercomputer?

What if you wanted to use a supercomputer for a few minutes?

Can Someone Else Use that Computer Now?

In the 80s and 90s, computers were still somewhat of a luxury item—most families did not own one.

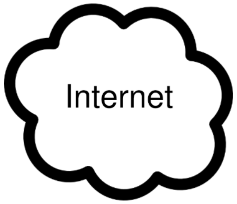
Universities, on the other hand, had many (tens or even hundreds!).

Why not create a model in which someone could use a computer for a little while, whenever they needed one?

Computing as a Service … in the "Cloud" !

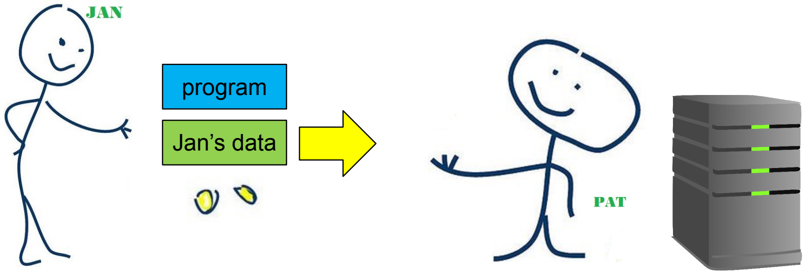
Eventually, this idea became **cloud computing**, the idea of **using someone else's computer as a service**.

Not just individuals:
- company A can use company B's computers
- to provide a public Internet service.
- Customers (or ads!) pay A, and A pays B.



Cloud Computing Model with Jan and Pat

There are some challenges…
Imagine that Jan wants to use Pat's computer.



Trust is a Big Issue for Cloud Computing

Does Jan trust Pat with the program?
Pat could use the program for Pat's own purposes without paying!

Does Jan trust Pat with Jan's data?
Pat could do many things with Jan's data!

Does Pat trust Jan's programs not to hurt Pat's computer or something else, such as by attacking another computer?
Pat could lose the computer or get in trouble!

Technologies Can Help with the Trust Issues

Some technologies were developed to help…

Sandboxing

- program executes in a "**sandbox**,"
- which **keeps** the **program from hurting anything** outside the sandbox
- Today, this technology is fairly mature,
  - but only if you use it!
  - Be extremely careful about running unknown code on your computer, phone, and so forth!

Harder to Hide Information from the Computer Itself

Encrypted computing

- **hide** the **program and**/or **data** from the computer that uses it
- Still fairly undeveloped—mostly the trust here is handled through human trust and/or contracts/law.

A Better Solution: Use Virtual Machines (VMs)!

Late 90s/early 2000s – virtual machines re-emerge!
Connectix developed virtual machines

- to help consumers use software
- developed for other computers,
- such as **VirtualPC** (x86 virtual machines)
- as well as Sony Playstation emulation
- (Sony sued, lost, bought, and terminated)

VMWare was one of the first and most important in terms of cloud computing

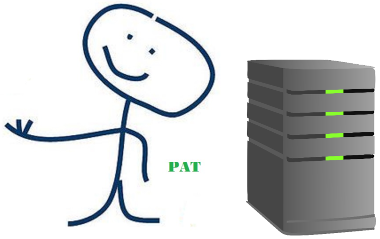
Sun VirtualBox (now Oracle) is still a good choice for home use.

Virtual Machines Enabled Cloud Computing

Today, Pat can provide a VM for Jan!

Hardware and software configuration completely **known in advance**!

VMs enabled cloud computing: use of someone else's computer as a service.



Virtual Machines also Made Sandboxing Accessible

VMs also enabled sandboxing at home.

- If affected by a virus,
- Wipe the virtual machine—no problem!

Today, Cloud Computing Offers Computers as a Service

Today, if you **want to use**
- a computer, or
- **1,000 computers, or**
- powerful **GPUs**,

for
- an hour, or
- a day, or
- a month, or
- on demand,

you can **rent from** one of many **cloud computing providers**!

Data Storage also a Service

Some cloud computing providers:
- Amazon Web Services
- Google Cloud
- IBM Cloud
- Microsoft Azure

These companies **also** offer to **store your data**, which
- **reduces need for** repeated data **transfers**,
- **But** also **locks you in** as their client:
- it's quite difficult to move Exabytes of data to another provider quickly.

Trust Issues? You Just Have to Trust Your Provider

Terminology You Should Know from These Slides

- client and server
- forwarding (a message or packet)
- virtual / overlay network
- HTTP: HyperText Transfer Protocol
- URL: Universal Resource Locator
- operating system (OS)
- virtual machine (VM)
- cloud computing
- sandboxing
- lock-in (by a company, product, or service)

Concepts You Should Know from These Slides

- examples of Internet services (and clients), including Web servers (and browsers)
- roles for the OS: abstract, virtualize, and protect
- Issues resolved by virtual machines and cloud computing: Use on demand and security
- issues for cloud computing: trust and variability/compatibility