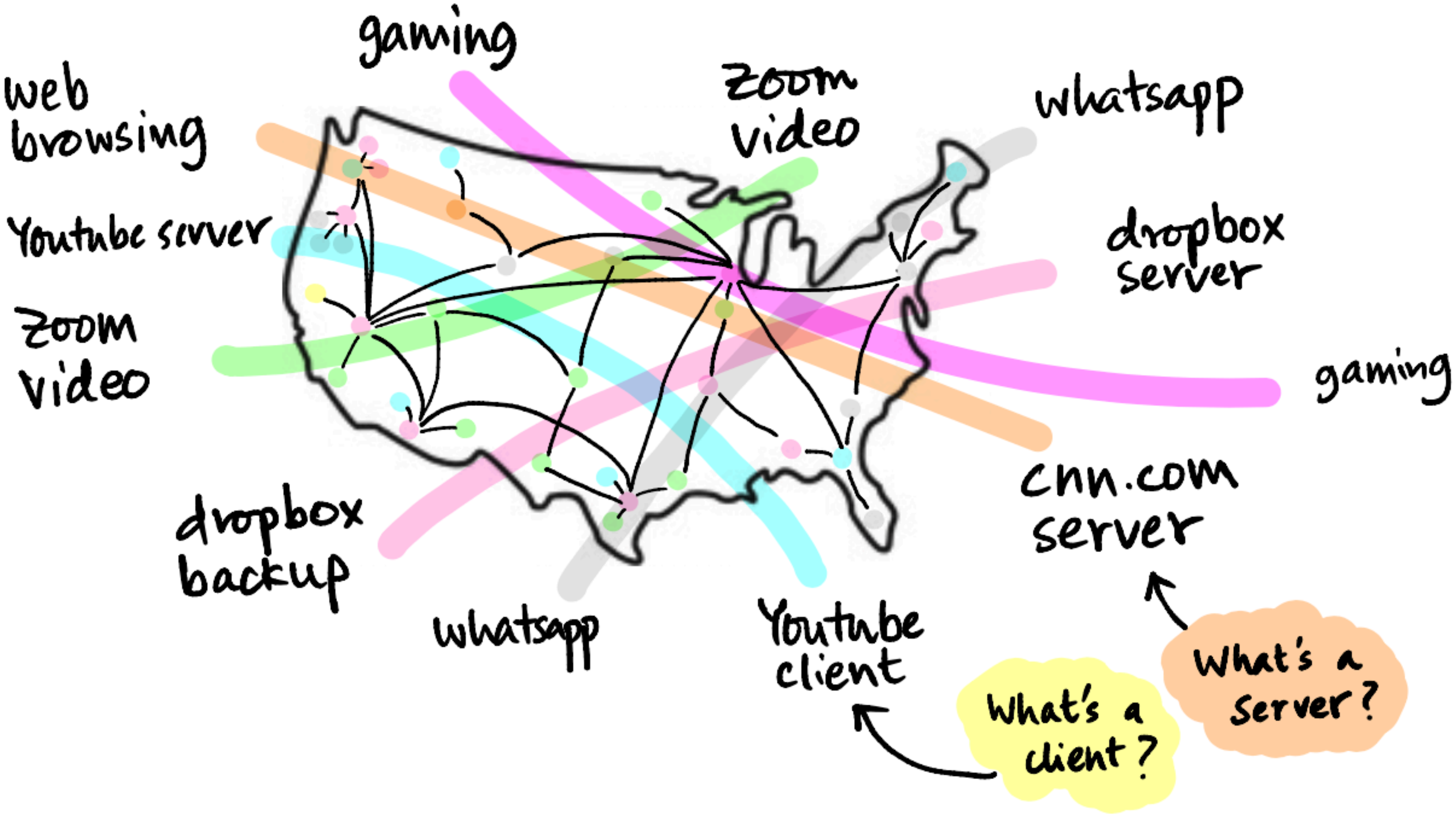
University of Illinois at Urbana-Champaign
Dept. of Electrical and Computer Engineering

ECE 101: Exploring Digital Information Technologies for Non-Engineers

Client and Server; Distribution and Streaming

Applications running on the "edge"



Client and Server

A Server Provides Some Sort of Service

Some computer may **provide a certain service**, such as

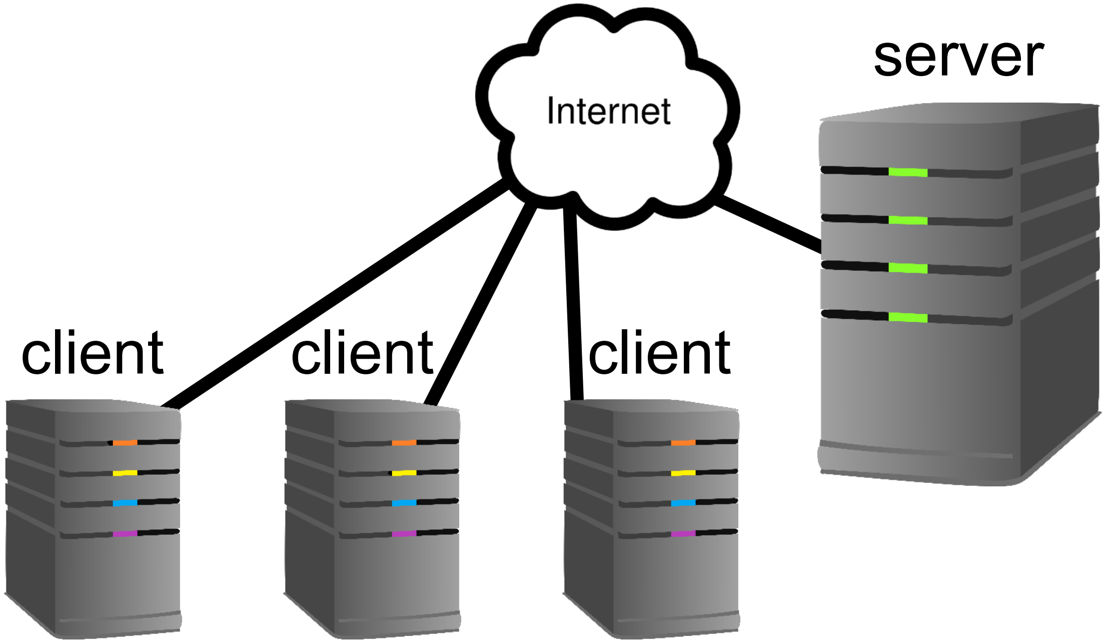
- providing copies of published IRS tax documents,
- accepting paper submissions to a research conference, or
- computing turbulence in fluid flow around a structure,

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.

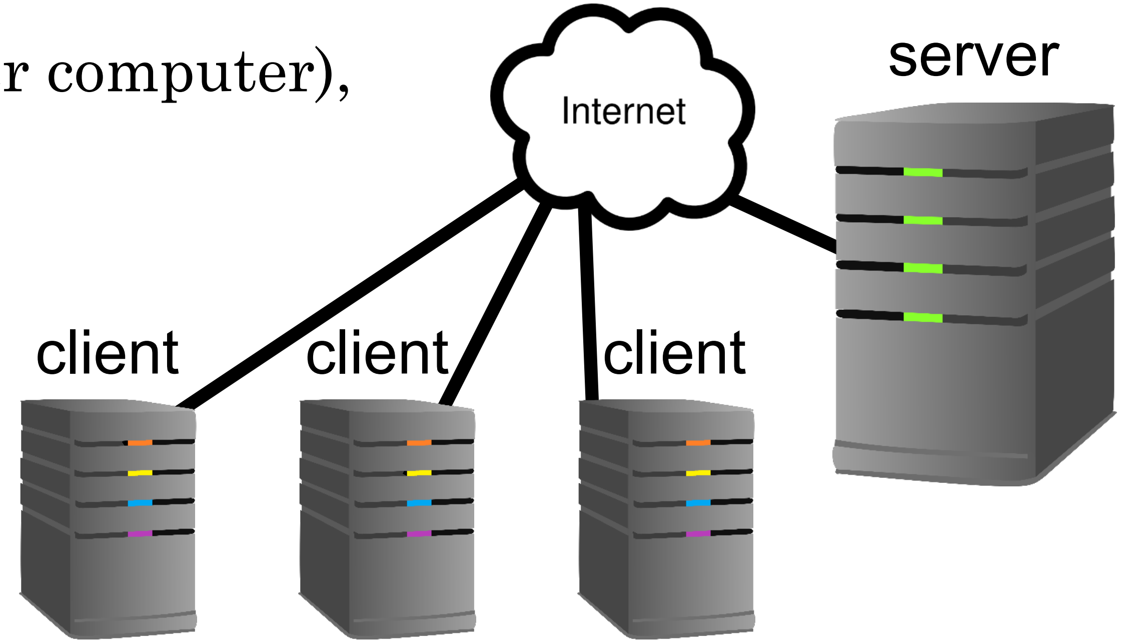


Any Computer Can Provide or Use a Service

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.



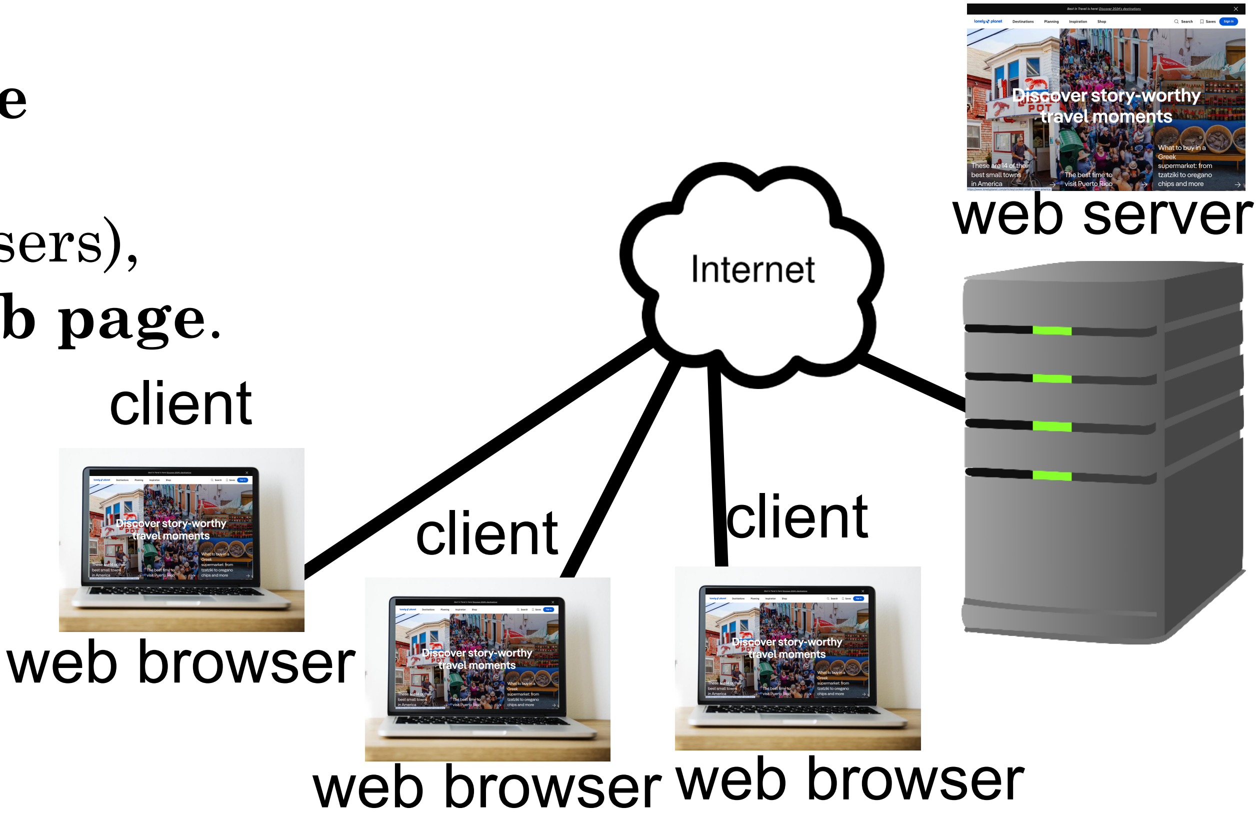
Example of a Service: the World Wide Web

The **server**

- **offers the web page**

The **clients** (web browsers),

- **want to see the web page.**



web server

Internet

client

web browser

client

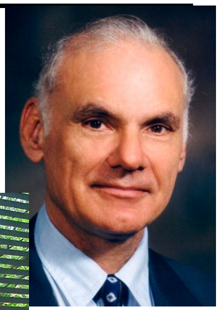
web browser

client

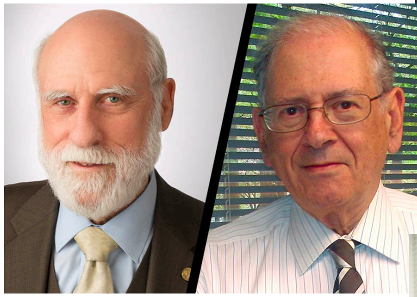
web browser

Aside: What good was the Internet if no one used it?

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



Mid 1970s - Vint Cerf and Bob Kahn builds 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

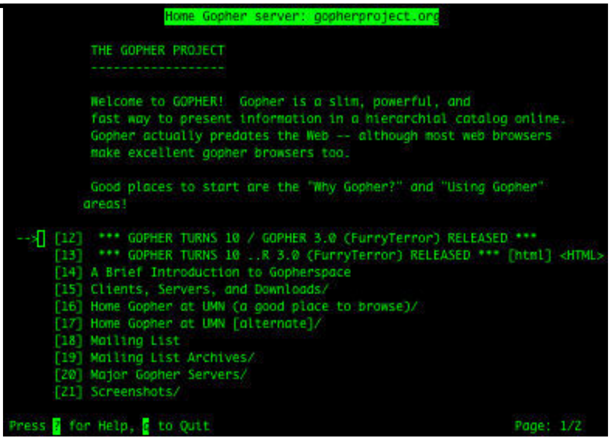
Examples of Early Services: Gopher and HyperCard

Early Internet services such as **Gopher**

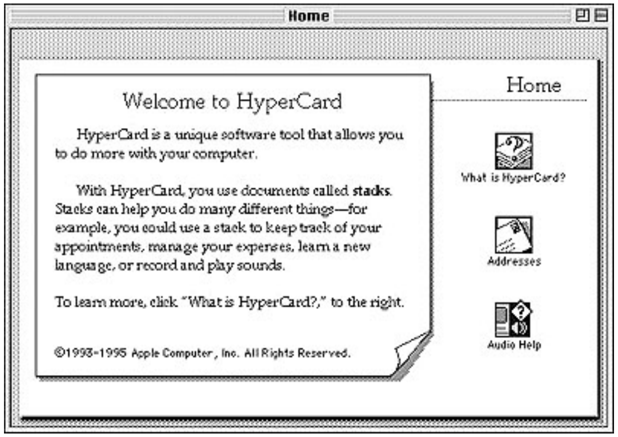
- allowed clients to **explore text documents**
- spread **across multiple servers**
- such as guidance on the requirements for undergraduate curriculum at UIUC.

Apple’s **HyperCard**

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



```
                    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
```



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).

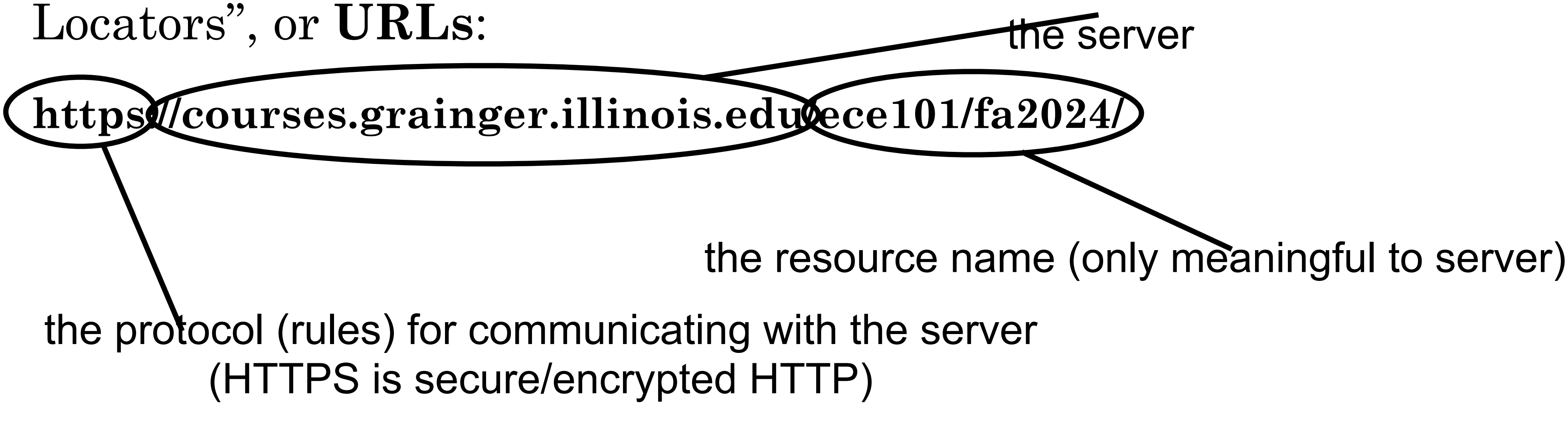
The **first web browser** (integrating images with text), **Mosaic**, was **developed by Marc Andreessen** and others (at UIUC) in 1992, and made public in 1993.

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

HTTP Protocol (Perhaps Familiar to You?)

HTTP specifies how online resources offered as **Internet service** by a **web server** can be found.

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



https://courses.grainger.illinois.edu/ece101/fa2024/

the server

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 couple of lectures, 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.

The World Wide Web is NOT the Internet

(Some people may call the WWW “the Internet”)

But not students of ECE101…

The Web is an **internet service** - consisting of websites, pages and other web services scattered around the world on many different computers acting as web servers.

It is provided on the **distributed network of computers** (clients, servers, routers, etc.) which is the **Internet**.

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/responses** 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)?**

Sharing a Valuable Resource

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!).

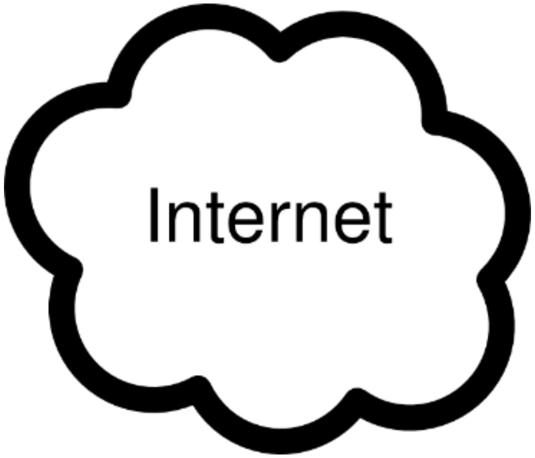
Why not create a model in which someone could use a computer for a little while, whenever they needed one?

Computing as a Service … in the “Cloud” !

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

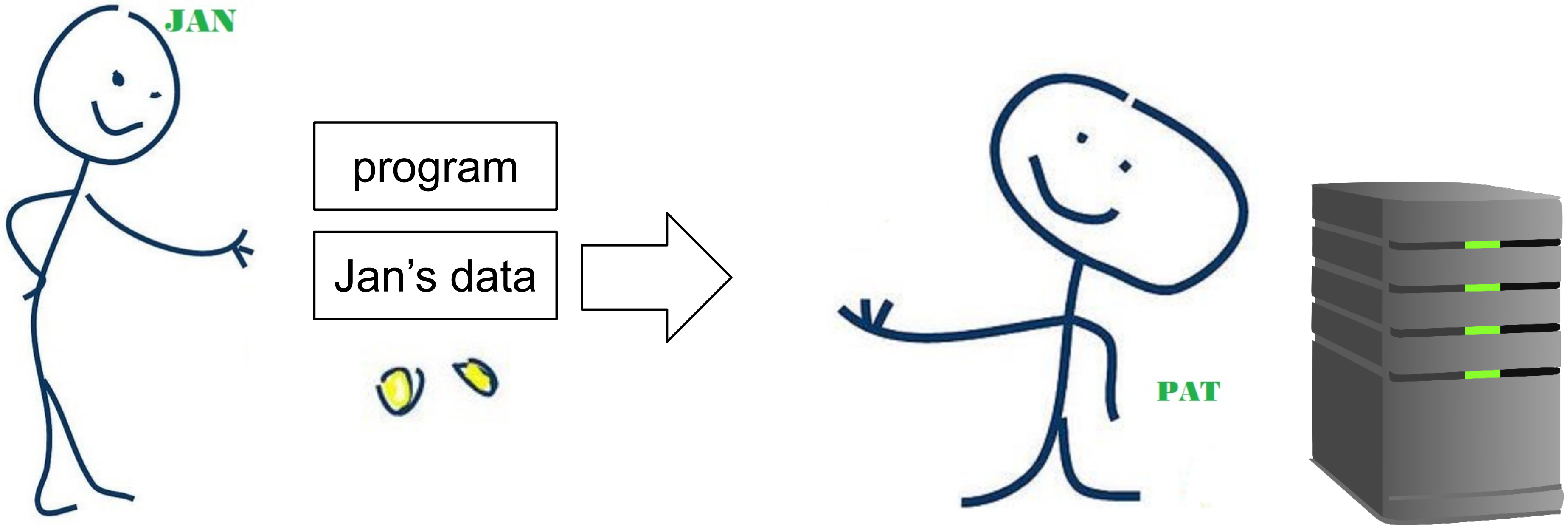
Not just individuals:

- 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.



program

Jan's data

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!

program

Jan's data

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

This technology is fairly mature (but only if you use it!)

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.

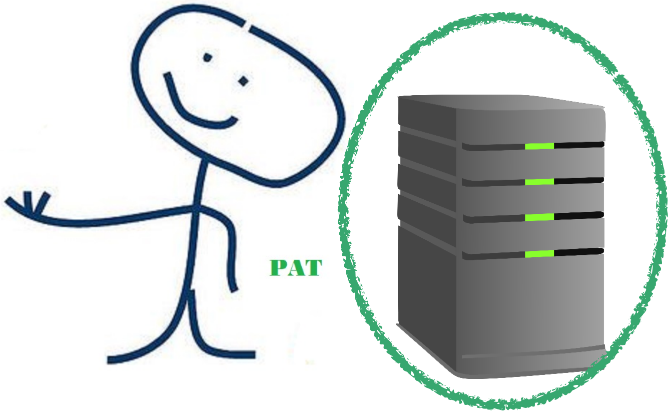
Virtual Machines Enabled Cloud Computing

A Virtual Machine is a software that mimics a computer

- A computer is hardware.
- Write software to do the same thing as hardware - so it can pretend to be the hardware.

Jan provides **hardware and software** configuration **in advance!**

Pat creates a VM for use by Jan. If anything goes wrong in the VM, Pat's computer is still fine.



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:

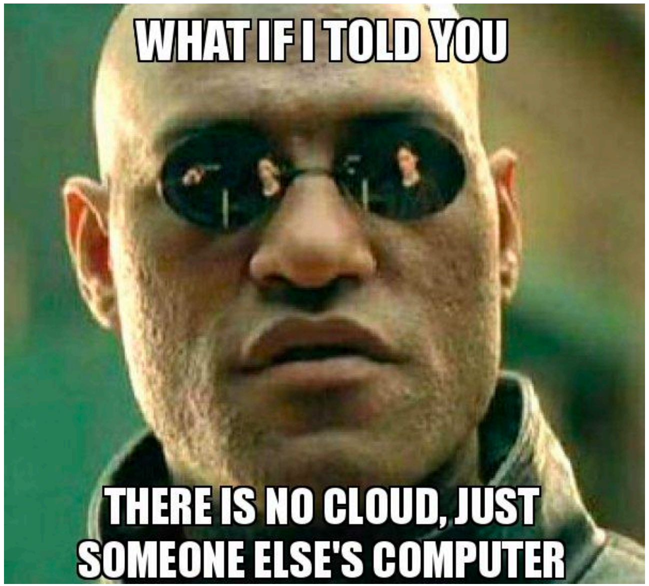
- Amazon Web Services
- Google Cloud
- IBM Cloud
- Microsoft Azure

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

Is it just someone else’s computer?

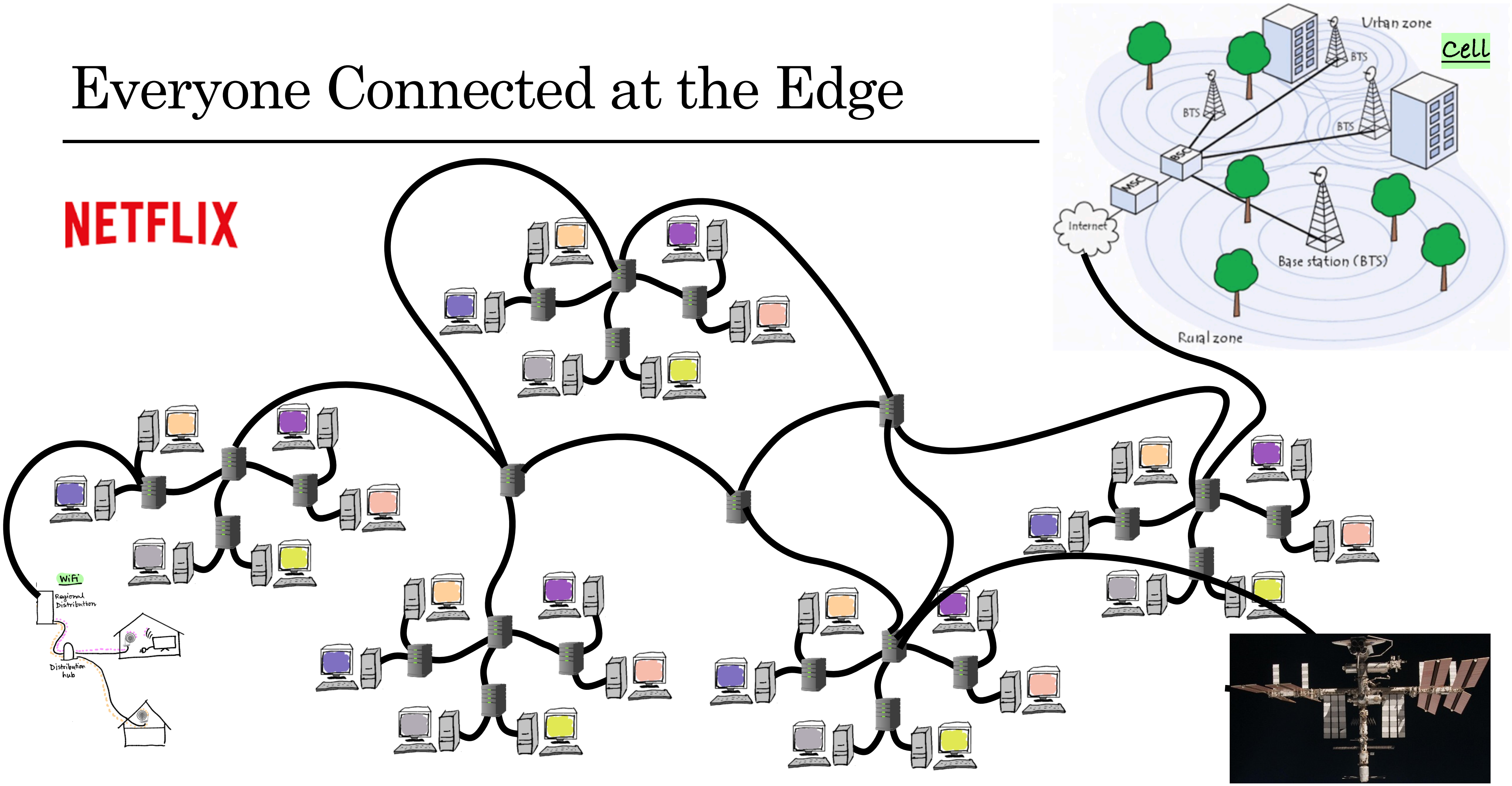


https://makeameme.org/meme/what-if-i-s1sk9l

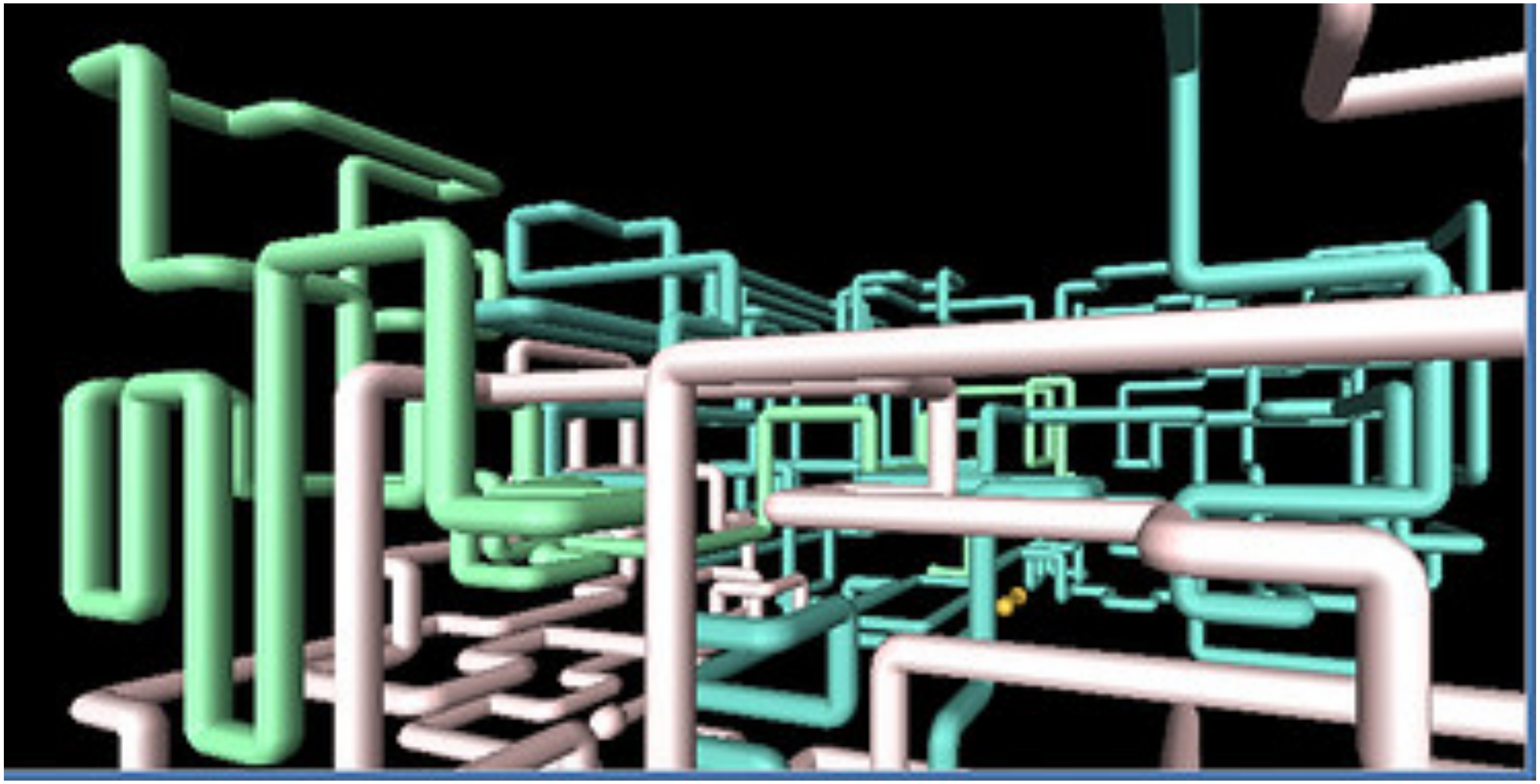
The Cloud is much more - a complex system of interconnecting parts.

Distribution and Streaming

Everyone Connected at the Edge



Network Links Can Be Viewed as Pipes for Data



https://www.flickr.com/photos/jvmanna/2844048090

Network Links Can Be Viewed as Pipes for Data

Imagine a pipe that carries water.

Now imagine that we have

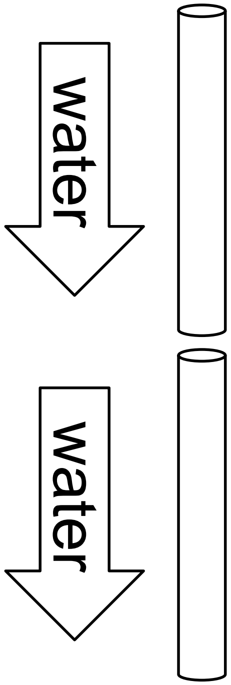
- **10 gallons per minute**
- **flowing** through the pipe.

Next, add a **second pipe**

- **with** the **same flow** rate
- and put it above the first pipe.

No problem, right?

Water from the top pipe goes into the bottom pipe.

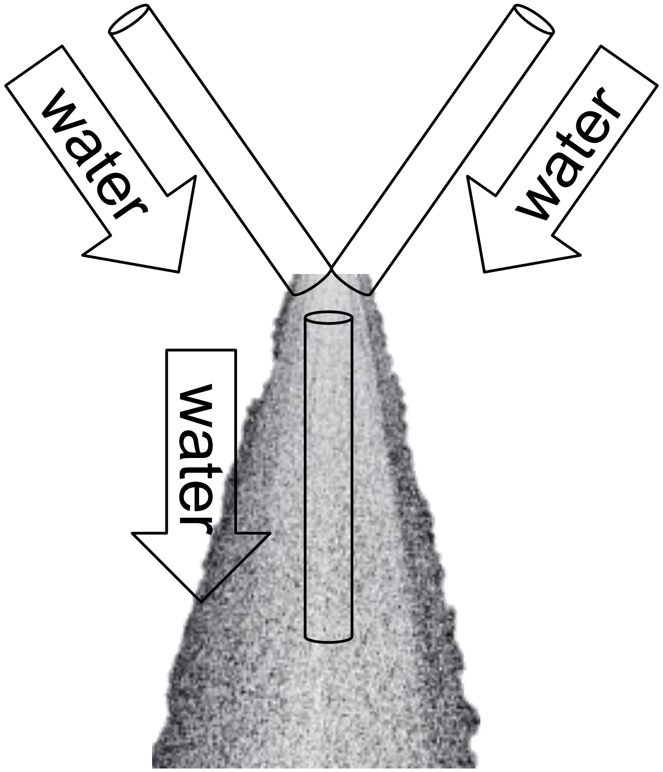


What Happens When a Pipe Can't Handle the Water?

Now let's **add a third pipe…**
with the same flow of water.

Now what happens?

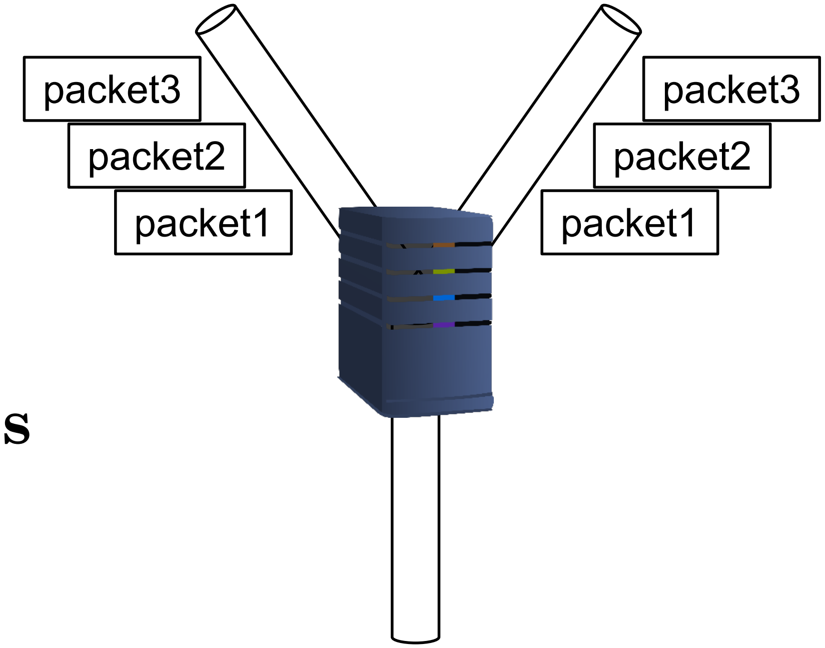
Oops! **Water spills** everywhere!



Similarly, Internet Routers May Discard Packets

The **same thing happens**

- **when** an Internet **router**
- **receives too many packets**
- for one outgoing link.



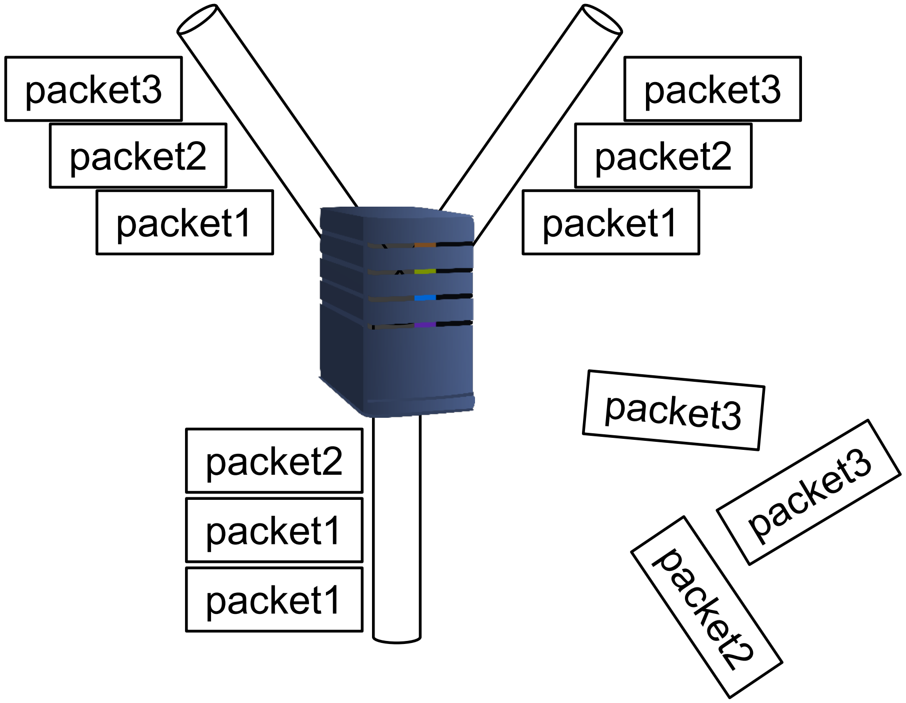
Remember that Internet Only TRIES to Deliver a Packet

That’s one reason that

- Internet **packet delivery**
- **is unreliable**.

So that routers

- can discard packets
- without causing problems.



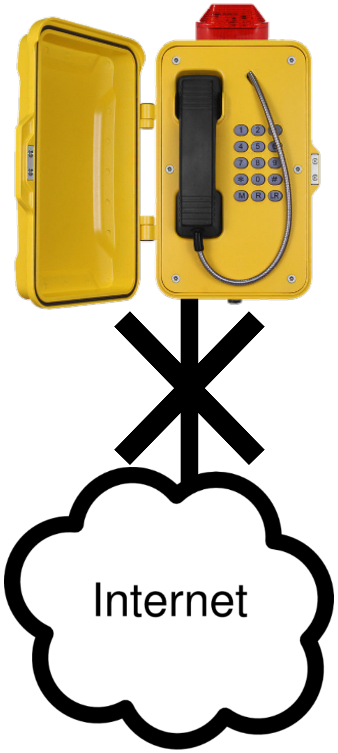
Internet Service Definition Does Not Suit All Needs

A brief aside…

Unreliability raises some major issues that we won't cover.

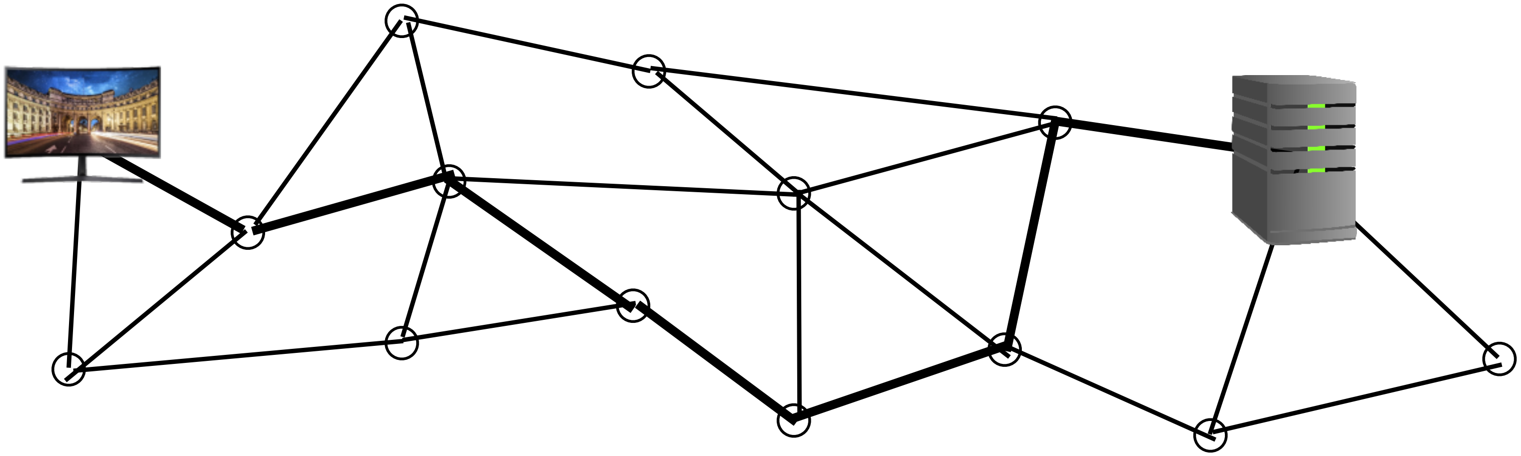
For example, the **Internet**

- **does NOT make a good substitute for 911**
- or any other campus/state/country's emergency telephone services.



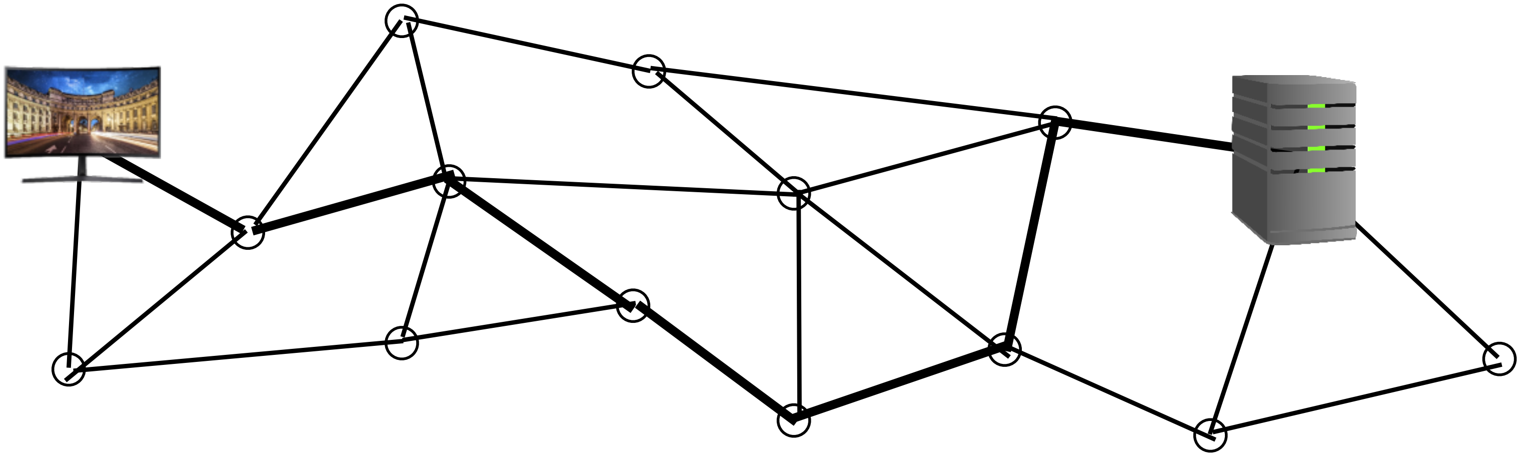
How Quickly Can a Client Communicate with a Server?

How much data can a client send through the Internet to a server each second?



How Quickly Can a Client Communicate with a Server?

The answer changes constantly
as other computers (TCP connections)
use the Internet.



TCP Tries to **Estimate** How Much Data to Send

TCP constantly tries

- to **estimate** the **achievable data rate**
- **in** bits or **bytes per second**,
- **called bandwidth**.*

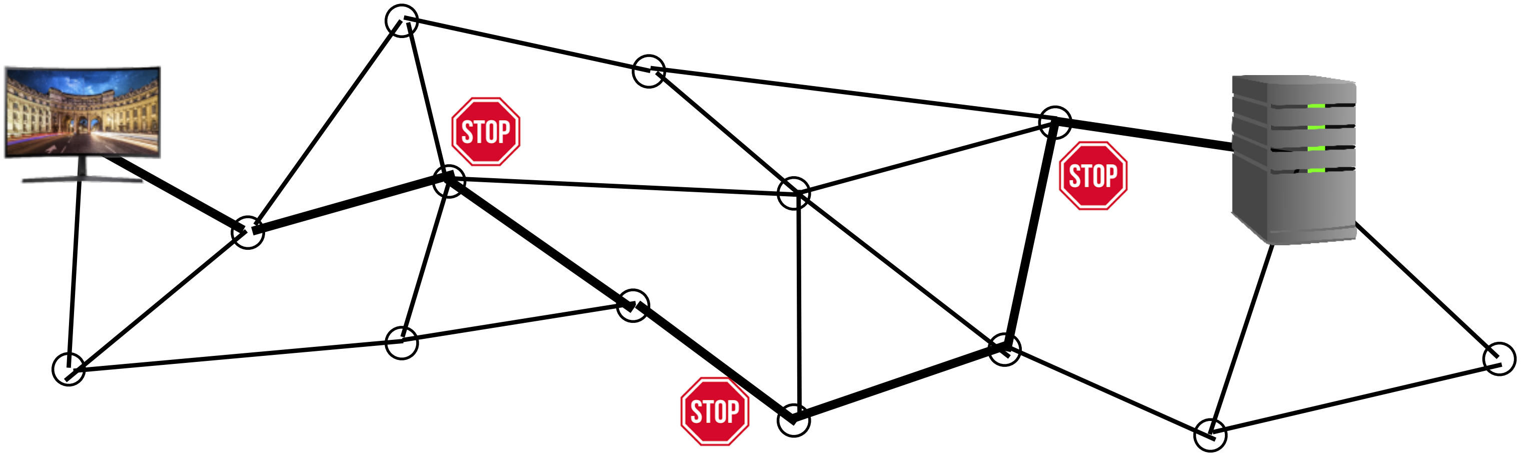
Information moves at nearly the speed of light:

~**130 msec** to the other side of the Earth

*Electrical engineers object to this usage because the term is also used to measure ranges of frequency.

TCP Tries to **Estimate** How Much Data to Send

More delay means a worse estimate!!!



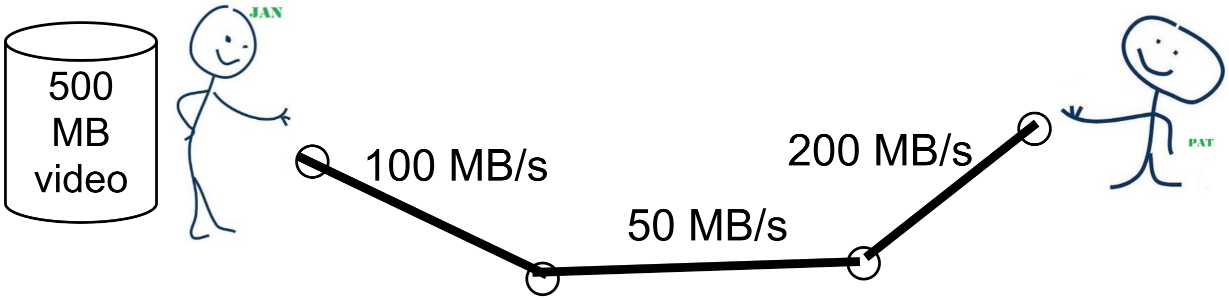
Let’s Quantify a Single Transmission

Let’s **quantify a simple problem** with our friends Jan and Pat.

Jan wants to **send** Pat a **500 MB** video.

The route between them goes **over three links**.

Each link has limited **bandwidth**.



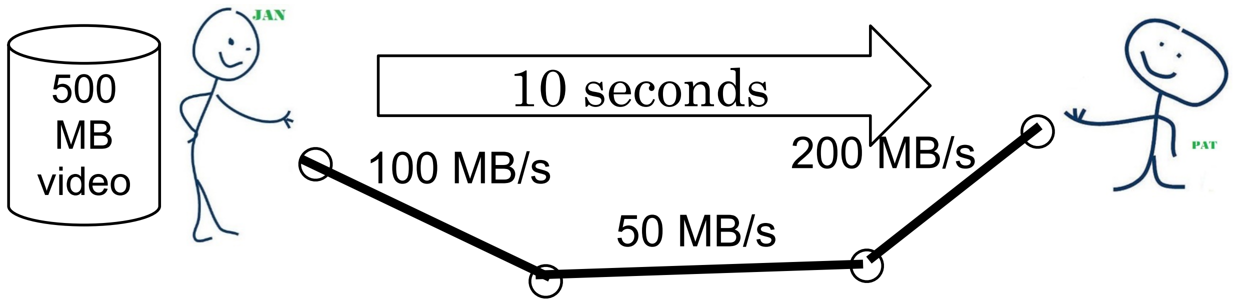
Let’s Quantify a Single Transmission

What is the bandwidth from Jan to Pat?

50 MB/s – the **minimum of** the **link bandwidths** on the route through the Internet

So how long does sending the video take?

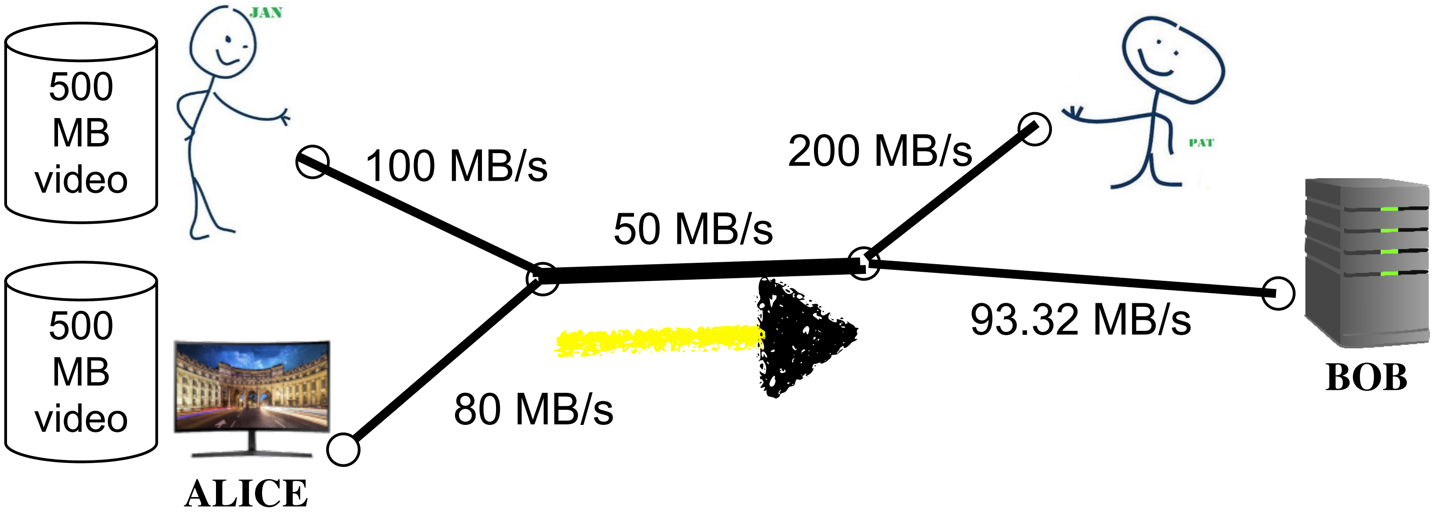
$$\frac{500\ \text{MB}}{50\ \text{MB/s}} = \mathbf{10\ seconds}$$



Connections Share Links in the Internet

What happens if Alice also sends Bob a video?

The **two transmissions** must **share** the middle link!

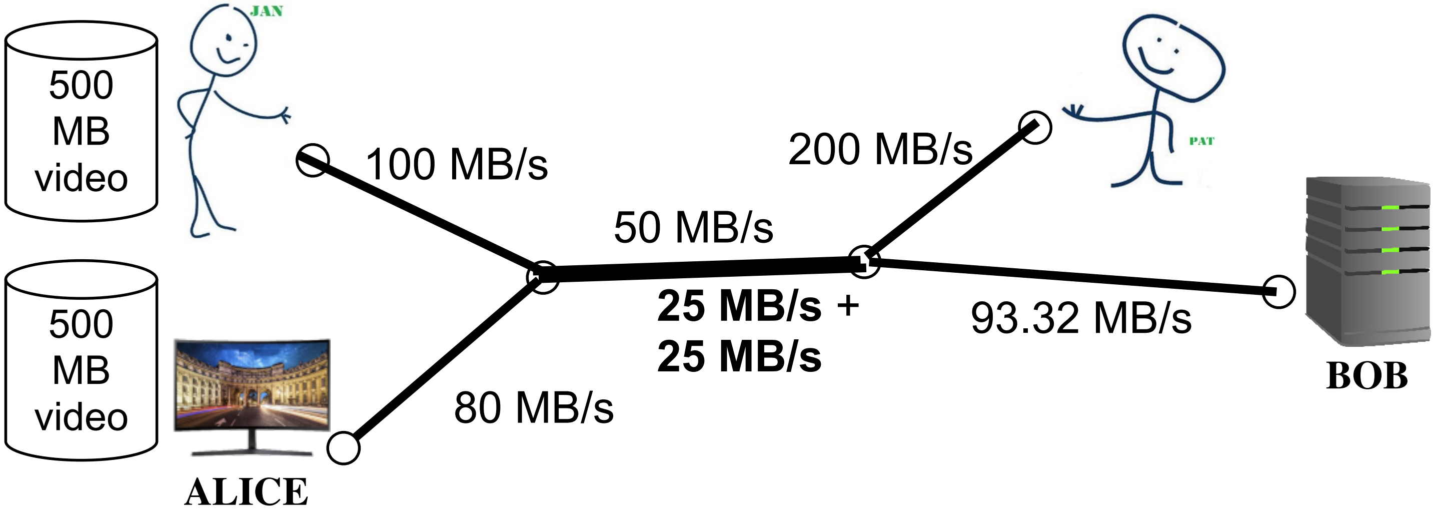


Each Connection May Use an Equal Share

If they share fairly, what is the link bandwidth for each connection?

$$\frac{\text{50 MB/s}}{\text{2 connections}} = \textbf{25 MB/s}$$

for each connection



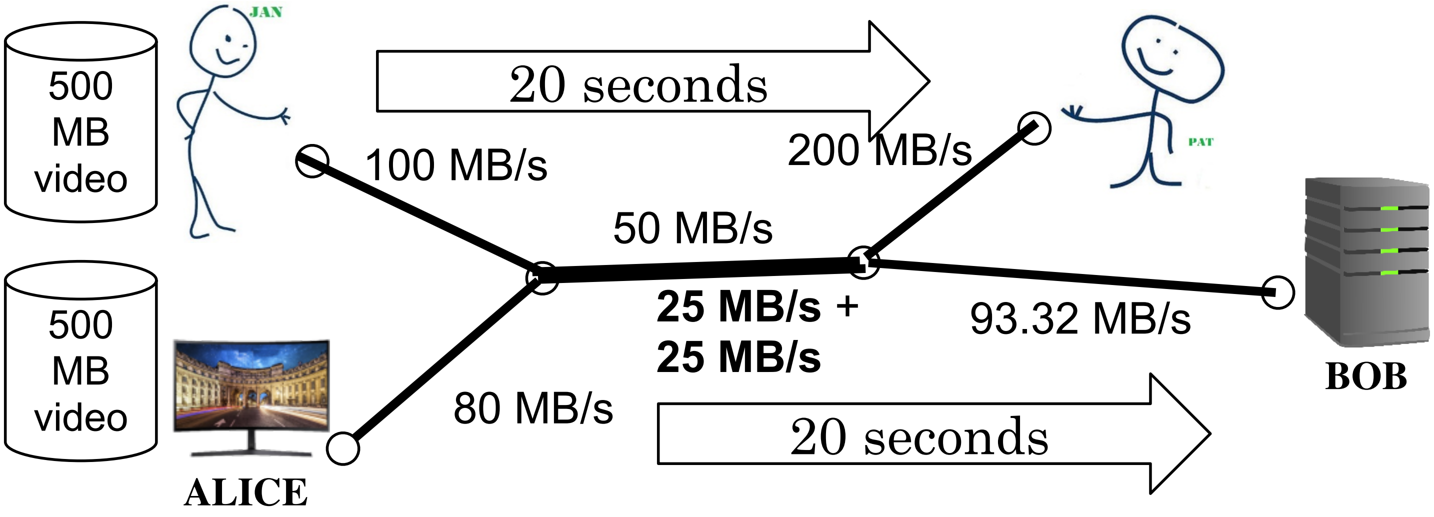
Shared Links Slow Down Transfers

What is Jan's connection bandwidth? **25 MB/s**

What about Alice's? **25 MB/s**

And how long to send either video (same size, same bandwidth)?

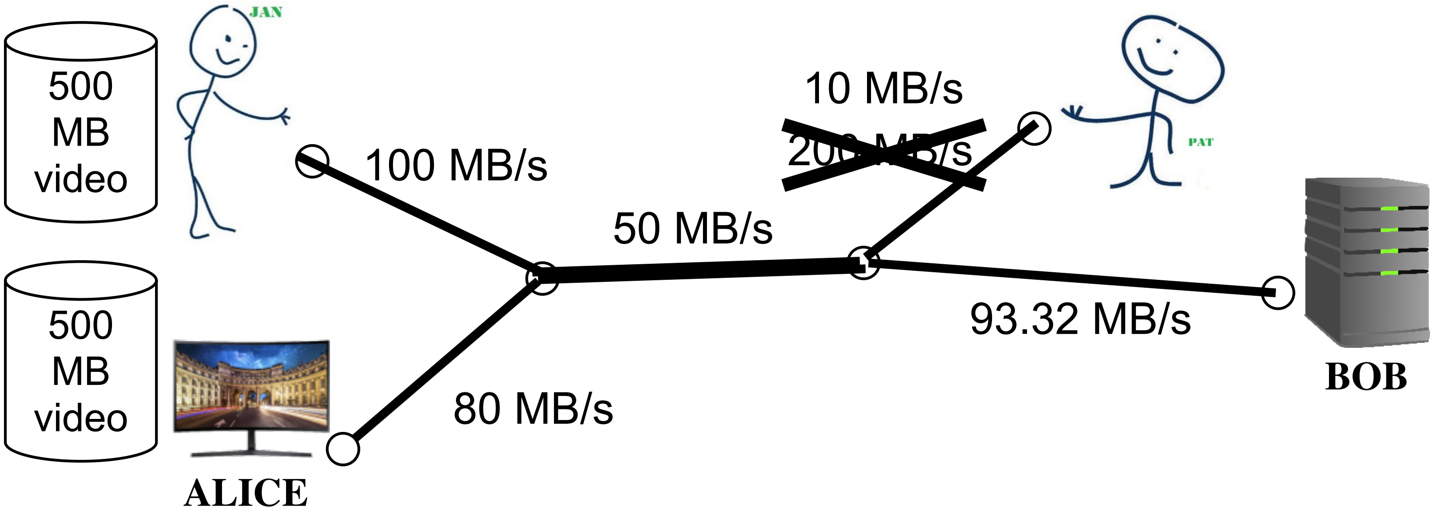
$$\frac{500\ \text{MB}}{25\ \text{MB/s}} = \textbf{20 seconds}$$



Another Scenario: a Low-Bandwidth Connection

Let’s change the situation:

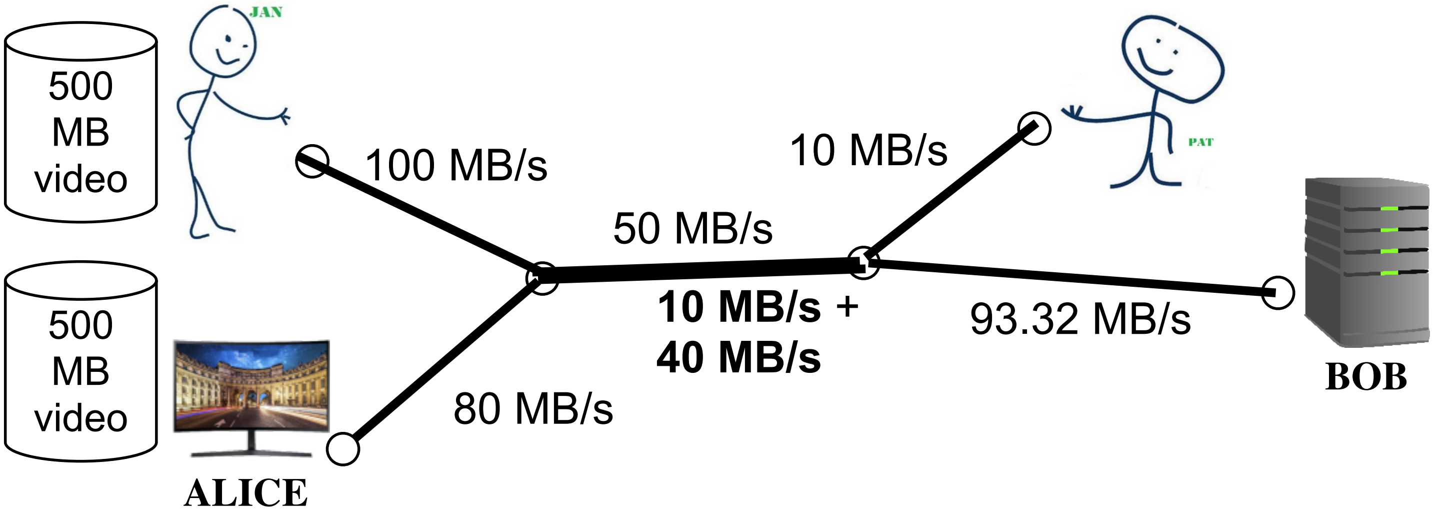
- Pat is now using his phone.
- The last Jan-to-Pat link carries only **10 MB/s**.



Another Scenario: a Low-Bandwidth Connection

What is Jan's connection bandwidth? **10 MB/s**

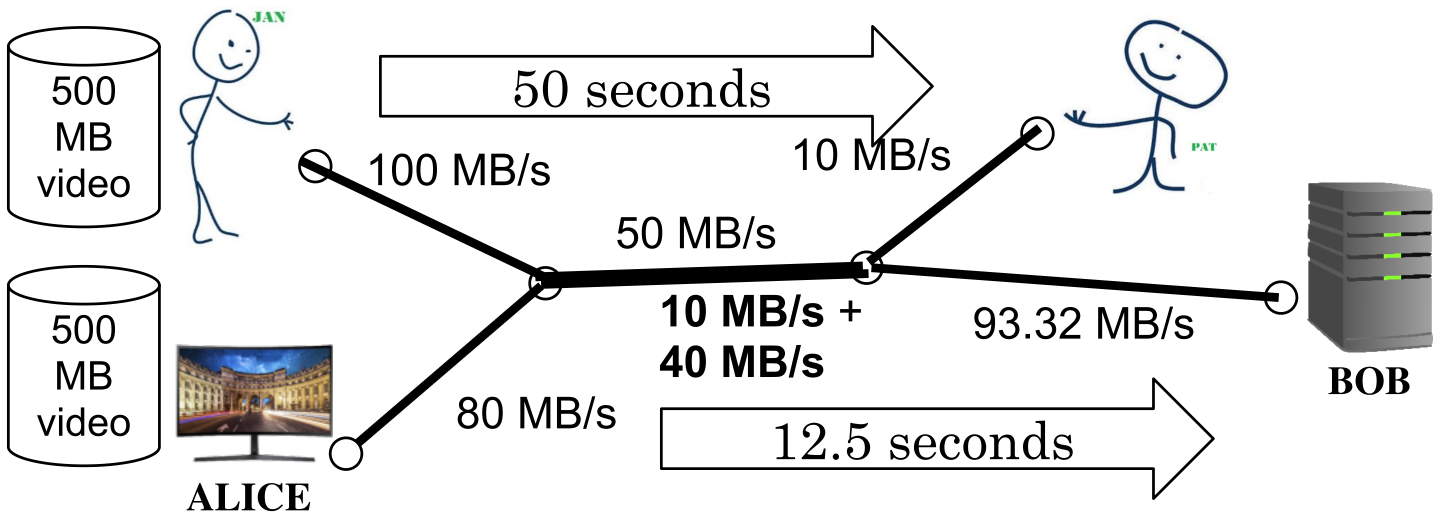
Alice can use the rest of the middle link's bandwidth!
So what is Alice's connection bandwidth? **40 MB/s**



Another Scenario: a Low-Bandwidth Connection

How long does Jan's transfer require? $\frac{\text{500 MB}}{\text{10 MB/s}}$ **=50 seconds**

What about Alice's transfer? $\frac{\text{500 MB}}{\text{40 MB/s}}$ **=12.5 seconds**



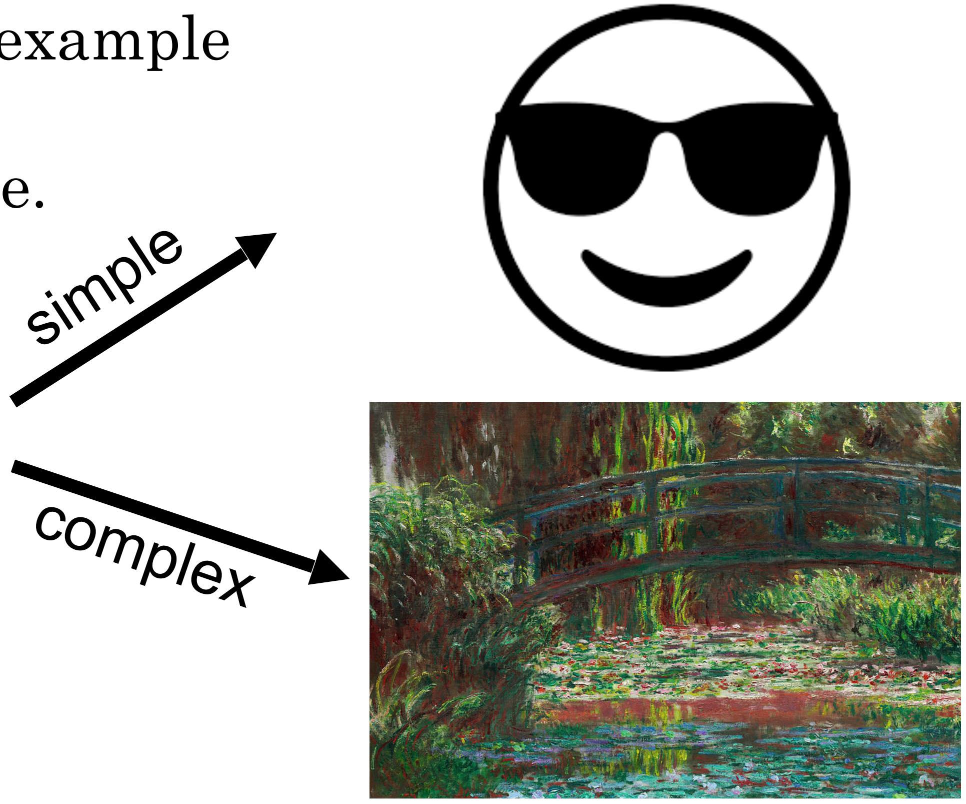
Both Video and Network Rates Vary Over Time

The (high) bandwidths in our example

- are more than enough
- to deliver a video in real-time.

However,

- the **rate of data needed** by a video **varies**,
- and the **bandwidth of** the **network varies**.



Buffering Helps Avoid Need for Video Stalls

To handle variability, video players **use** a technique called **buffering**.

- Before the video starts playing, your computer downloads the first second of video.
- While the first second plays, the computer downloads the second second.
- And so forth.

Buffering Can’t Hide Inadequate Bandwidth

Often, a video player starts downloading before you press “Play.” That way, the video seems to start instantaneously.

But **you’ve probably noticed occasional delays**

- for ads, or even for videos,
- **when** network and/or video content **variability**
- **or insufficient** network **bandwidth**
- **made your computer run out of video** to show you!

Terminology You Should Know from These Slides

- client and server
- Internet service
- HTTP: HyperText Transfer Protocol
- URL: Universal Resource Locator
- cloud computing
- Virtual machine
- Sandboxing
- lock-in (by a company, product, or service)
- bandwidth
- buffering

Concepts You Should Know from These Slides

- The Internet is NOT the web
- examples of Internet services (and clients), including
  - Web servers (and browsers)
  - Cloud computing (and its users)
  - Data storage (and its users)
- issues resulting from cloud computing: trust and variability/ compatibility
- routers can drop packets
- connection bandwidth is the minimum over all links in a route
- bandwidth is shared between connections
- more delay means less achieved bandwidth
- transfer time = (# bytes) / (achieved bandwidth)