CSCI 1680: Computer Networks

Nick DeMarinis

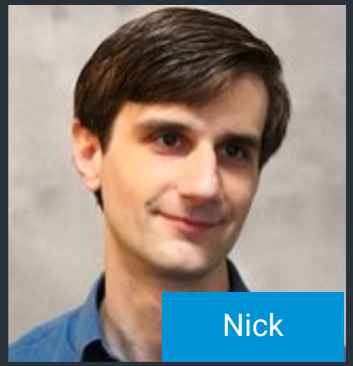
https://brown-csci1680.github.io

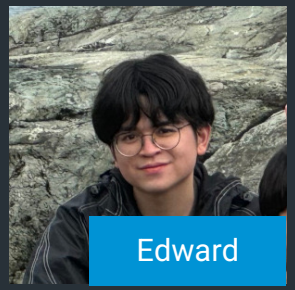
https://feedback.cs1680.systems/main/

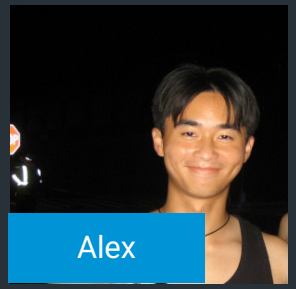
For anonymous questions & feedback during lecture:

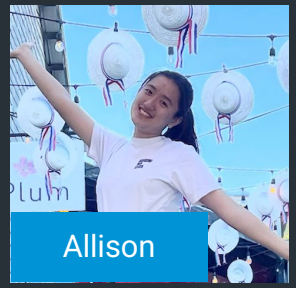
Based partly on lecture notes by Rodrigo Fonseca, David Mazières, Phil Levis, John Jannotti, Peterson & Davie

Cast

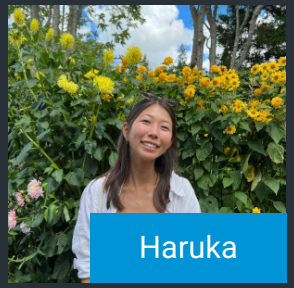

Nick

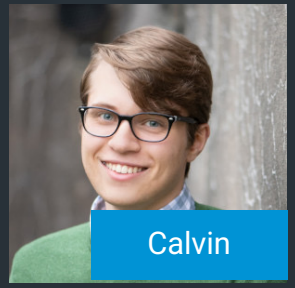

Edward

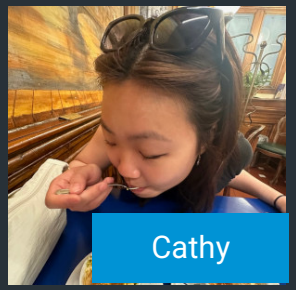

Alex


Allison

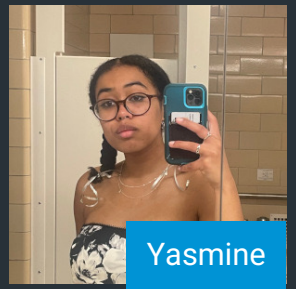

Brandon

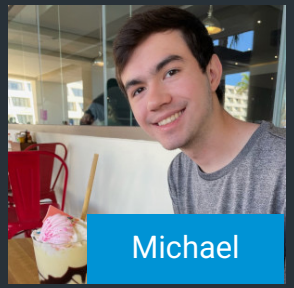

Haruka


Calvin


Cathy


Jason


Yasmine


Michael

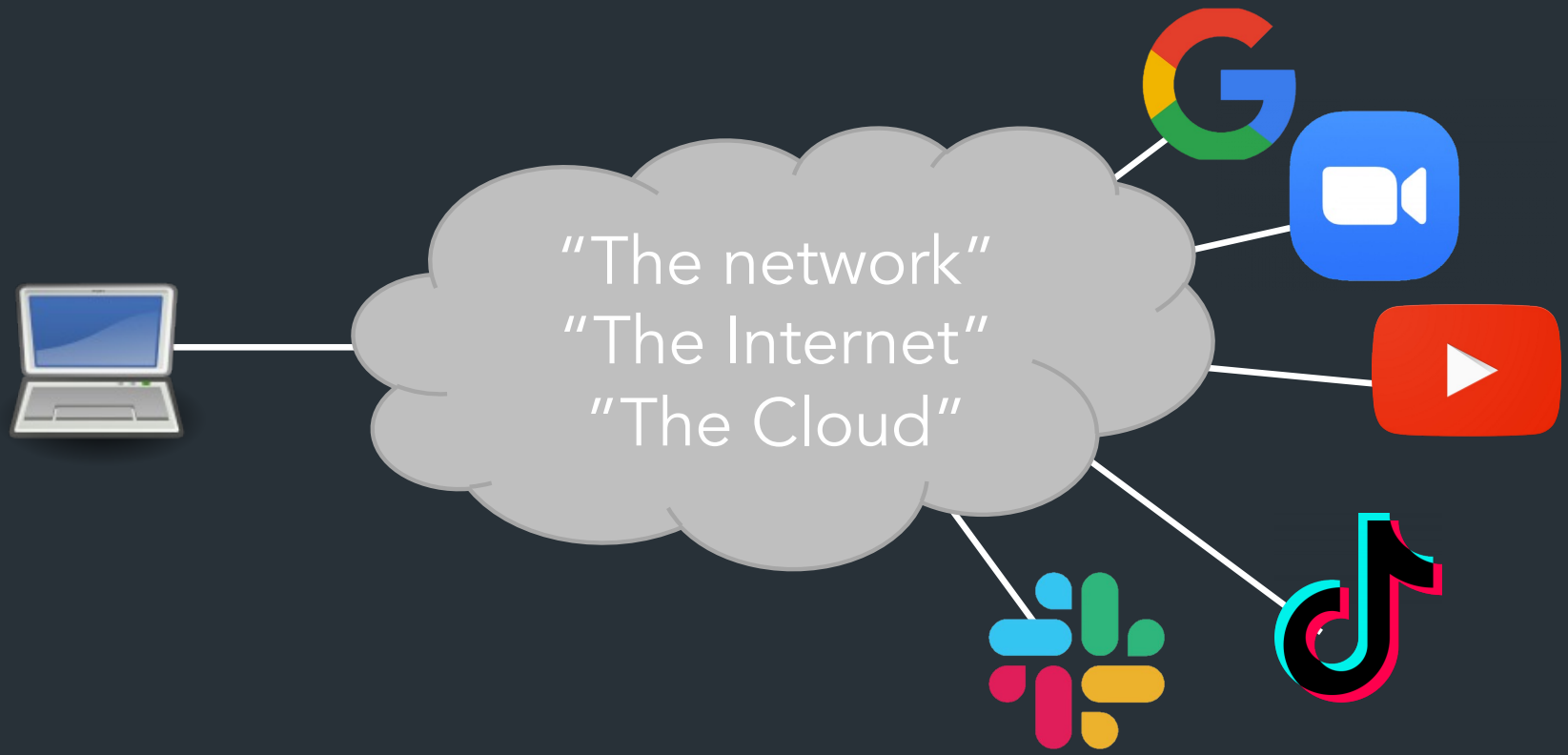

Yi Hao

Why are we here?

You (the user)

Applications

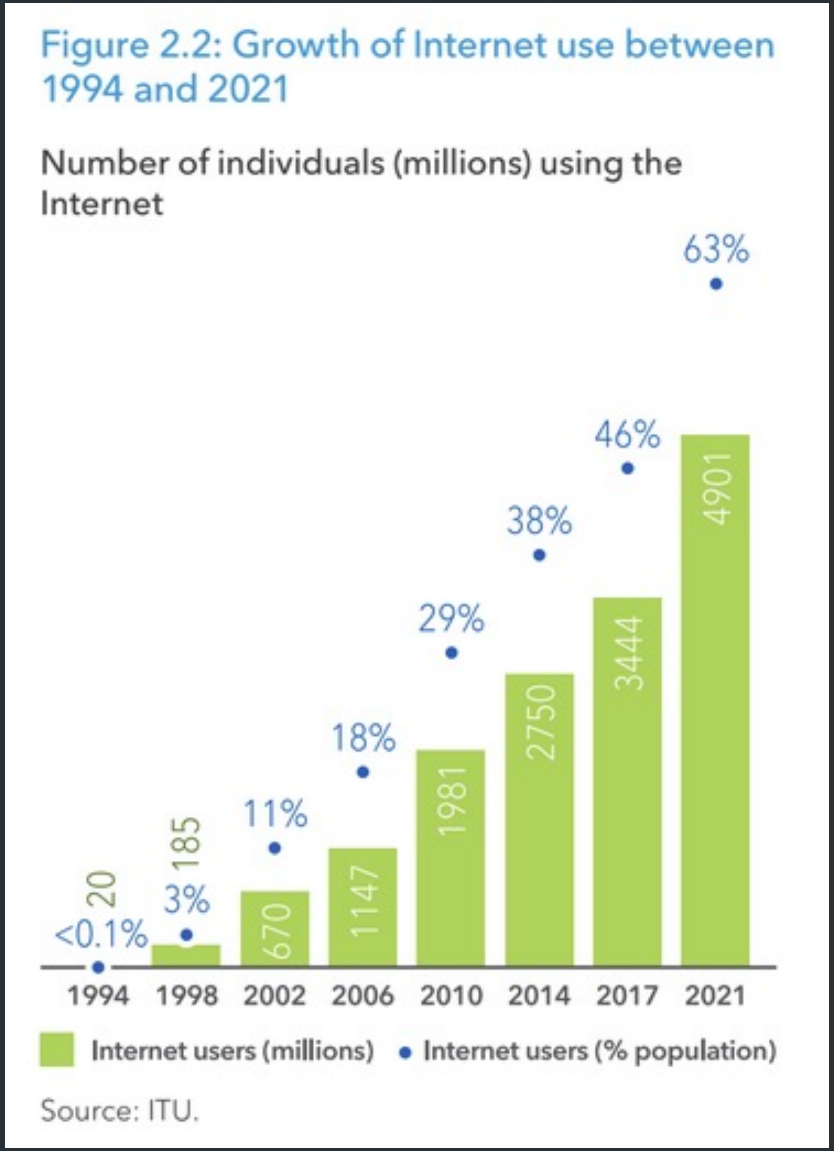
Why are we here?



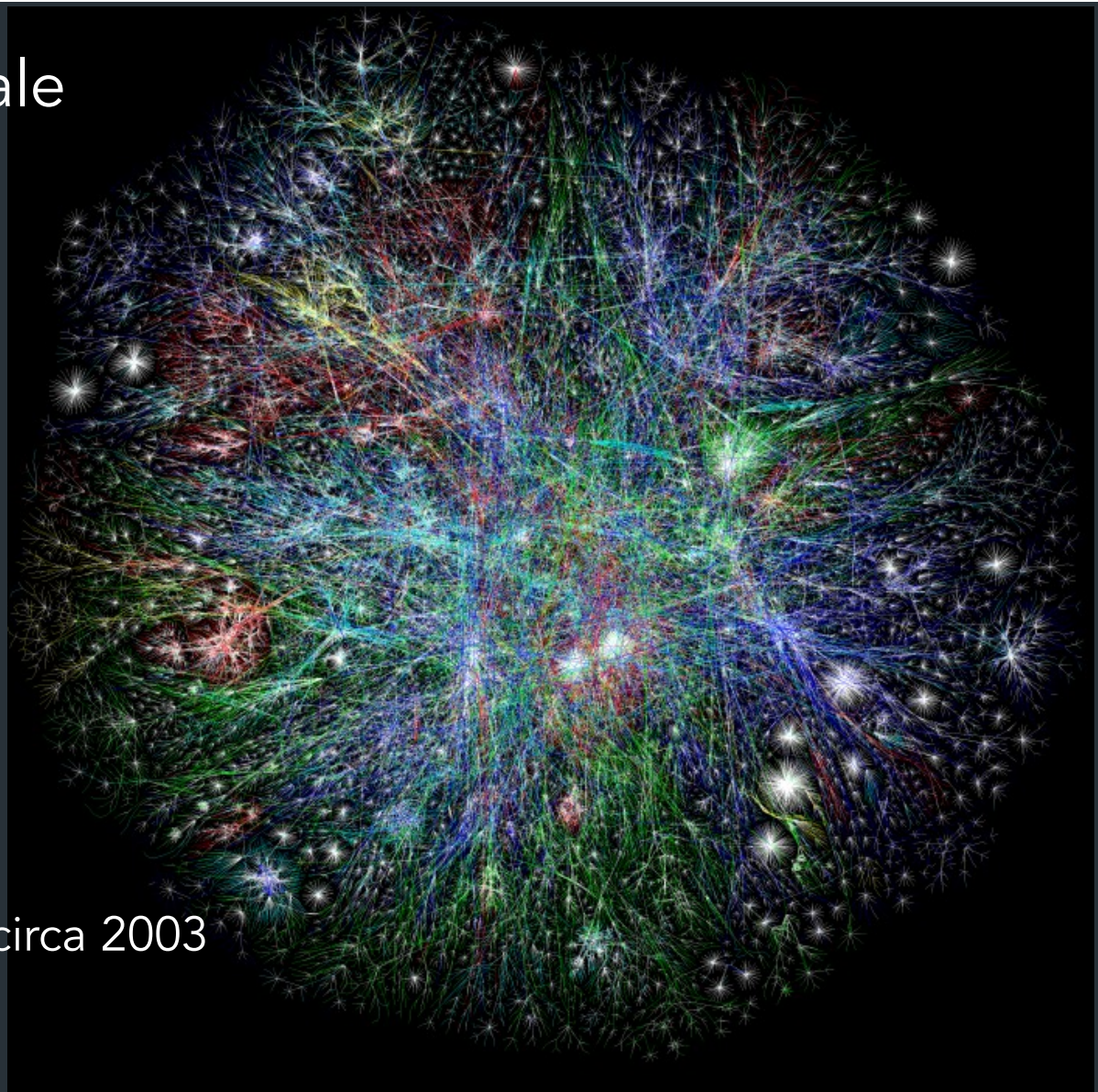
You (the user)

Applications

Challenge: Scale

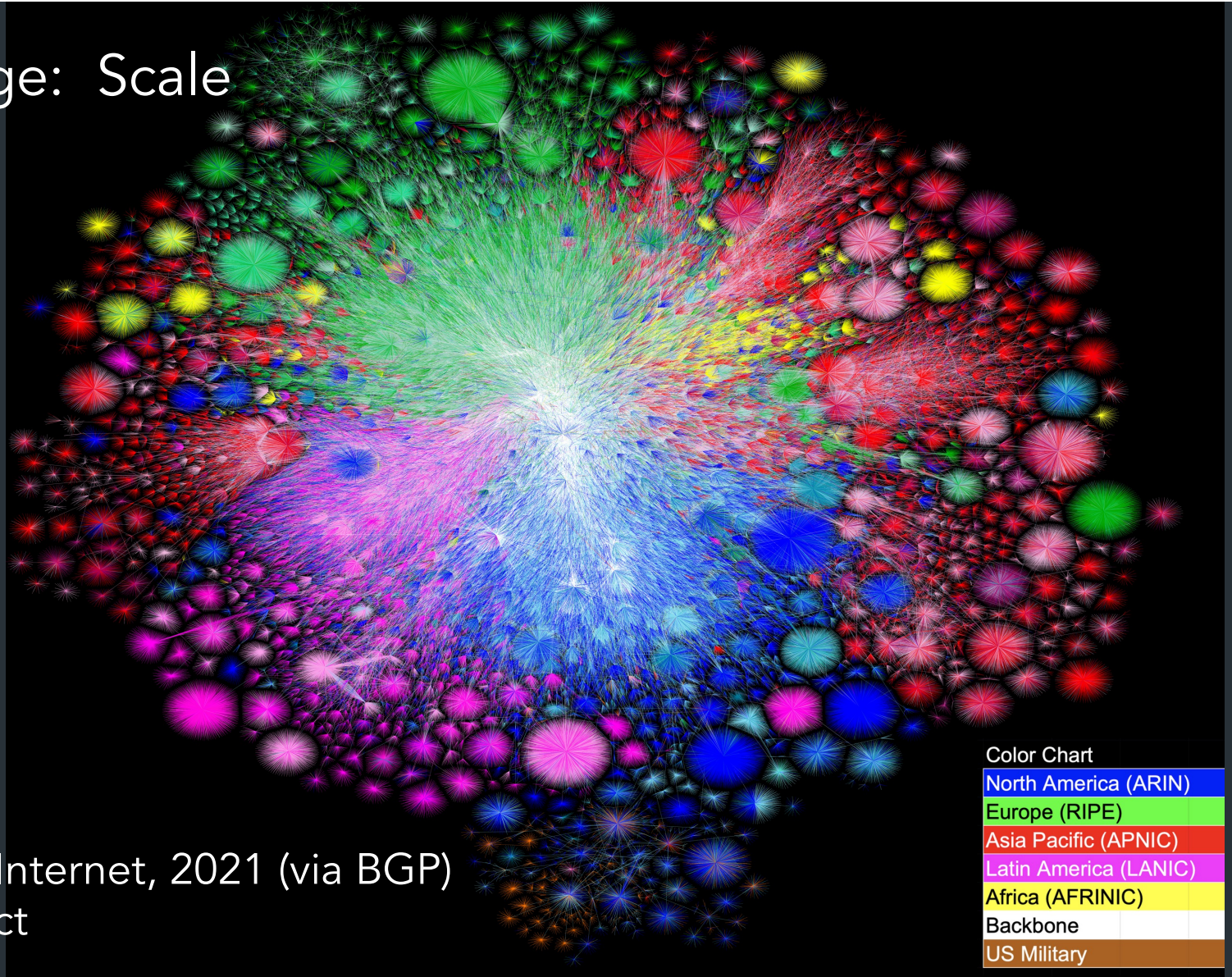


Challenge: Scale



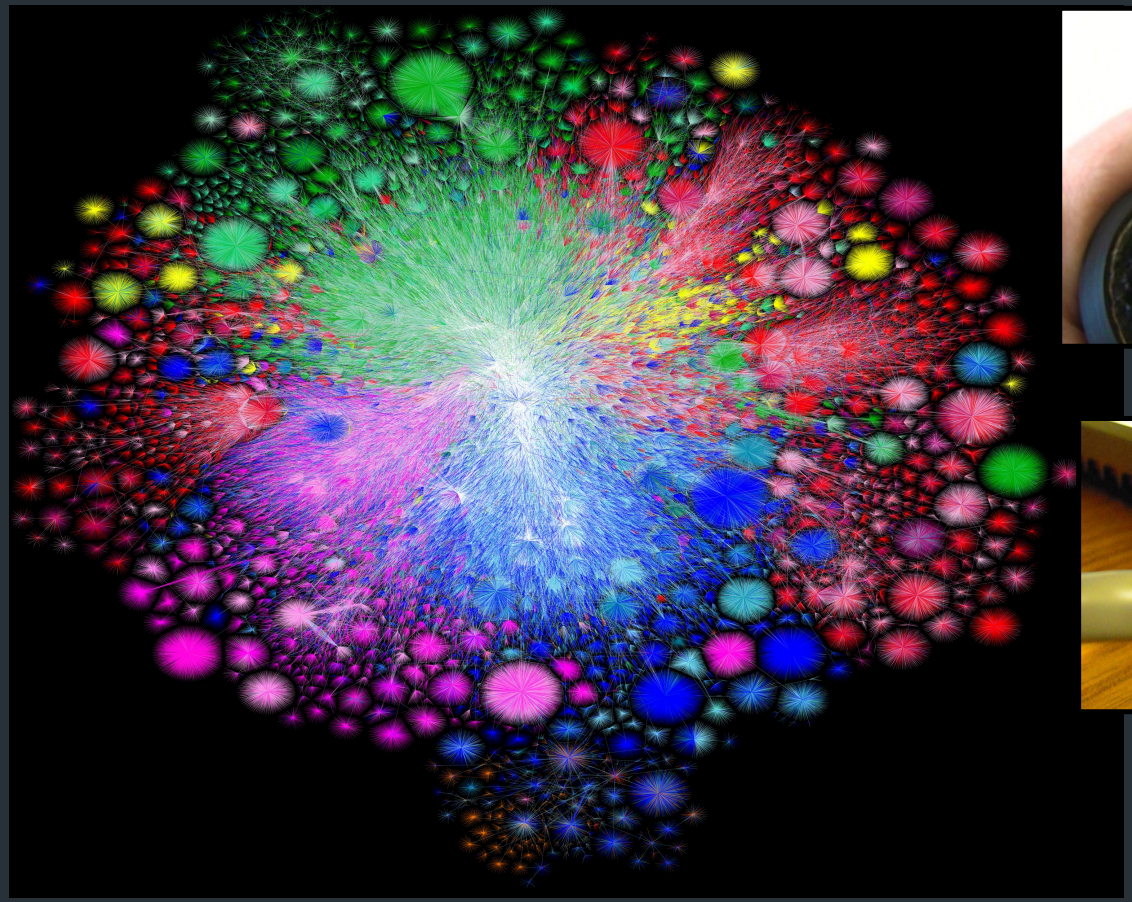
Map of the Internet, circa 2003
(via Traceroute)
OPTE project

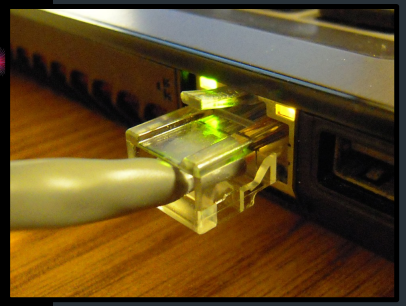
Challenge: Scale

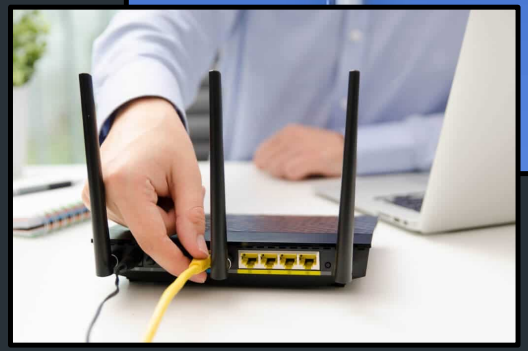


Map of the Internet, 2021 (via BGP)
OPTE project

Challenge: Heterogeneity



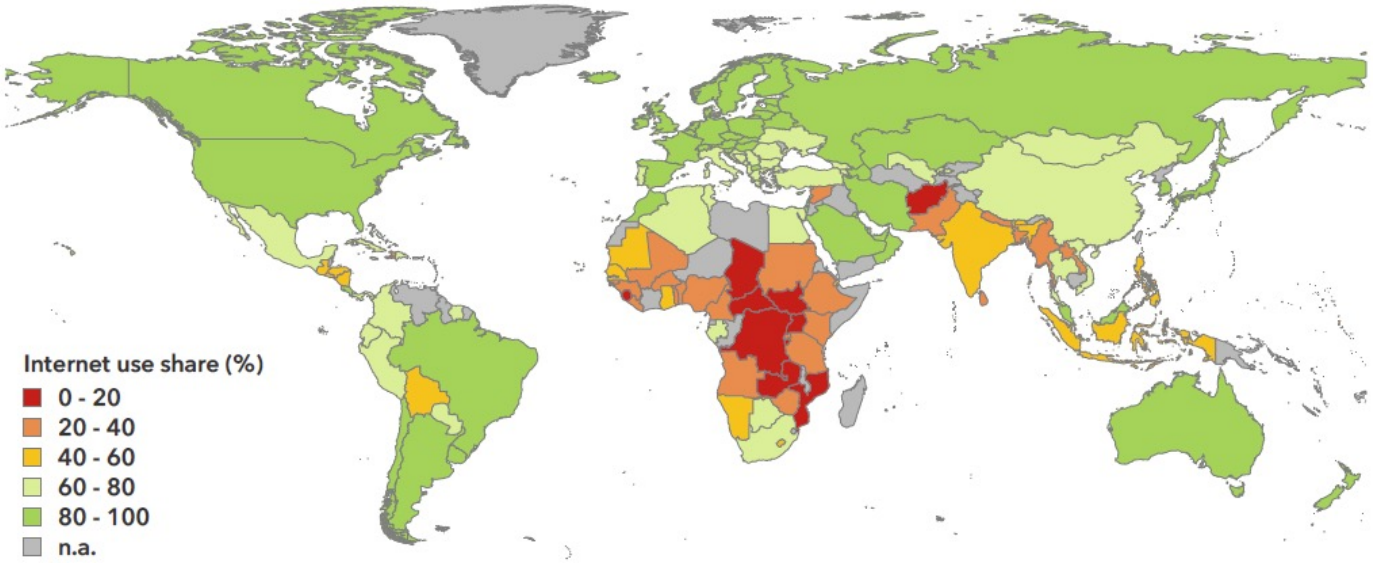




Challenges: Access

Figure 2.5: The global digital divide

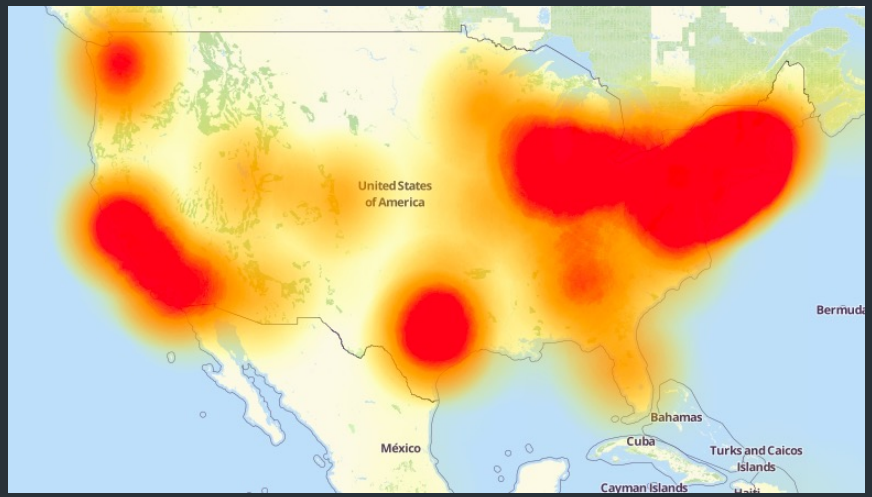
Percentage of the population using the Internet, 2020



Challenge: Security

Mirai Botnet (2016)

- Vulnerable DVRs, Home Routers, Cameras disrupted Dyn
  - DNS provider for Twitter, Netflix, Reddit, others
- Largest denial-of-service (DDoS) attack at the time, over 1TBit/s



Challenges: Politics and Oversight

DDoS attack that disrupted internet was largest of its kind in history, experts say

RYAN SINGEL SECURITY FEB 25, 2008 10:37 AM

Pakistan's Accidental YouTube Re-Routing Exposes Trust Flaw in Net

Explainer

Facebook outage: what went wrong and why did it take so long to fix after social platform went down?

TECHNOLOGY

How Was Egypt's Internet Access Shut Off?

DYLAN TWENEY BUSINESS FEB 3, 2011 9:58 AM

No Easy Fixes as Internet Runs Out of Addresses

Editors given c provid

MARVIN AMMORI OPINION JAN 18, 2014 7:00 AM

Internet Freedom Day: This Year We Go to War for Net Neutrality

How Russia Took Over Ukraine's Internet in Occupied Territories

By Adam Satariano and Graphics by Scott Reinhard
Aug. 9, 2022

Why should you take this course?

- Networks have a huge impact on computing on society

- Continuously changing and evolving
  - Incredible complexity
  - *Facts* you learn will be out of date quickly
  - Instead: learn how to navigate the underlying *principles and abstractions*

Networks are cool…
and you will learn to program them!

How will we do this?

Build networks from the ground up

- How to link 2 things, then ~10 things
- Then, how to link all the things (the Internet)
- Transport: how to send *messages* between *programs*
- Applications: how key protocols/apps make our modern Internet

A few cross-cutting issues:

=> Security, access, social impact, key abstractions…

Logistics

How will we do this?

- 4 Programming Projects (65%)
- 4-5 Written homeworks (35%)
- No exams!

Mechanics: Resources

Lecture slides/notes: authoritative content

Tools

- Course website (notes, handouts, guides): https://brown-csci1680.github.io
- Discussions: **EdStem**
- You are responsible for checking Ed for announcements/updates

Complete HW0 (on website) ASAP so we can add you to resources!

Prerequisites

Prereqs: CS330/CS1330, CS300/CS1310 (or equivalent)

- You should have worked with systems programming and basic OS concepts before
  - Threads, processes, Kernel vs. Userspace
  - Bits and bytes, memory management, synchronization, …

- Graduate students: our systems courses are intensive, make sure your background is equivalent

TODAY 24

If you aren't sure if the course is right for you, please talk to us!

Lectures

- T/Th 9-10:20am, CIT 368
- Recorded and streamed live on zoom
- Lots of live demos/coding, time for discussion, etc.

I encourage you to attend synchronously,
but I won't think less of you if you don't

Mechanics: Homeworks

- Short, conceptual problems based on lecture material
- Experiment with some fundamental networking tools, tutorial-style

<u>How these work</u>

- Shouldn't take a huge amount of time
- 4 or 5 homeworks total (TBA based on enrollment)

Mechanics: Projects

Build fundamental Internet protocols and client/server apps from the ground up

- 4 Programming projects
  - Snowcast: streaming music server
  - IP: build your own networking library
  - TCP: extending your IP
  - Final (shorter, choose-your-own)
- First project is individual, others in groups of 2

Mechanics: Projects

Build fundamental Internet protocols and client/server apps from the ground up

- 4 Programming projects
  - Snowcast: streaming music server
  - IP: build your own networking library
  - TCP: extending your IP
  - Final (shorter, choose-your-own)
- First project is individual, others in groups of 2

You get: lots of freedom to design your own system

Languages

You can use any systems language

- Go (recommended)
- C
- C++
- Rust

- You'll be making a lot of threads, manipulating bits/bytes, …

We recommend Go, even if it's new to you.
We'll do a bunch of live coding in Go in lectures 2-3

How projects work

- This is where you will spend most of your time.

Learn how to design big systems… that happen to use the network

  - No stencil code! You get to design everything
  - We will provide lots of examples and meetings to help

What this means

nick: no stencil code

everyone:

The workload for the projects is non-trivial.

If you meet the prereqs, you have the skills.

But you need to be prepared for the time commitment.

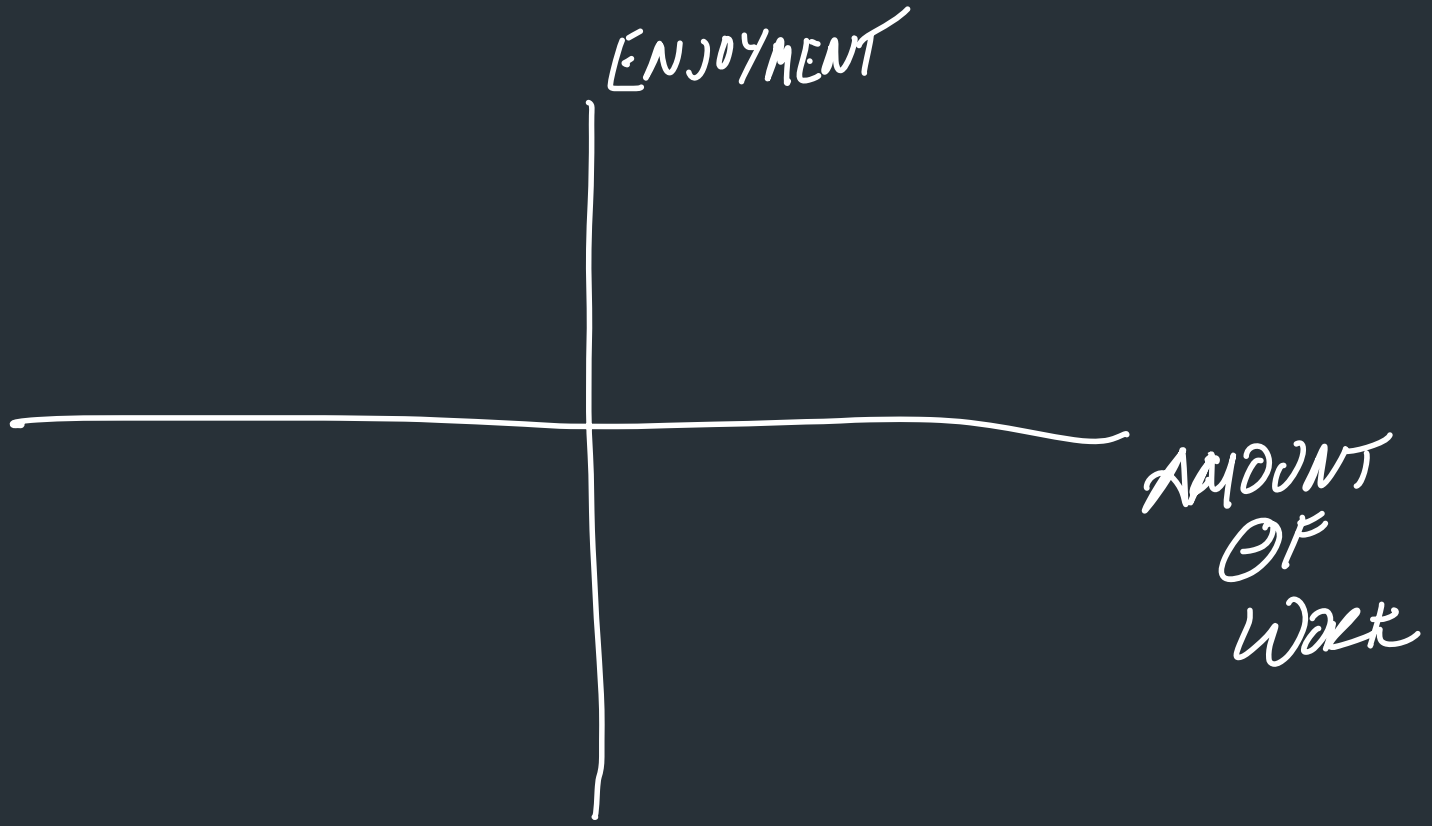
How to succeed? **Start early**, take advantage of the tools and resources we provide (there's a lot!)

How we support you on projects

- Lots of posted code examples
- Live demos during class
- "Warmup" tutorials to get started with mechanics
- Gearups at least once per project
  - When scheduled: Thursdays 5-7pm in CIT368 (+ Zoom, recording)
- Milestone meetings with TA to check in about design
- Interactive grading => more partial credit

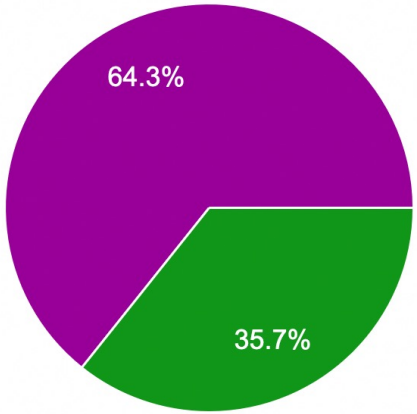
Most of our time is spent here too

How to compare this class to others



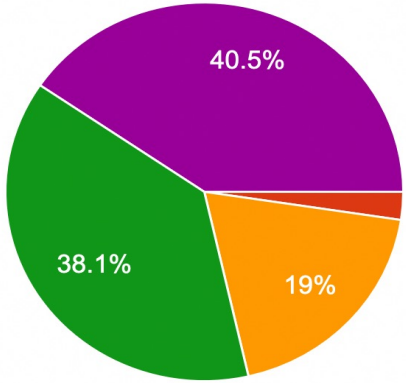
Overall, I found this course a worthwhile experience

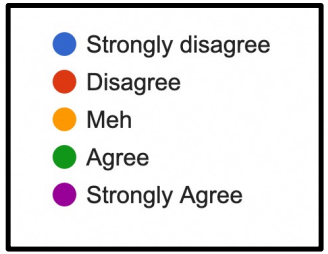
42 responses



Overall, I found this course an enjoyable experience

42 responses





Notes from former students:

- *"If you spend an hour or two hours on IP and TCP every day it is *very* manageable"*
- *"I am really glad that I took this course"*
- *"it's an amazing course and we had a wonderful staff to help us out"*
- *"Just take it, worth it"*
- *"Pace yourself, and don't wait until the last minute"*

Asking for help

- Online help: Ed discussion
- Office hours: collaborative (group, in-person)

Can help with…

- Debugging
- Project planning/design
- Concepts
- We're here for you—don't be afraid to ask!

I don't bite—Instructor's office hours are just like regular hours!

Asking for help

Collaboration: work with your peers!

- Collaboration policy on course website
- I encourage you to collaborate, so long as the code you write down is your own

- Your health is most important
  - If you have concerns, feel free to talk to us

Why should you listen to me?

- My background
  - Received my PhD from Brown in 2021
  - My areas: software security, networking, network security
  - My second year as Lecturer, was a long-time TA before that

THIRD

- No one knows everything about networks, and I am no exception!

Brief history of this class

Fall 2019: ~35 students

Spring 2022: Nick joins as instructor (Capped at 40)
(+ demos, examples, tutorials, dev environment)

Fall 2022: Second new offering (Capped at 40)
(+ warmups/tutorials)

Fall 2023: Scaling up (127 students!)
(+ Gearups)

Fall 2024: You are here
(+ Refinements, new lecture content, HW tutorials)

We're listening!

Our course is now more stable, but we're still evolving and learning how to scale.

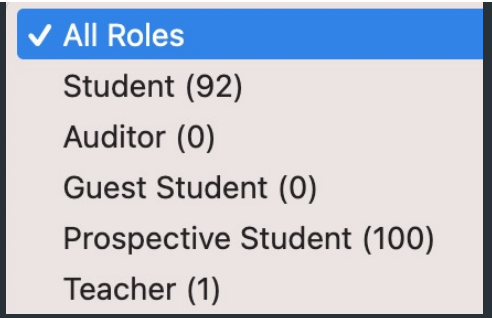
We may adjust course content/policies over time, paying equal attention to:

- Making sure we provide support for everyone
- Managing TA workloads

We value your feedback! (Ed, email, anonymous form, …)

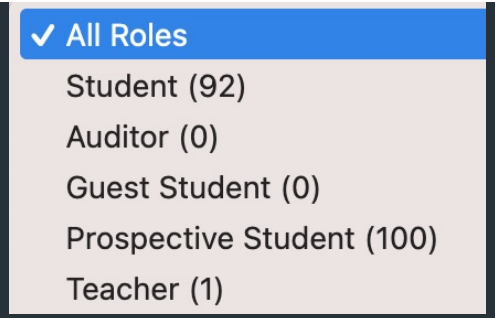
Registration and Overrides

Registration/Overrides



- Our enrollment is still fluctuating as overrides in other courses are being granted
- We can support ~160 students. A lot of you will get seats.
- On Monday (9/9), we'll send an email on whether we can offer you a seat, based on latest data (+ more updates as seats open up)

Registration/Overrides



- Our enrollment is still fluctuating as overrides in other courses are being granted
- We can support ~160 students. A lot of you will get seats.
- On Monday (9/9), we'll send an email on whether we can offer you a seat, based on latest data (+ more updates as seats open up)

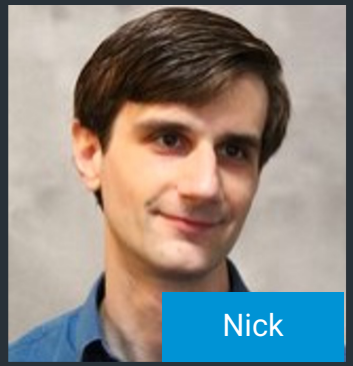
To receive an override, you must complete our assignments during shopping period: HW0, Project 0 (Setup), and the milestone for Project 1 (Snowcast)

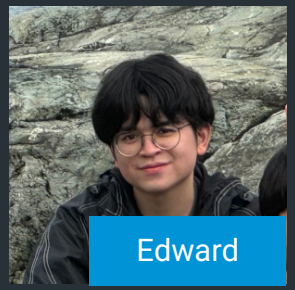
What you should do now

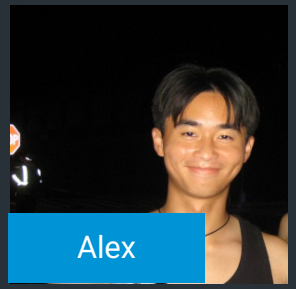
- Decide if you want to take this course
  - If you don't, no hard feelings!
  - Please come chat with me if you're unsure
- If you have not done so already: fill out our registration request form
- Start on assignments; look for email on Monday (to prioritize if you want to do the work)
- More generally: watch this doc for updates on the process as a whole

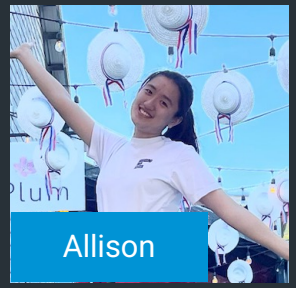
I know this is frustrating. Thanks for your patience and understanding!

Cast

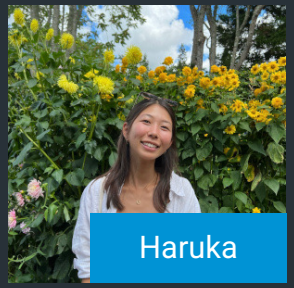

Nick

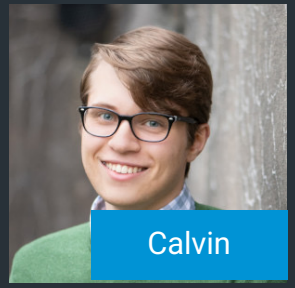

Edward

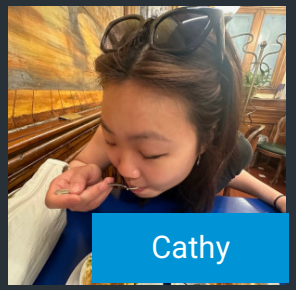

Alex


Allison

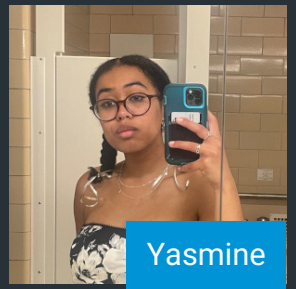

Brandon

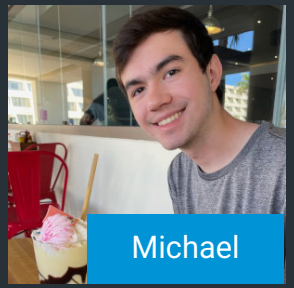

Haruka


Calvin


Cathy


Jason


Yasmine


Michael

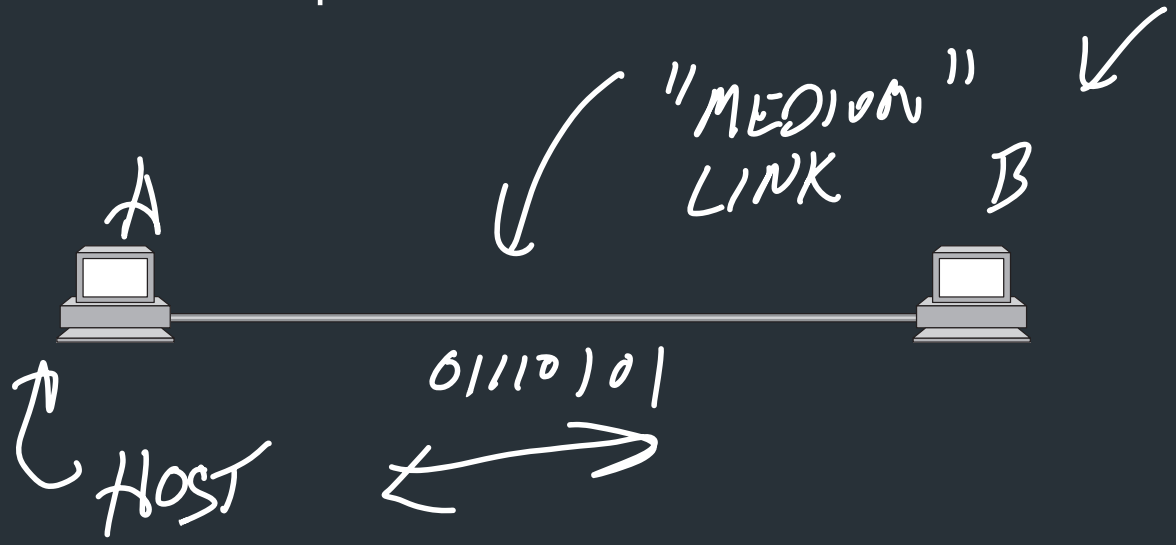

Yi Hao

Stretch

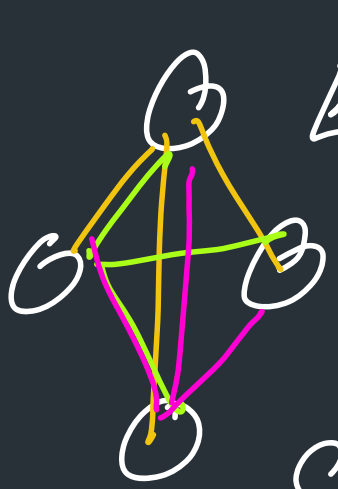
(and I won't look if you are shopping and want to flee)

Building Blocks

- Nodes: Computers (hosts), dedicated routers, …
- Links: Coax, twisted pair, fiber, radio, …



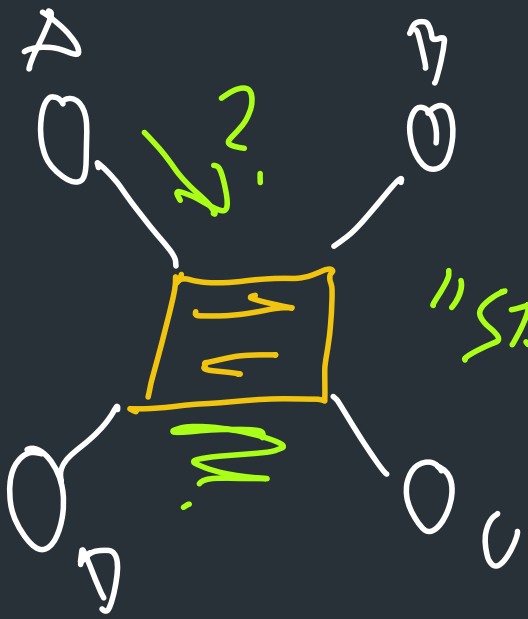
How to connect more nodes?



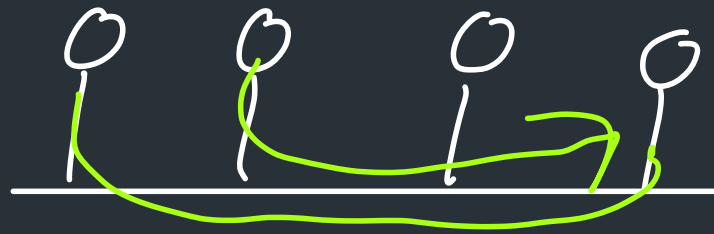
MULTIPLE LINKS

⇒ MULTIPLEXING

WANT: SHARE THE MEDIUM

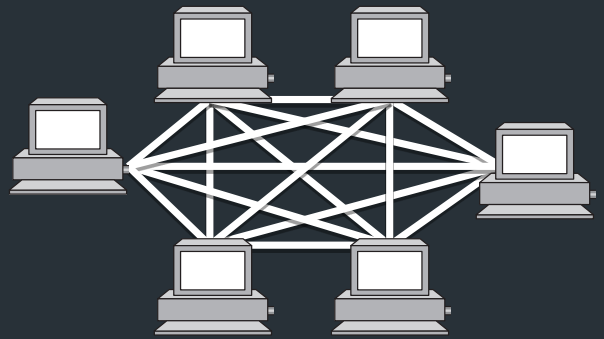


"STAR"



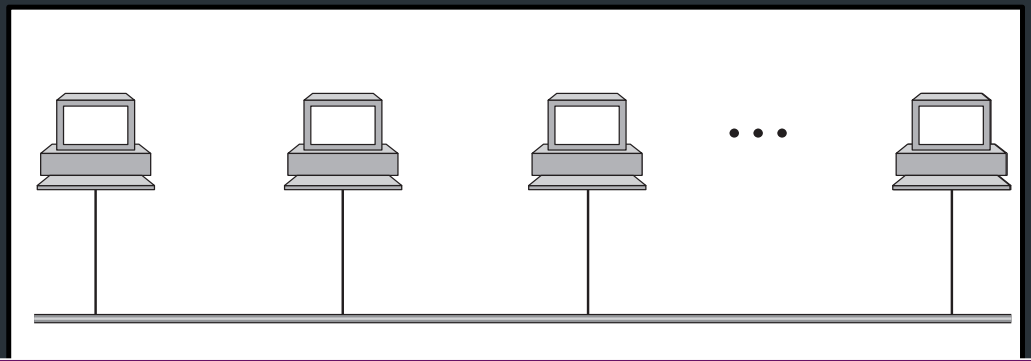
SHARED LINE/ MEDIUM

How to connect more nodes?



Multiple wires

Shared medium



Need a mechanism to share network resources

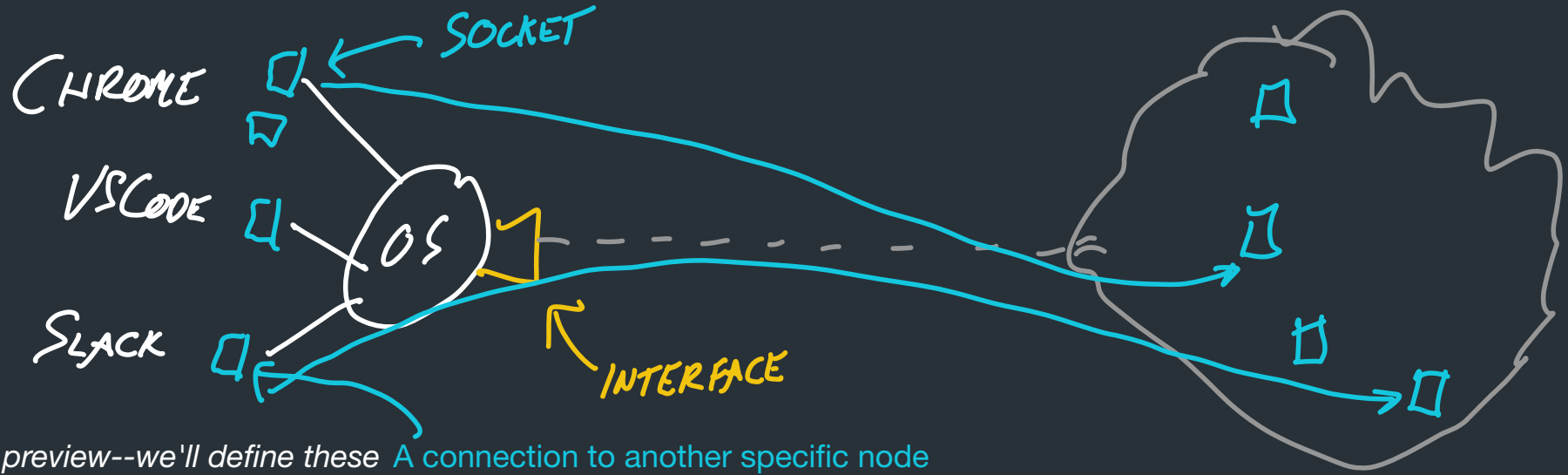
Multiplexing

- Mechanisms for simultaneous communication on the same channel
  - (or at least *nearly* simultaneous)

- Lots of different methods, depending on the medium and abstraction

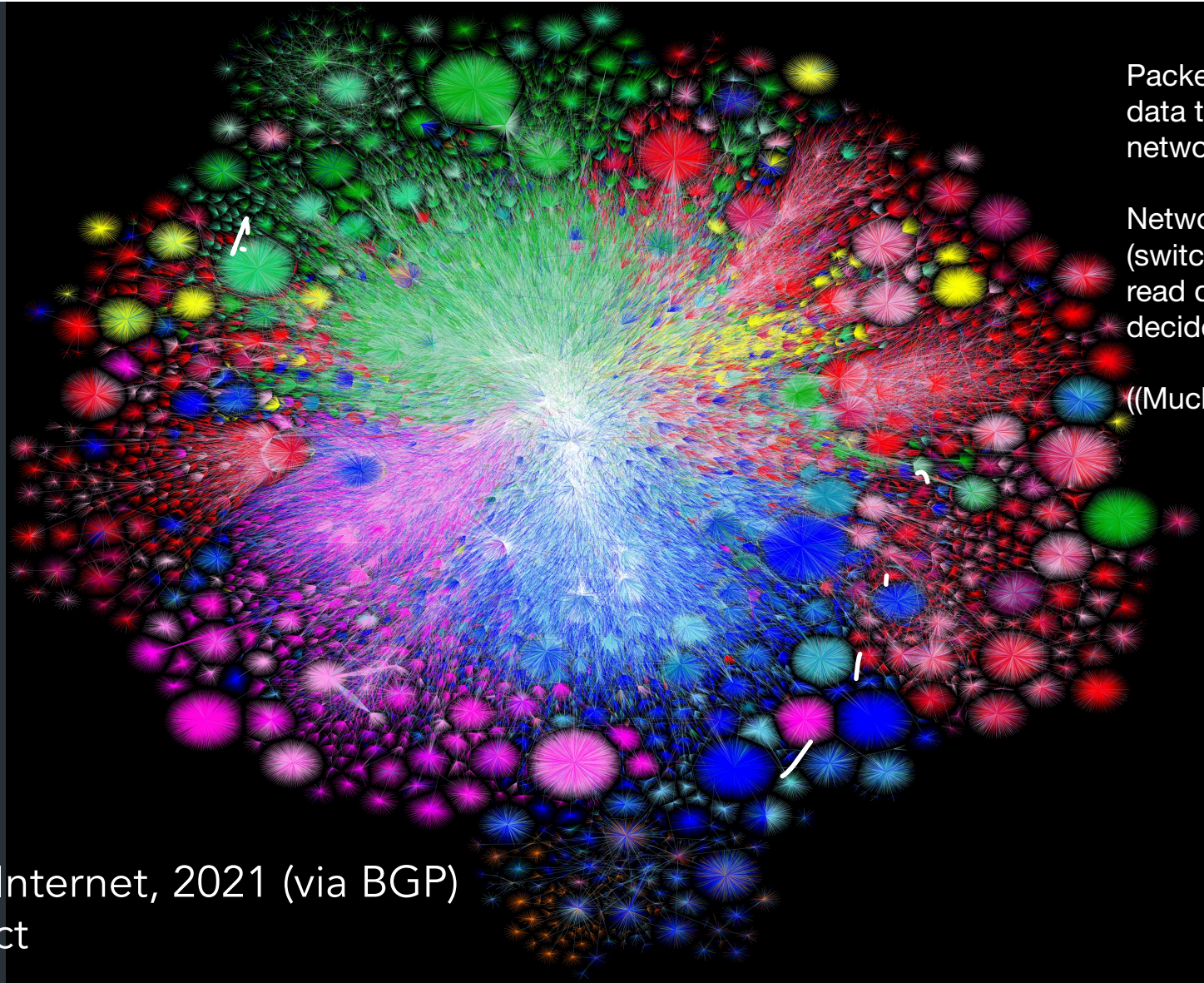
How do you connect *apps*?

- Your computer runs multiple applications
- How to make them share the same resources?



(This is just a preview--we'll define these terms more concretely later!)

A connection to another specific node (app server, other host, etc.)



Packets: small unit of data that moves across network

Network devices (switches, routers, etc) read data in packet to decide where to send it

((Much) more on this later!)

Color Chart

- North America (ARIN)
- Europe (RIPE)
- Asia Pacific (APNIC)
- Latin America (LANIC)
- Africa (AFRINIC)
- Backbone
- US Military

Map of the Internet, 2021 (via BGP)
OPTE project

Demo: first steps in wireshark

Take a look at the packets you see: there's a lot of stuff that won't make sense yet (and that's fine!), but is there anything you recognize?

Some things we can see now, without getting into the details:
 - More packets as more people connect to the server
 - Some strings we can recognize: we can see names and files that look like they belong on the test site (eg. images of the staff members)
 - Some protocol names may seem familiar: for example, some packets are tagged "HTTP", like the link: http://...

REMEMBER: this is just a preview! It's completely normal that most of what you see won't make sense right now. We're going to spend the whole semester breaking down the different components here! Do not fear, we'll get there!