UNIVERSITY OF
WATERLOO

CS 856: Programmable Networks

Mina Tahmasbi Arashloo

Winter 2024

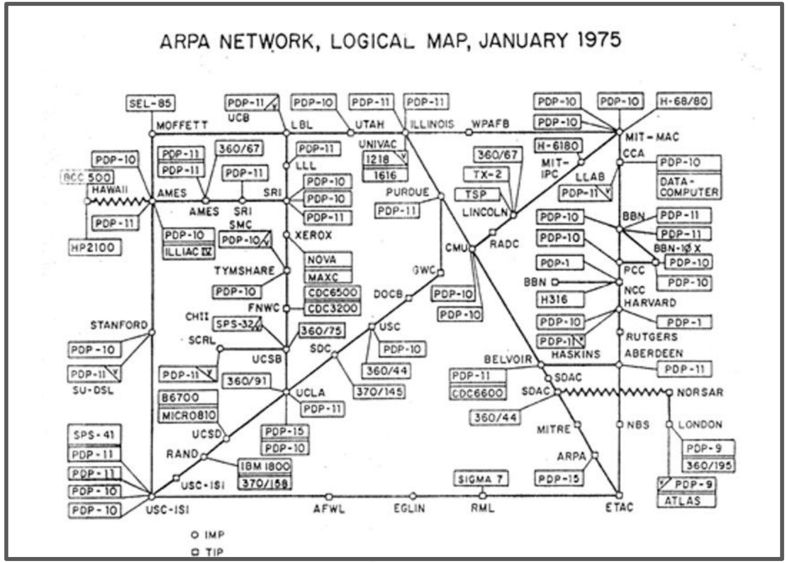
Networks when they started (1970s)

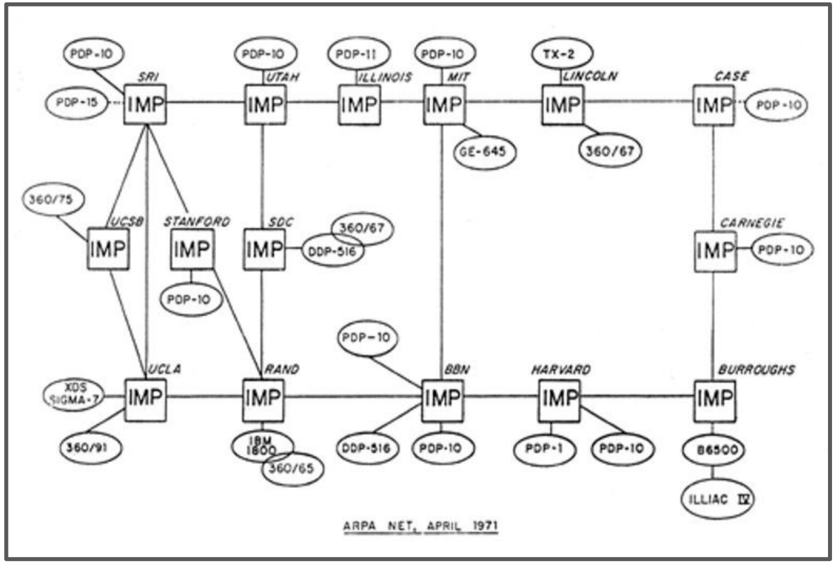
- Small and simple

Networks when they started (1970s)

- Small and simple

Tens of nodes


ARPA NETWORK, LOGICAL MAP, JANUARY 1975


ARPA NET, APRIL 1971

* photo credit: https://www.computerhistory.org/internethistory/1970s/

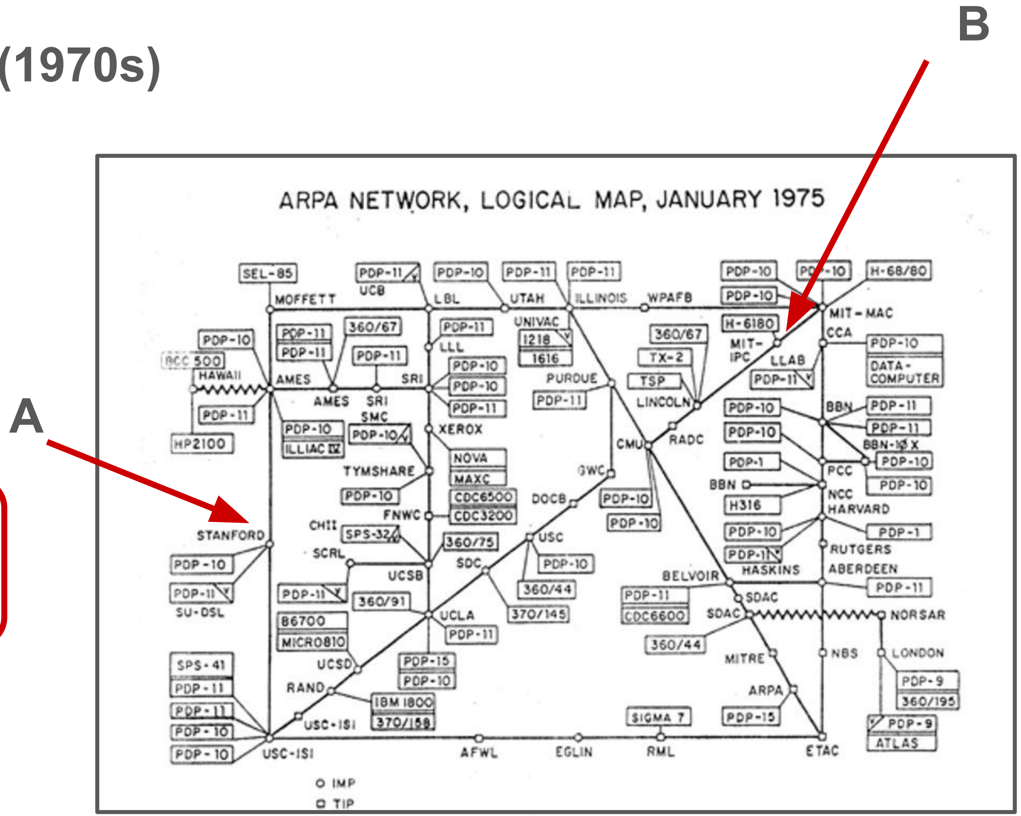
Networks when they started (1970s)

- Small and simple
- A scientific experiment

Networks when they started (1970s)

- Small and simple
- A scientific experiment
- Few simple requirements

Get data from A to B
(preferably without losing it 🙂)



ARPA NETWORK, LOGICAL MAP, JANUARY 1975

Networks when they started (1970s)

- Small and simple
- A scientific experiment
- Few simple requirements

Networks today (2020s)

Networks when they started (1970s)

- Small and simple
- A scientific experiment
- Few simple requirements

Networks today (2020s)

- Large and complex

Thousands, even millions of nodes.

Networks when they started (1970s)

- Small and simple
- A scientific experiment
- Few simple requirements

Networks today (2020s)

- Large and complex
- Critical infrastructure/ Public utility

Networks when they started (1970s)

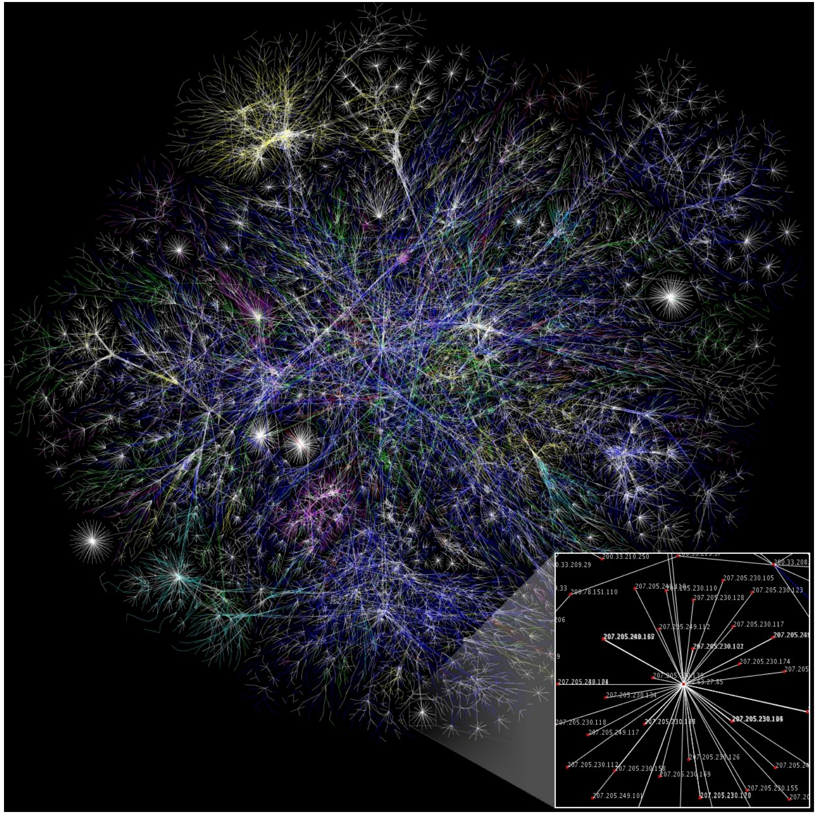
- Small and simple
- A scientific experiment
- Few simple requirements

Networks today (2020s)

- Large and complex
- Critical infrastructure/ Public utility
- Many complex requirements

- Get data from A to B
- Ensure isolation
- Maintain quality of service
- High throughput
- Low latency
- Low jitter
- …

Networks today



Canada
United States of America
México
Colombia
Perú
Brasil
Bolivia
Paraguay
Россия
Қазақстан
Монгол Улс
中华人民共和国
India
Pakistan
Nepal
Bangladesh
Mali
Niger
Tchad
South Sudan
Angola
Malawi
Zimbabwe
Tanzania
Australia
New Zealand
Indonesia
Timor-Leste
Nauru
Cook Islands
Pitcairn Islands
Saint Helena

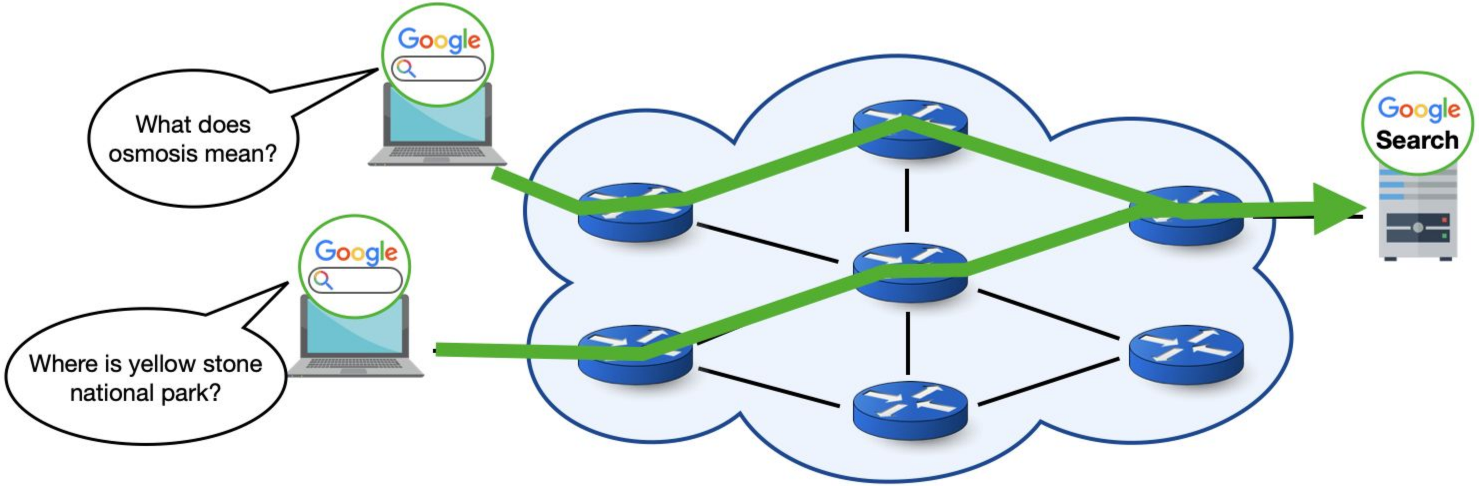
map.info/ Greg's Cable Ma

North Dakota
South Dakota
Minnesota
Wisconsin
Wyoming
Nebraska
Nevada
Utah
Colorado
Kansas
Missouri
Kentucky
West Virginia
United States
Las Vegas
Albuquerque
Arizona
New Mexico
Oklahoma
Arkansas
Mississippi
Alabama
Texas
New Orleans
Baja California
Sonora
Chihuahua
Coahuila
Monterrey
Mexico
Toronto
Ottawa
Montreal
Quebec City
Gulf of Mexico
The Bahamas
Havana

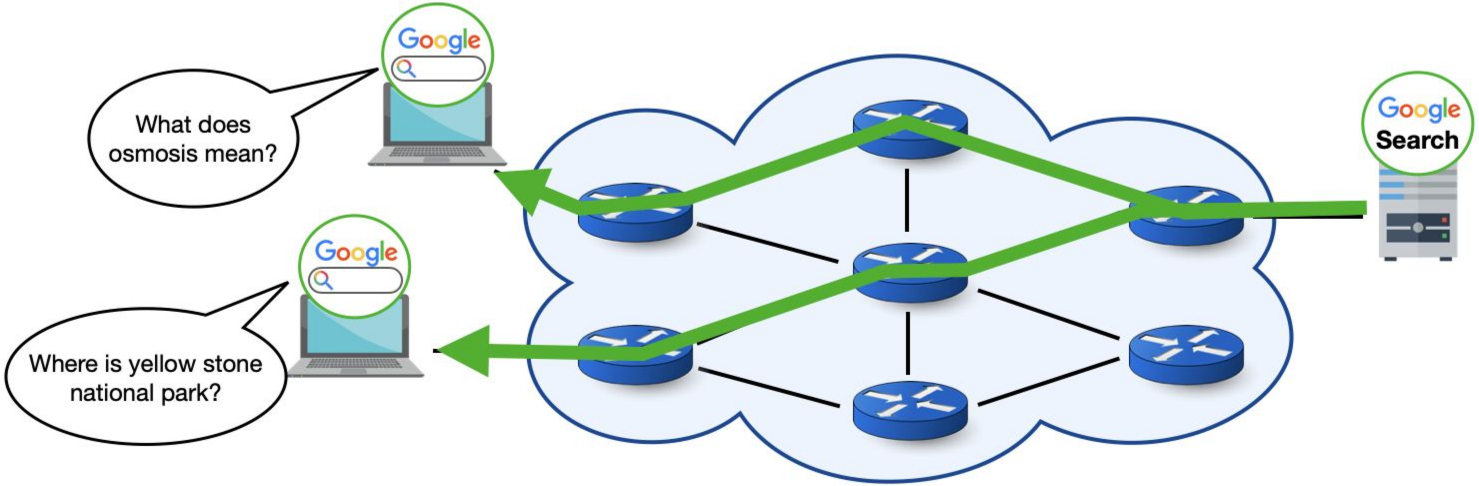
/internetatlas.org
sity of Wisconsin - Madison

How does this affect network design, operation, and management?

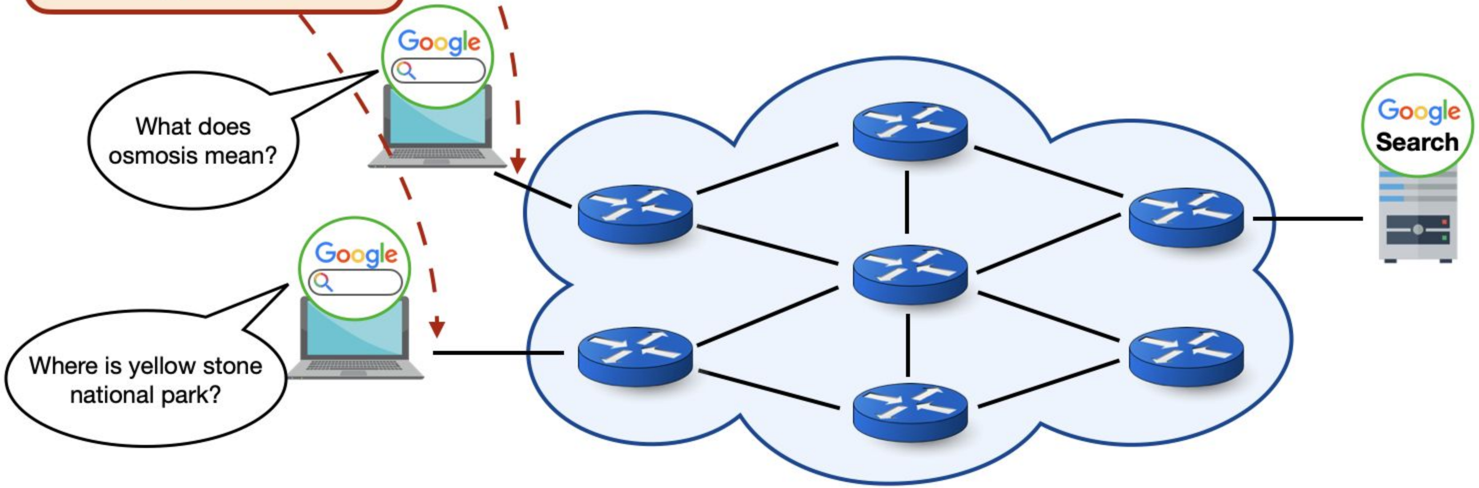
Example Network



Example Network



Example Algorithms and Protocols



How fast should data enter the network?

e.g., congestion control algorithms

x

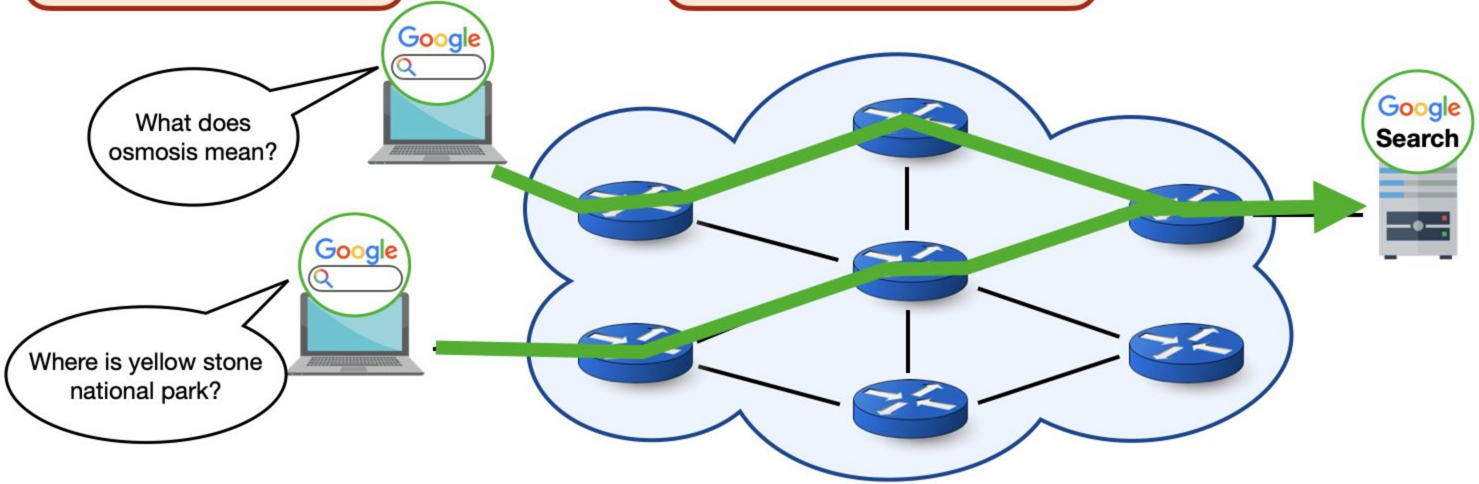
Example Algorithms and Protocols

How fast should data enter the network?

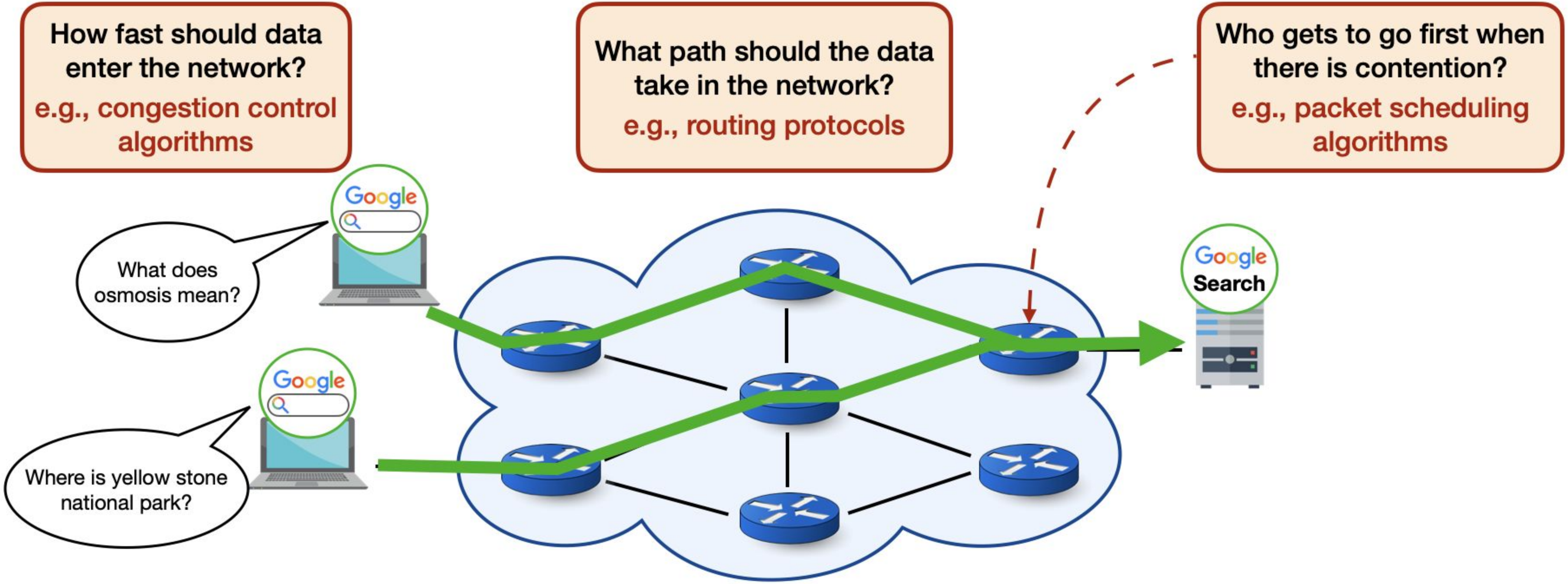
e.g., congestion control algorithms

What path should the data take in the network?

e.g., routing protocols



Example Algorithms and Protocols



How fast should data enter the network?

e.g., congestion control algorithms

What path should the data take in the network?

e.g., routing protocols

Who gets to go first when there is contention?

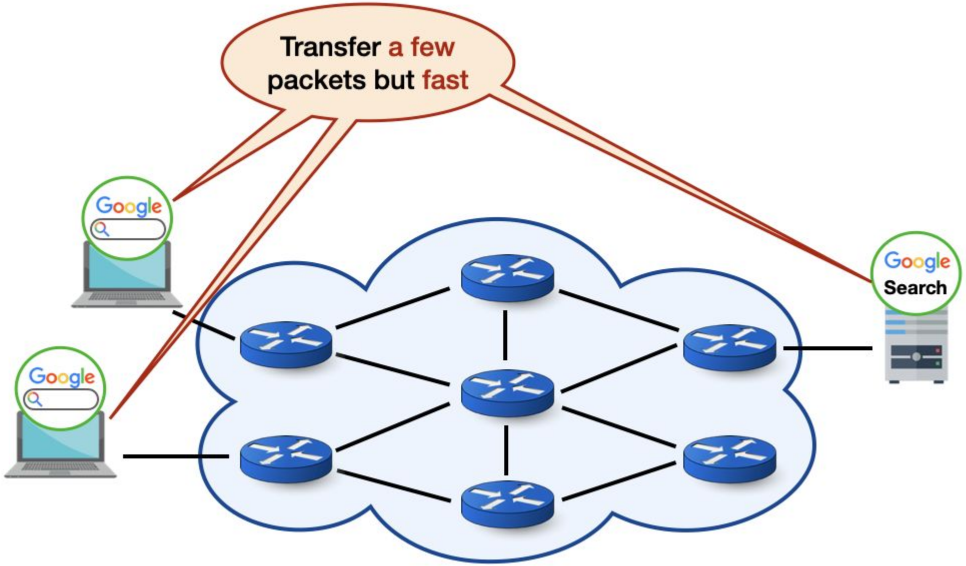
e.g., packet scheduling algorithms

Small Network, One Application, A Few Endpoints

How fast to transmit?

What path to pick?

Who goes first?



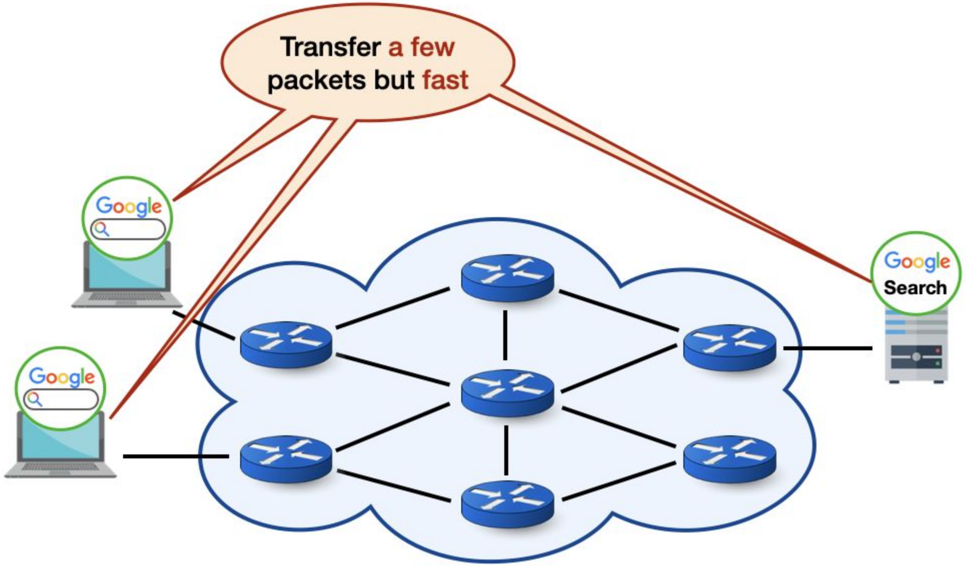
Small Network, One Application, A Few Endpoints

How fast to transmit?

Start fast and back off on loss.

What path to pick?

Who goes first?



Small Network, One Application, A Few Endpoints

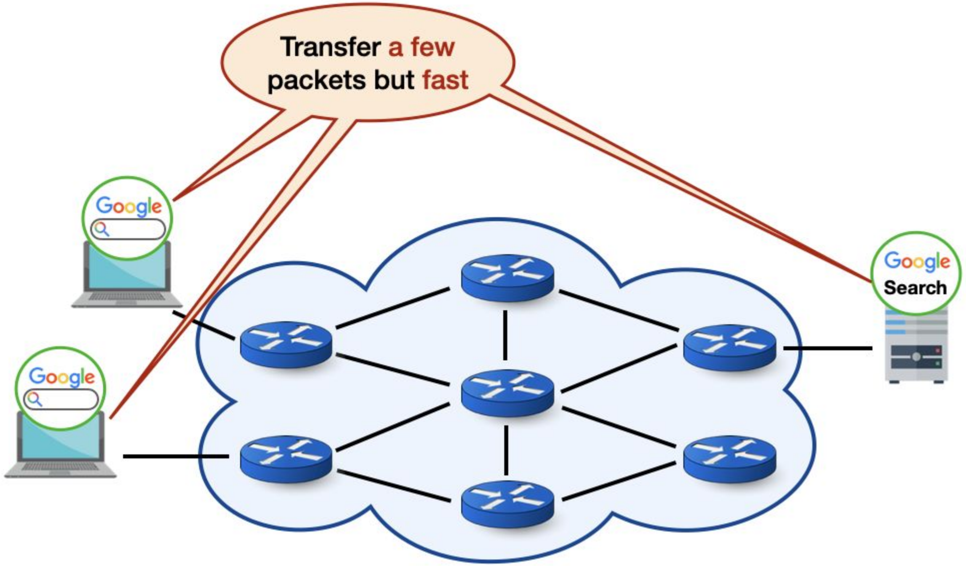
How fast to transmit?

Start fast and back off on loss.

What path to pick?

Pick one of the shortest path at random.

Who goes first?



Small Network, One Application, A Few Endpoints

How fast to transmit?

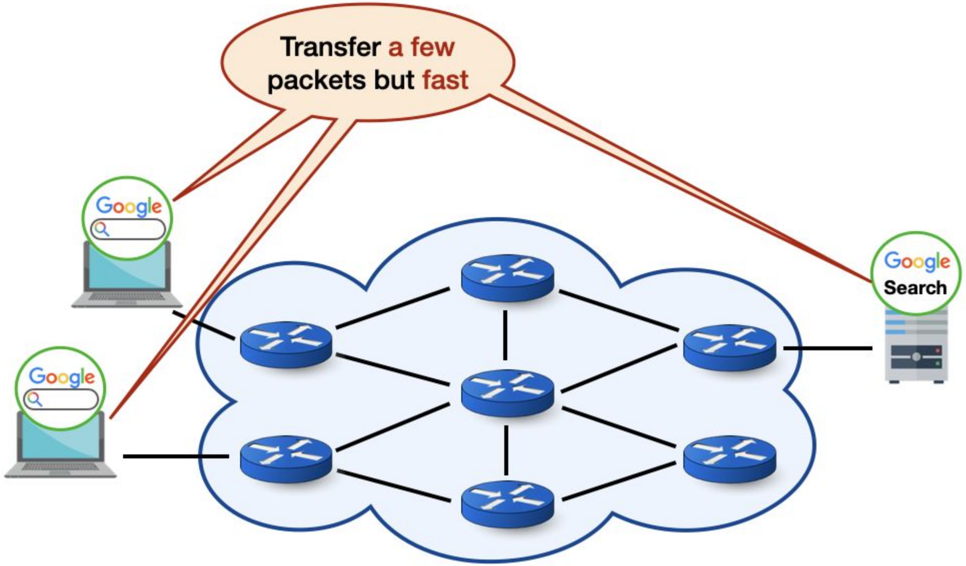
Start fast and back off on loss.

What path to pick?

Pick one of the shortest path at random.

Who goes first?

First come, first serve.



Small Network, More Applications, A Few Endpoints

How fast to transmit?

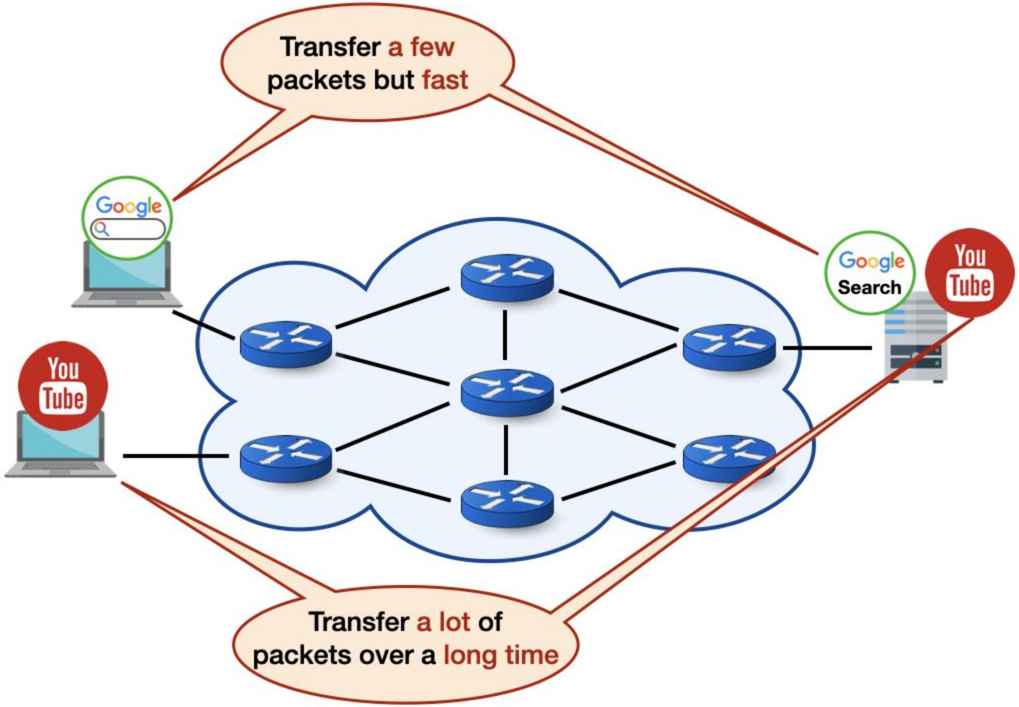
Start fast and back off on loss.

What path to pick?

Pick one of the shortest path at random.

Who goes first?

First come, first serve.


Transfer a few packets but fast
Google
Google Search
You Tube
You Tube
Transfer a lot of packets over a long time

Small Network, More Applications, A Few Endpoints

How fast to transmit?

Start fast and back off on loss.

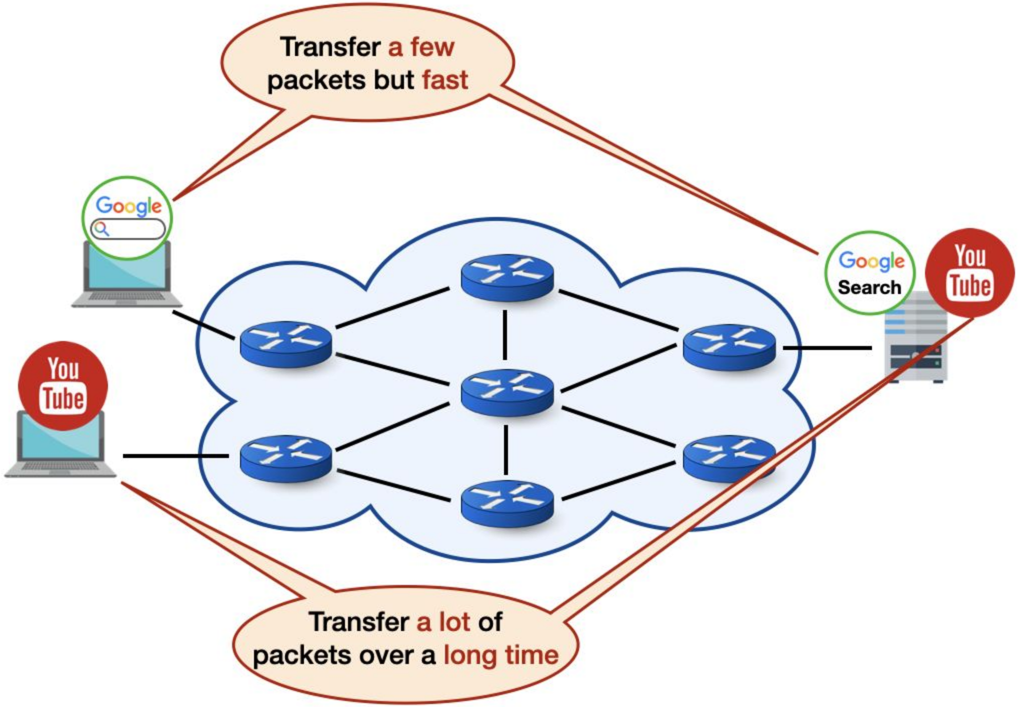
What path to pick?

~~Pick one of the shortest path at random.~~

Pick the least loaded path so search traffic avoids video traffic.

Who goes first?

First come, first serve.



Transfer a few packets but fast
Google
Google Search
You Tube
You Tube
Transfer a lot of packets over a long time

Small Network, More Applications, A Few Endpoints

How fast to transmit?

Start fast and back off on loss.

What path to pick?

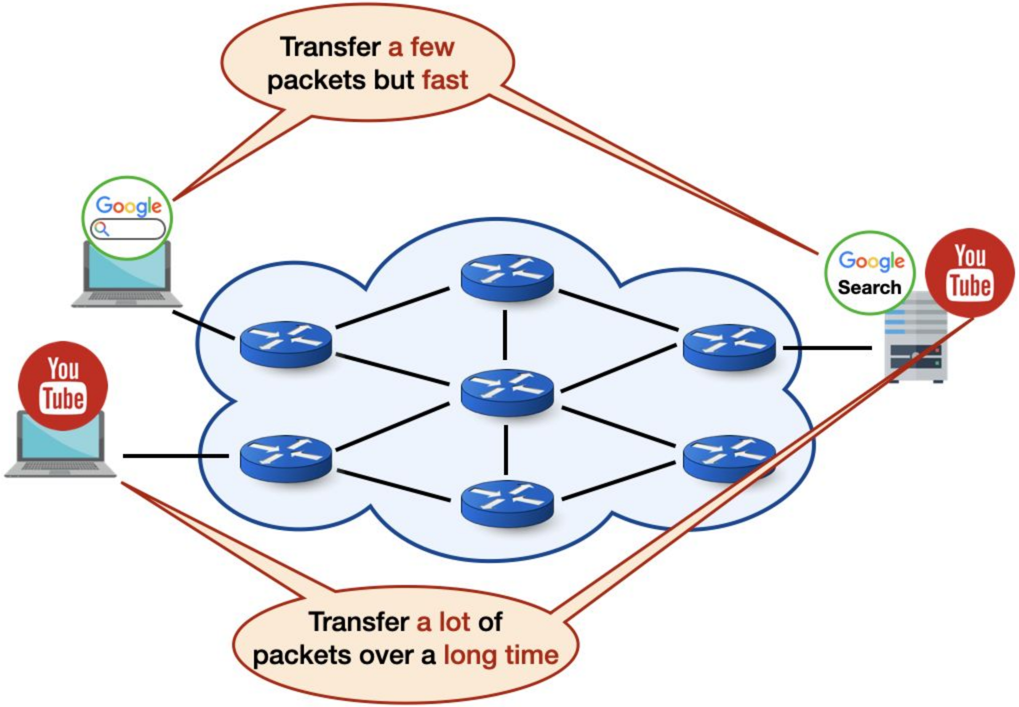
~~Pick one of the shortest path at random.~~

Pick the least loaded path so search traffic avoids video traffic.

Who goes first?

~~First come, first serve.~~

Prioritize search over video.


Transfer a few packets but fast
Google
Google Search
You Tube
You Tube
Transfer a lot of packets over a long time

Large Network, More Applications, Many Endpoints

How fast to transmit?

Start fast and back off on loss.

What path to pick?

~~Pick one of the shortest path at random.~~

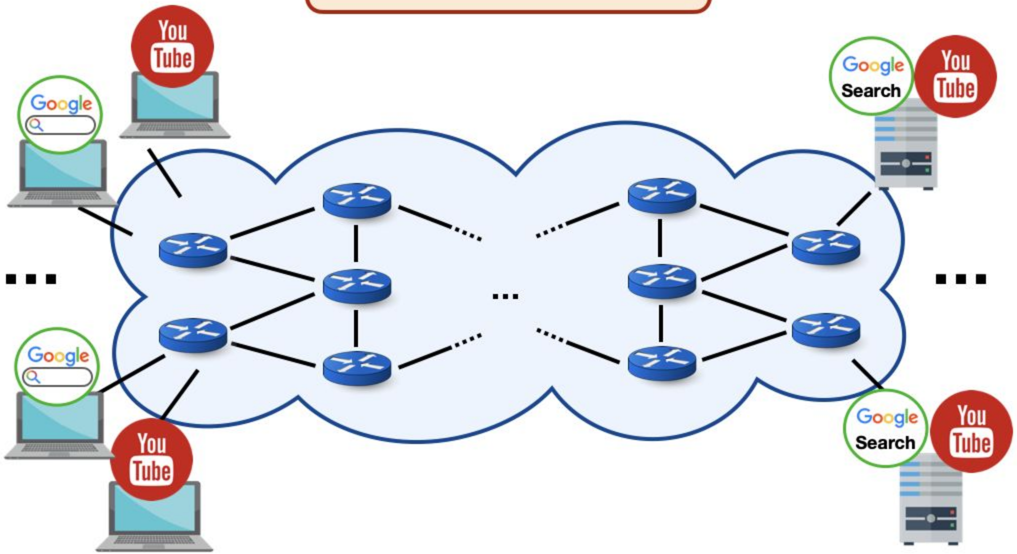
Pick the least loaded path so search traffic avoids video traffic.

Who goes first?

~~First come, first serve.~~

Prioritize search over video.

1000x more flows
100x more switches



Large Network, More Applications, Many Endpoints

How fast to transmit?

~~Start fast and back off on loss~~

Search: start fast and back off on loss

Video: start slow and increase if no loss

What path to pick?

~~Pick one of the shortest path at random.~~

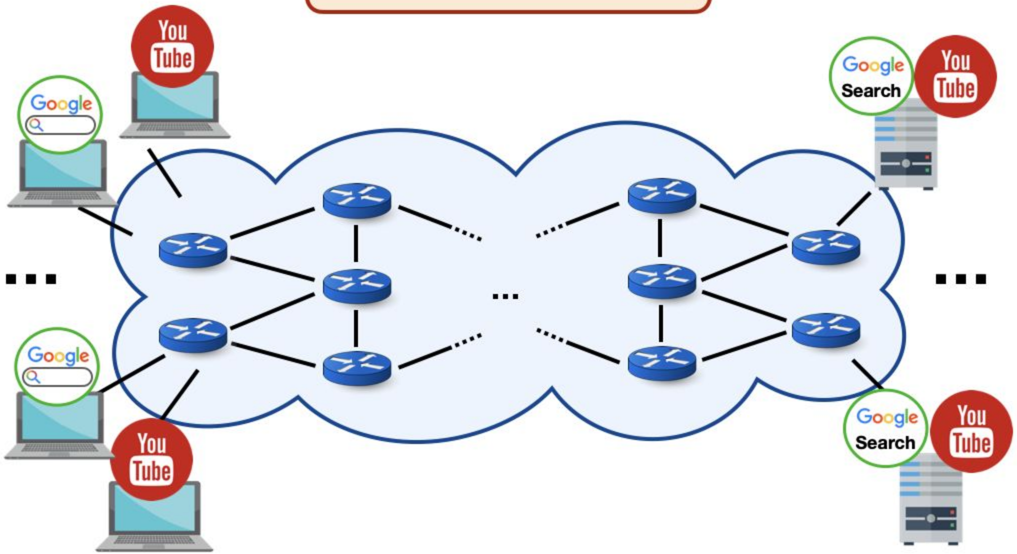
Pick the least loaded path so search traffic avoids video traffic.

Who goes first?

~~First come, first serve.~~

Prioritize search over video.

1000x more flows
100x more switches



Large Network, More Applications, Many Endpoints

How fast to transmit?

~~Start fast and back off on loss~~

Search: start fast and back off on loss

Video: start slow and increase if no loss

What path to pick?

~~Pick one of the shortest path at random.~~

Pick the least loaded path so search traffic avoids video traffic.

Who goes first?

~~First come, first serve.~~

Prioritize search over video.

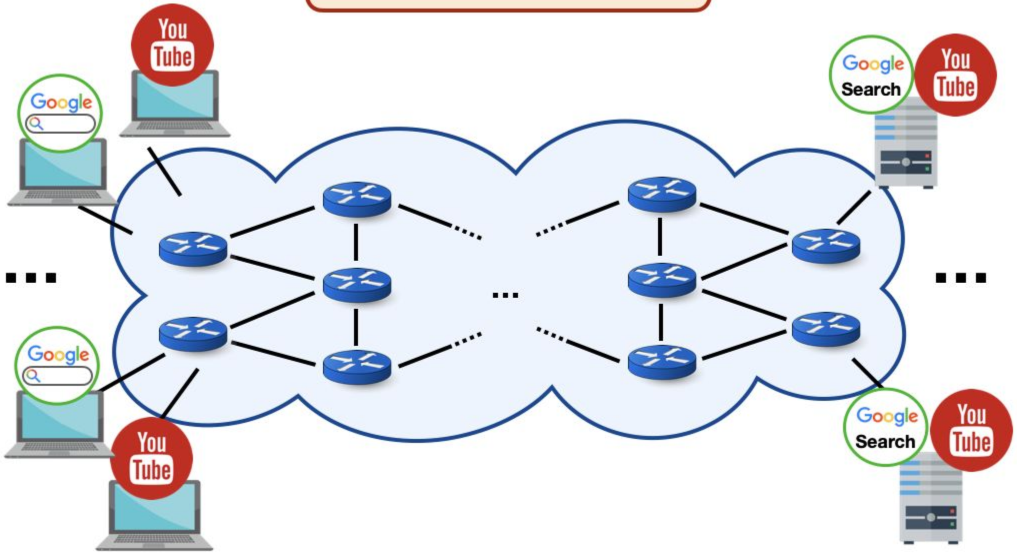
Where do I implement them?

On the edge switches and two of the cores.

How much time do I have?

1μs per packet.

1000x more flows
100x more switches

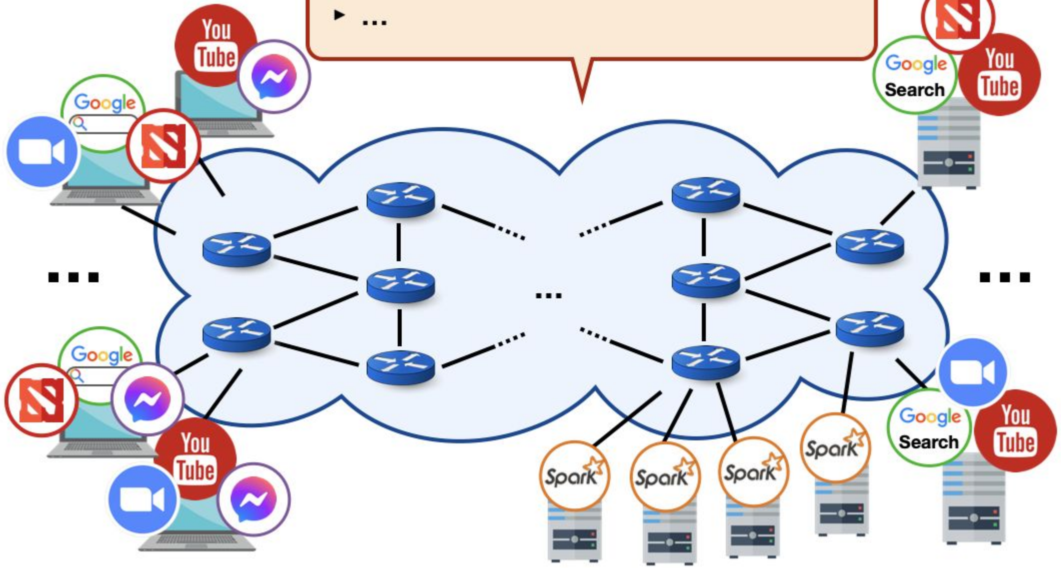


Diverse Applications

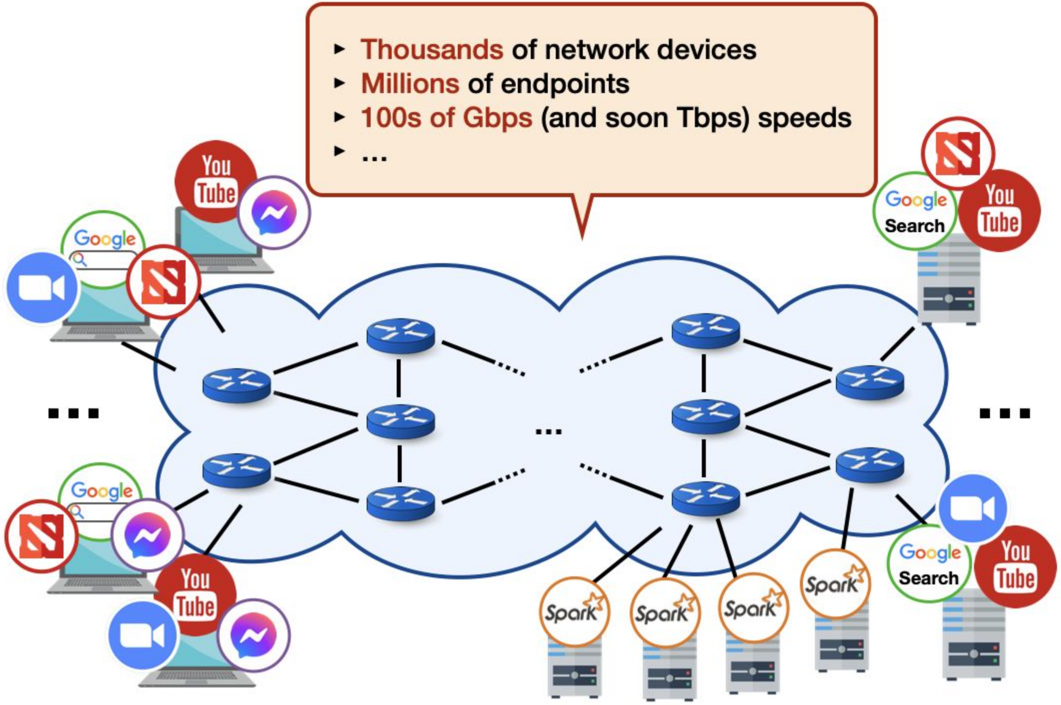
Large Scale and High Speed

Constantly Evolving Algorithms & Protocols

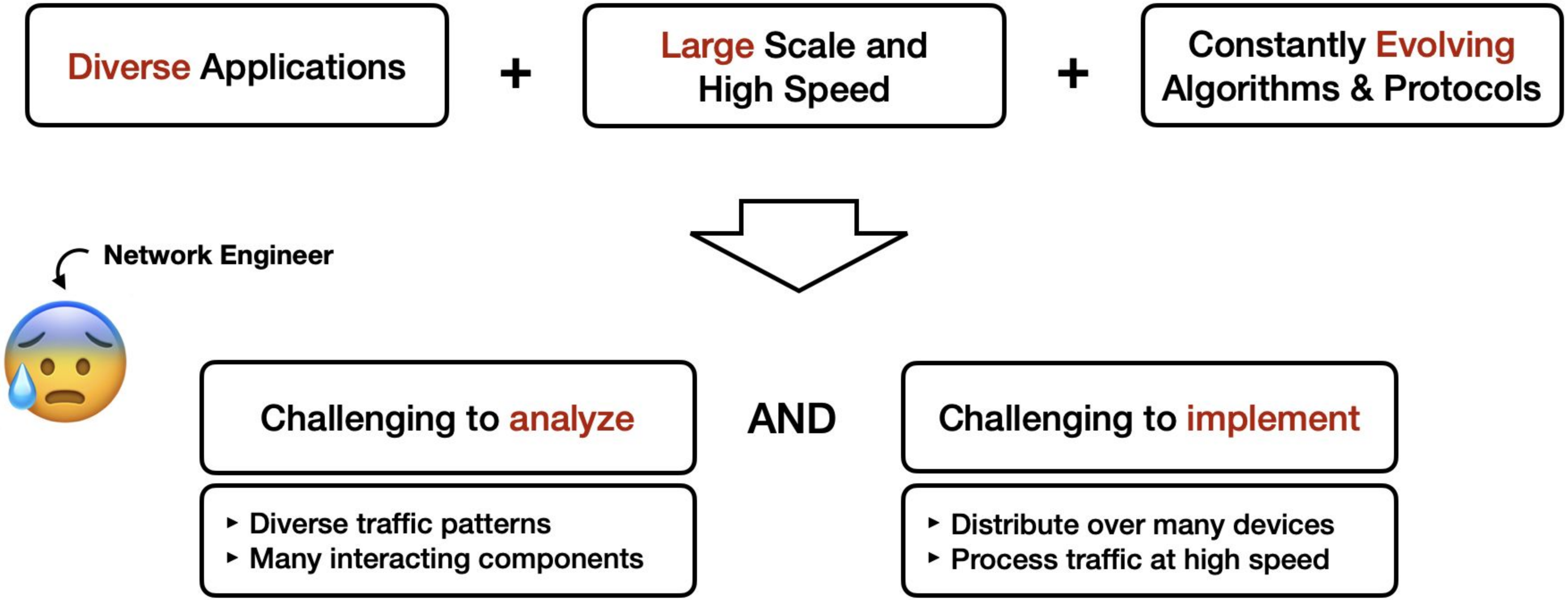
- Thousands of network devices
- Millions of endpoints
- 100s of Gbps (and soon Tbps) speeds
- ...



Diverse Applications + **Large Scale and High Speed** + **Constantly Evolving Algorithms & Protocols**



- Thousands of network devices
- Millions of endpoints
- 100s of Gbps (and soon Tbps) speeds
- …



Diverse Applications + Large Scale and High Speed + Constantly Evolving Algorithms & Protocols

Network Engineer 😰

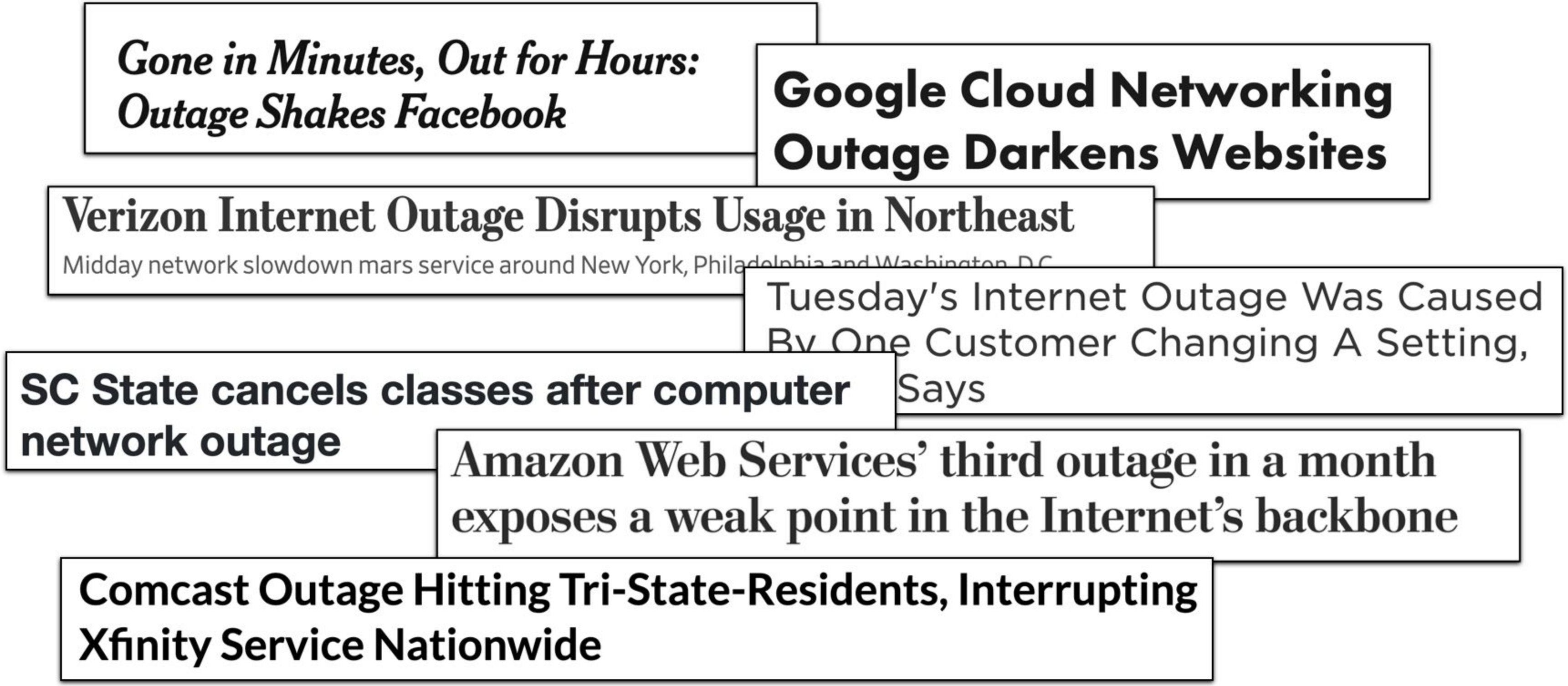
Challenging to analyze

- Diverse traffic patterns
- Many interacting components

AND

Challenging to implement

- Distribute over many devices
- Process traffic at high speed



Gone in Minutes, Out for Hours: Outage Shakes Facebook

Google Cloud Networking Outage Darkens Websites

Verizon Internet Outage Disrupts Usage in Northeast

Midday network slowdown mars service around New York, Philadelphia and Washington, D.C.

Tuesday's Internet Outage Was Caused By One Customer Changing A Setting, Says

SC State cancels classes after computer network outage

Amazon Web Services' third outage in a month exposes a weak point in the Internet's backbone

Comcast Outage Hitting Tri-State-Residents, Interrupting Xfinity Service Nationwide

How can we make it better?

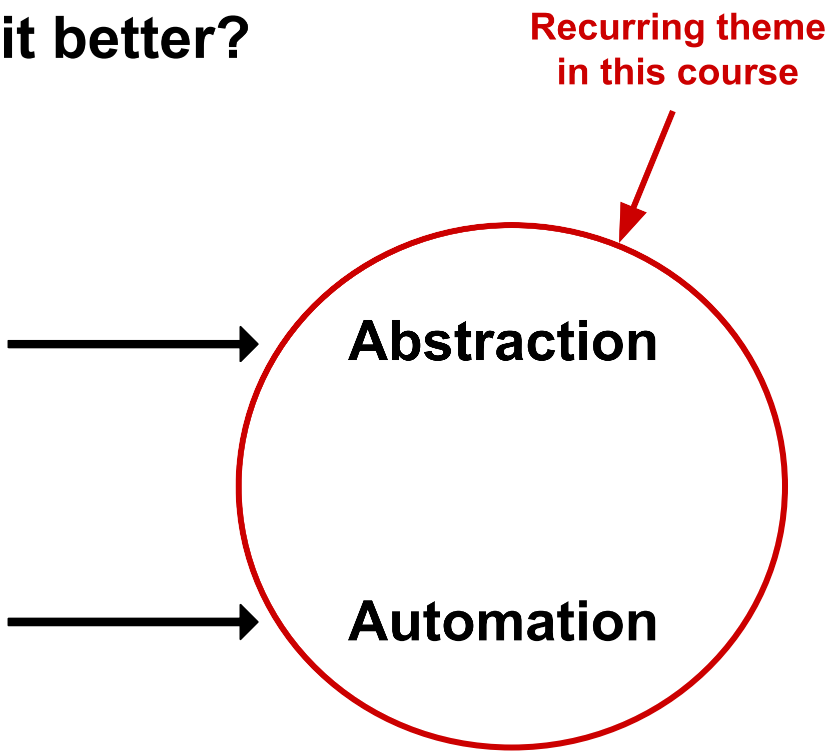
Separate *what* you want the network to do from *how* it is implemented → **Abstraction**

Don't implement in *manually* 🙂 → **Automation**

How can we make it better?

Separate *what* you want the network to do from *how* it is implemented

Don't implement in *manually* 🙂



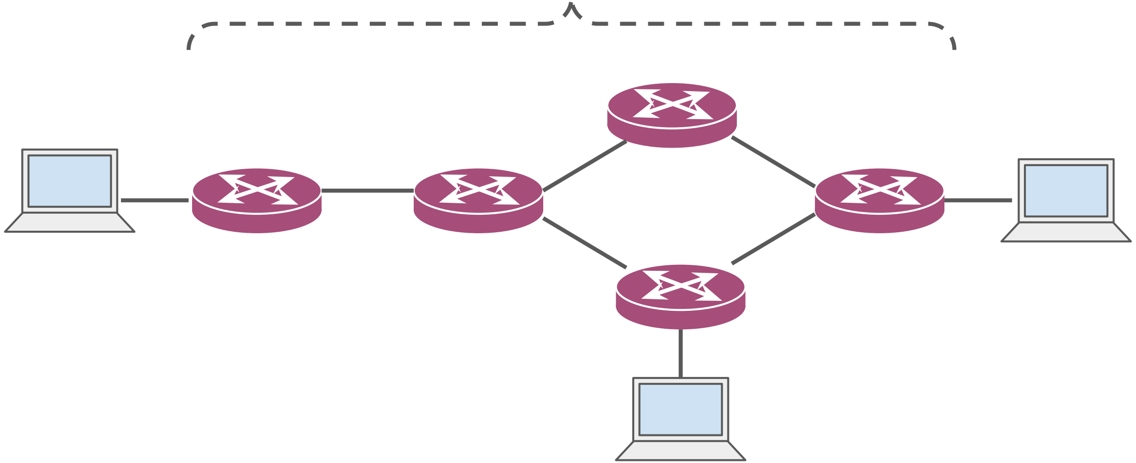
Recurring theme in this course

Abstraction

Automation

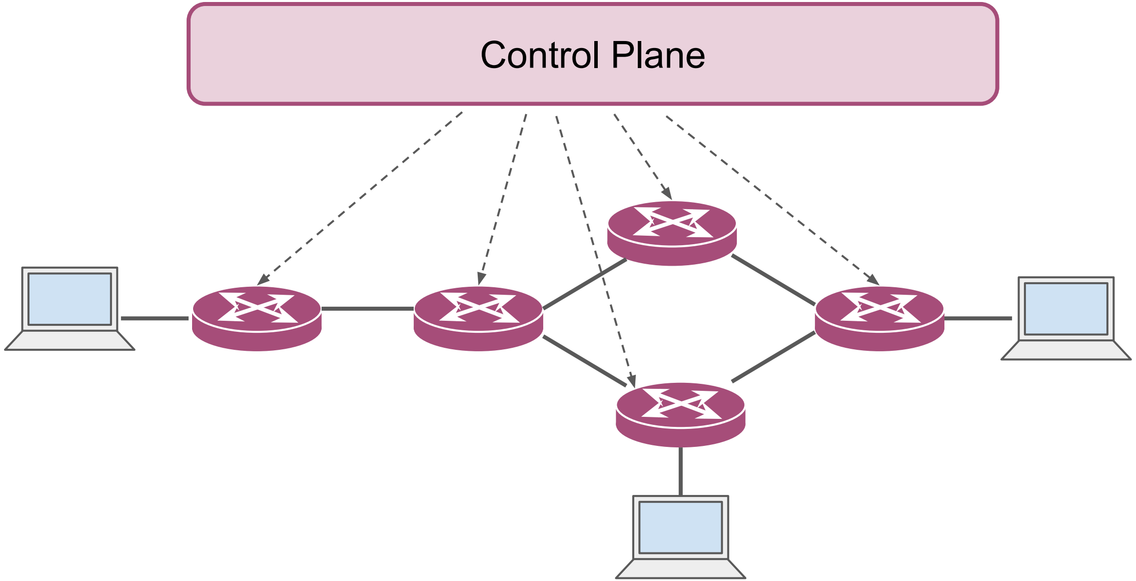
Here are some examples…

Configure a pre-defined set of distributed protocols (e.g., OSPF, BGP, etc.) to pick your desired forwarding paths.



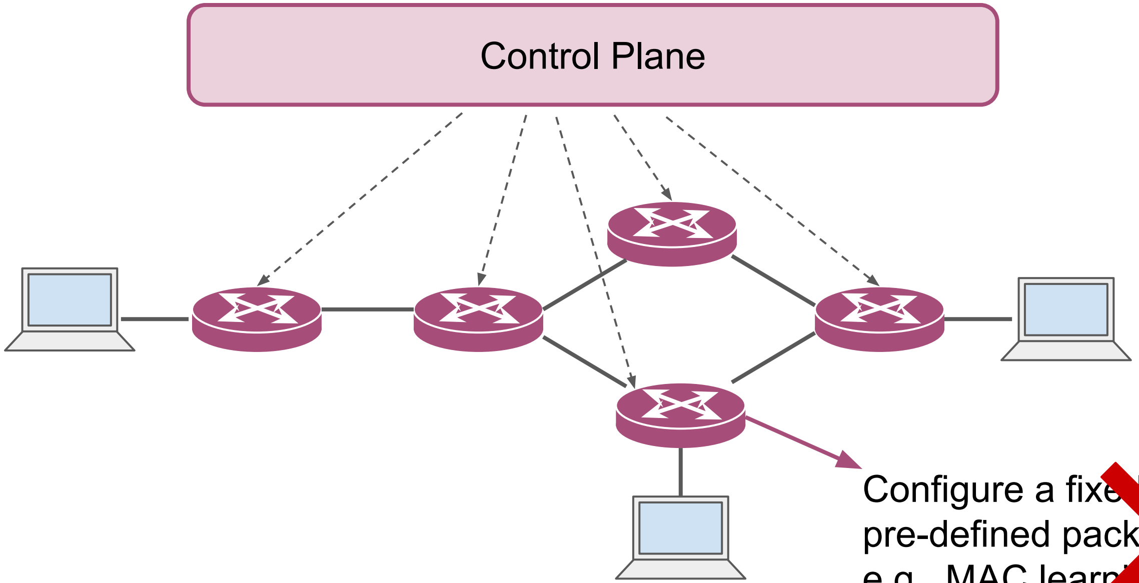
Here are some examples…

- Write a program that decides the forwarding paths.
- Have a runtime compute and communicate proper configurations to network devices.



Here are some examples…

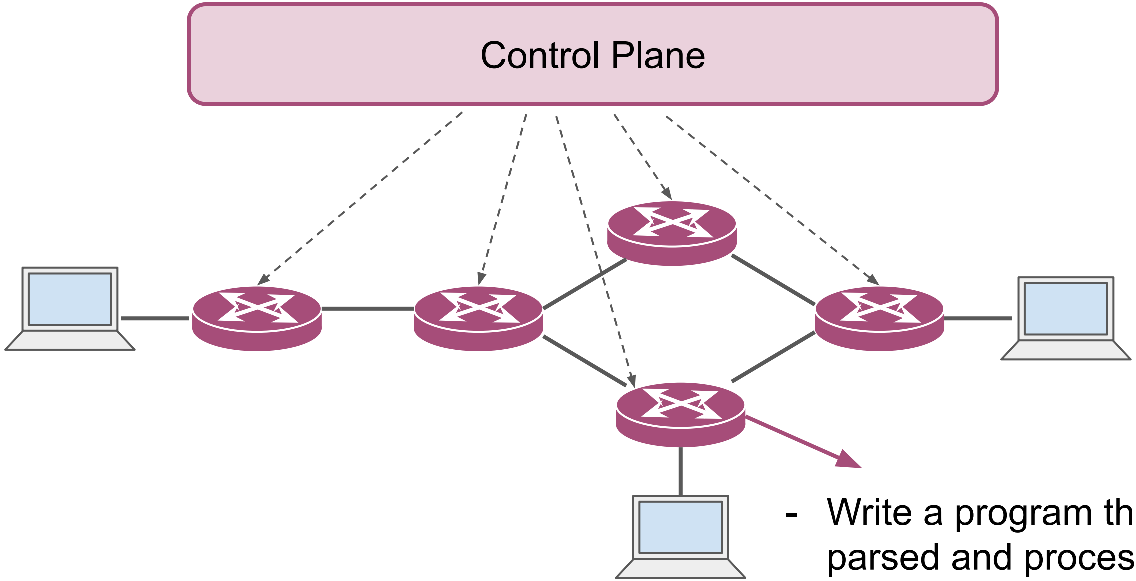
- Write a program that decides the forwarding paths.
- Have a runtime compute and communicate proper configurations to network devices.



~~Configure a fixed-function hardware with pre-defined packet processing steps, e.g., MAC learning → GRE-Tunnel Processing → IP forwarding~~

Here are some examples…

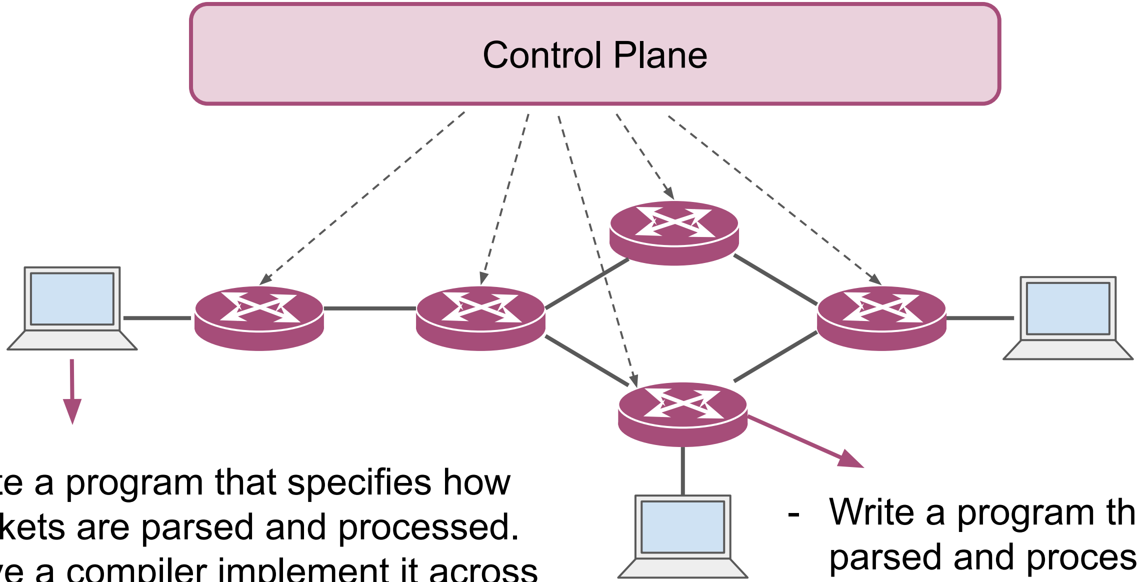
- Write a program that decides the forwarding paths.
- Have a runtime compute and communicate proper configurations to network devices.



- Write a program that specifies how packets are parsed and processed.
- Have a compiler translate that into instructions for switch hardware.

Here are some examples…

- Write a program that decides the forwarding paths.
- Have a runtime compute and communicate proper configurations to network devices.

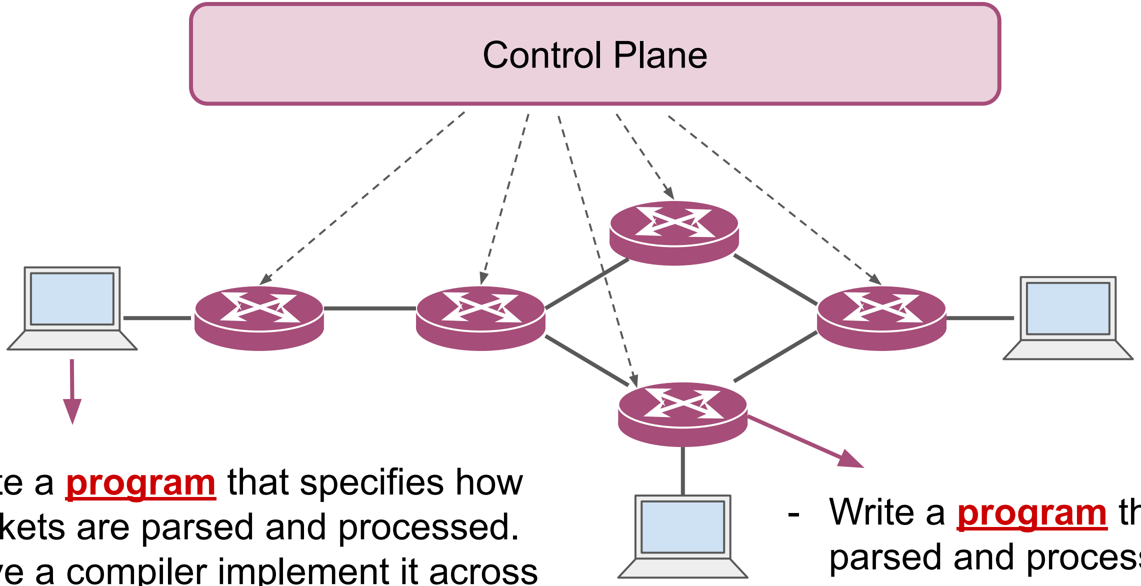


- Write a program that specifies how packets are parsed and processed.
- Have a compiler implement it across user-space, the Kernel, and hardware accelerators.

- Write a program that specifies how packets are parsed and processed.
- Have a compiler translate that into instructions for switch hardware.

Here are some examples…

- Write a **program** that decides the forwarding paths.
- Have a runtime compute and communicate proper configurations to network devices.

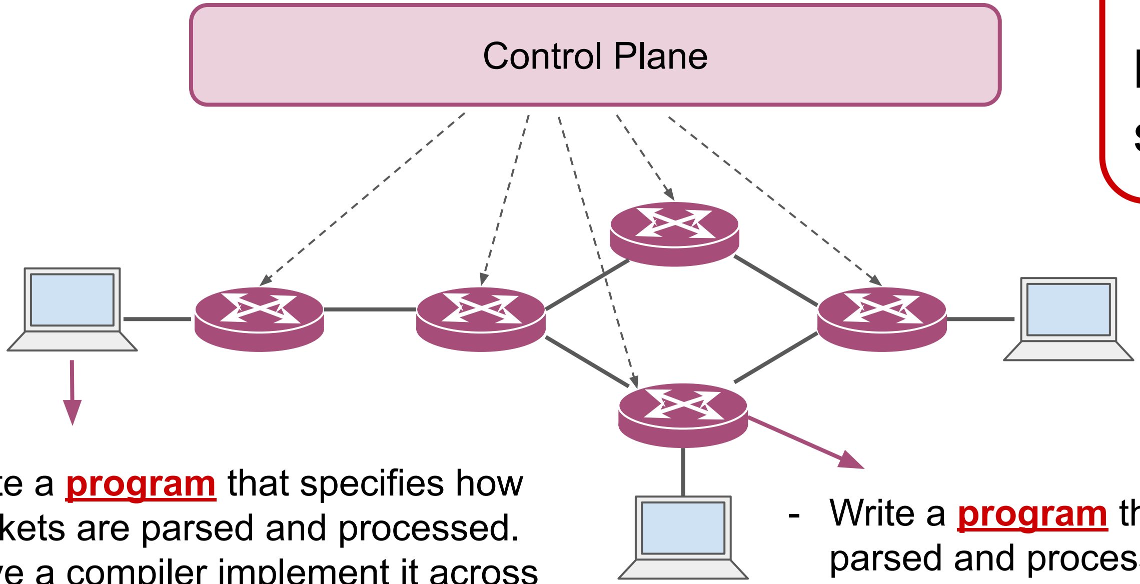


- Write a **program** that specifies how packets are parsed and processed.
- Have a compiler implement it across user-space, the Kernel, and hardware accelerators.

- Write a **program** that specifies how packets are parsed and processed.
- Have a compiler translate that into instructions for switch hardware.

Here are some examples…

- Write a **program** that decides the forwarding paths.
- Have a runtime compute and communicate proper configurations to network devices.



Treat the network as a big, distributed, and specialized computer

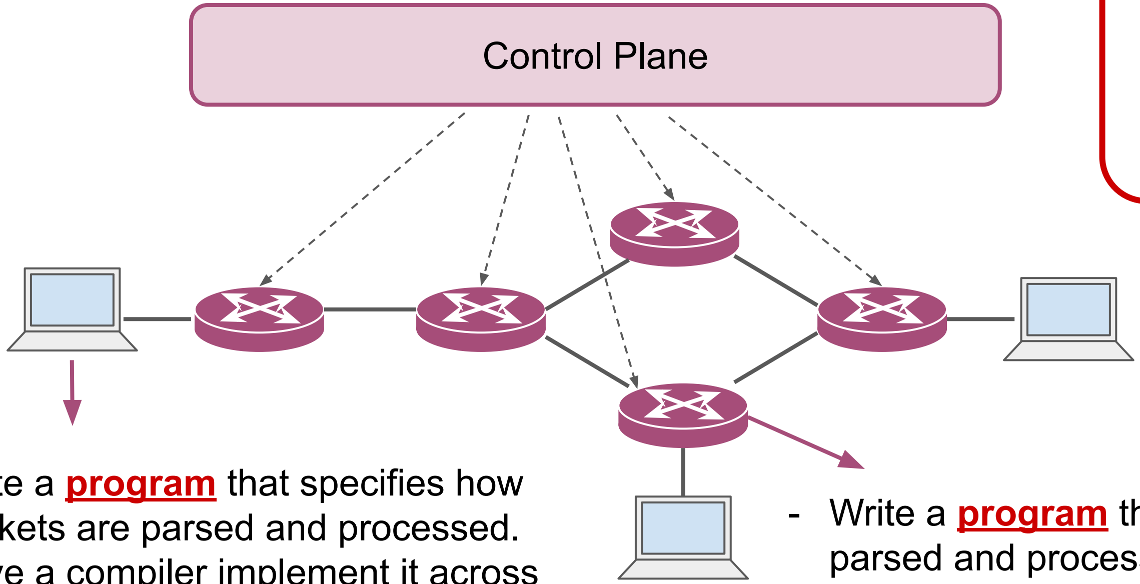
- Write a **program** that specifies how packets are parsed and processed.
- Have a compiler implement it across user-space, the Kernel, and hardware accelerators.

- Write a **program** that specifies how packets are parsed and processed.
- Have a compiler translate that into instructions for switch hardware.

Here are some examples…

- Write a **program** that decides the forwarding paths.
- Have a runtime compute and communicate proper configurations to network devices.

Programmable Networks



- Write a **program** that specifies how packets are parsed and processed.
- Have a compiler implement it across user-space, the Kernel, and hardware accelerators.

- Write a **program** that specifies how packets are parsed and processed.
- Have a compiler translate that into instructions for switch hardware.

When we can "program" the network…

We can

- Analyze high-level programs to verify network functionality
- Customize network devices to process packets exactly how we need
    - measure fine-grained statistics about traffic
    - add a variety of signals about congestion to packets for end-to-end congestion control algorithms
    - implement sophisticated and customized packet scheduling algorithms to provide quality of service (QoS) guarantees
    - accelerate distributed applications (!)
    - …
- …

In this course, we will discuss

- (Programming) abstractions and automation applied to different components of networks
- How they have improved networks
- The new functionalities and tools they have enabled
- Open research questions in the area

Logistics

- Class is Tuesdays and Thursdays, 12:00pm to 1:20pm.
- Thursdays: lecture followed by discussion
    - Lay of the land for that topic
    - Context about the papers we want to read
- Tuesdays: Paper discussion

Logistics - Continued

- Instructor is me! Email me for any questions and to request office hours
    - prefix the email with [CS856] for a timely reply
- We will use Piazza for announcements, questions, and discussions.
- Project submissions and grades will be through LEARN.

Course Components

- Reviews (20%)
- Paper Presentation (15%)
- In-class Discussion (10%)
- Assignment (5% + Bonus)
- Project (50%)

Reviews

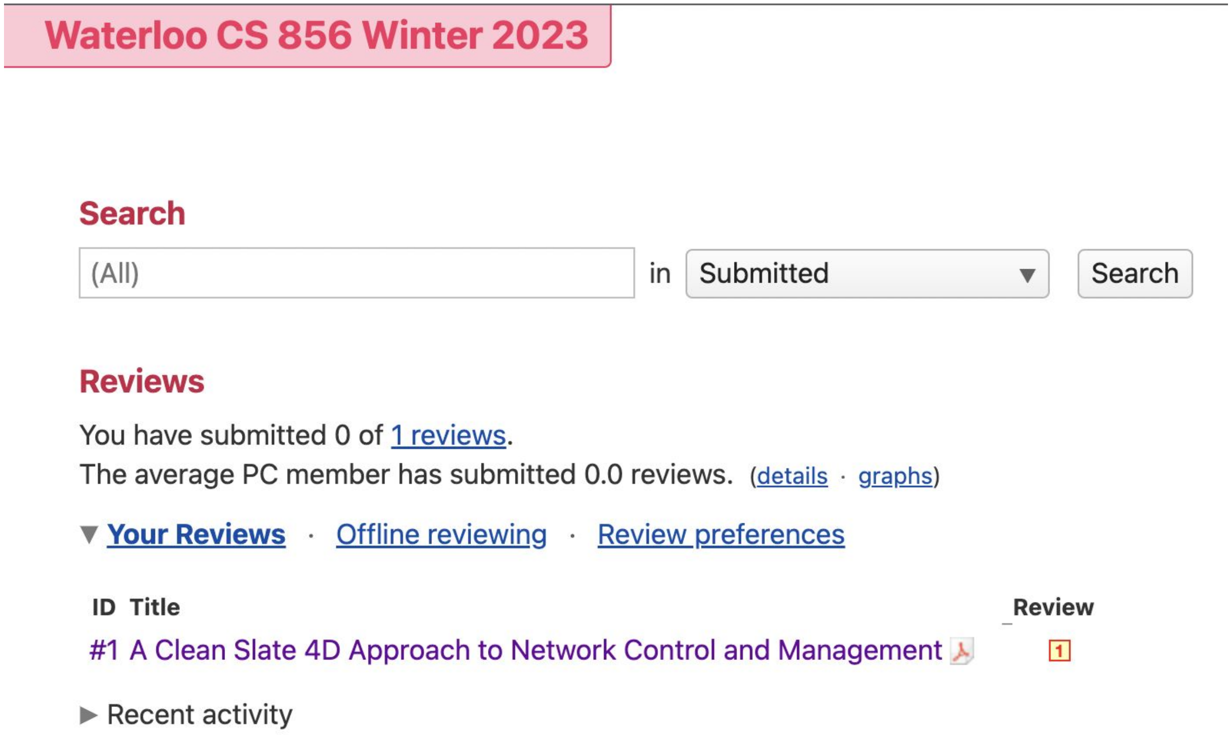
- Two papers each week
- Due on **<u>Mondays at 5pm EST.</u>**
- Will be visible (anonymously) afterwards, so make sure to check them before class on Tuesday.
- Review grading
  - Complete (2 points): adheres to the reviewing guidelines (next slide), clearly demonstrates that the reviewer has read and thought about the paper.
  - Partially Complete (1 point): Misses some but not all the reviewing guidelines, demonstrates that the reviewer has some understanding of the paper.
  - Incomplete (0 points)

Reviewing Guidelines

Each review should be ~500 words and contain the following sections, following the typical format of reviews in networking and systems conferences:

- A concise **summary** of the paper (1 paragraph)
- A list of the paper's main **strengths** (at least 2 bullet points)
- A list of **opportunities for improvement** (at least 2 bullet points)
- **Critical analysis** and comments (justifying the strengths and improvement opportunities listed in the previous sections)
- **Trade-offs:** There is almost never a free lunch! a paragraph or two about the trade-off space that is relevant to the proposed approach of the paper, and where the proposed approach is in that trade-off space.

Reviewing Platform: HotCRP



Waterloo CS 856 Winter 2023

Search

(All) in Submitted ▼ Search

Reviews

You have submitted 0 of 1 reviews.
The average PC member has submitted 0.0 reviews. (details · graphs)

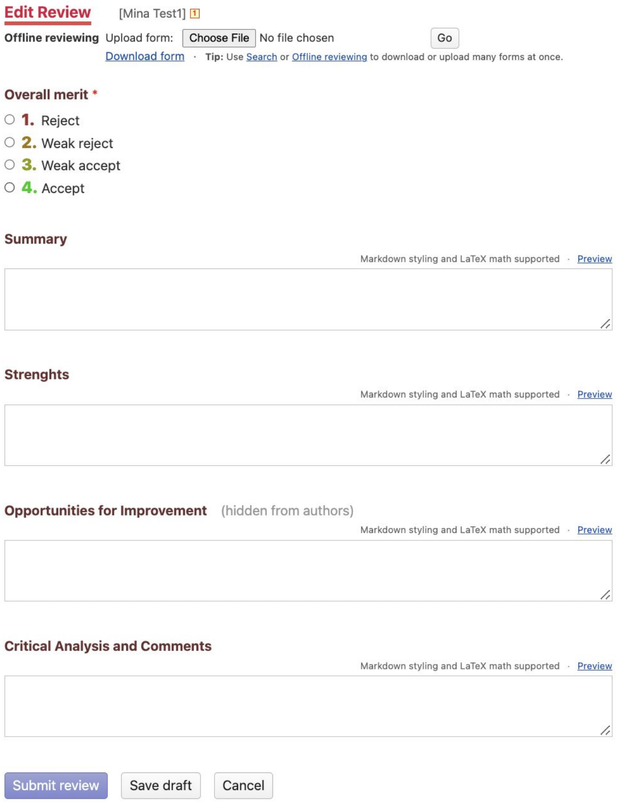
▼ **Your Reviews** · Offline reviewing · Review preferences

ID	Title	Review
#1	A Clean Slate 4D Approach to Network Control and Management	1

▶ Recent activity

Reviewing Platform: HotCRP

- When ready, submit review
- Every Monday at 5pm, the review form is deactivated and you can see all the other reviews submitted for the paper.



Edit Review [Mina Test1]

Offline reviewing Upload form: Choose File No file chosen Go

Download form · **Tip:** Use Search or Offline reviewing to download or upload many forms at once.

Overall merit *

○ **1.** Reject
○ **2.** Weak reject
○ **3.** Weak accept
○ **4.** Accept

Summary

Markdown styling and LaTeX math supported · Preview

Strenghts

Markdown styling and LaTeX math supported · Preview

Opportunities for Improvement (hidden from authors)

Markdown styling and LaTeX math supported · Preview

Critical Analysis and Comments

Markdown styling and LaTeX math supported · Preview

Submit review Save draft Cancel

Paper Presentation

- Each Paper discussion starts by a 10-minute presentation:
  - Describe the context and motivation behind the paper
  - The main problem the paper is trying to solve
  - The main design choices and/or techniques used in the solution
  - A summary of evaluation results
  - 4-5 discussion questions
- Each student is expected to do 1-2 presentations
- Feel free to send me a draft a few days before for feedback

Programming Assignment

- Assignment 1 (5%): implement a simple network functionality using P4
- Assignment 2 (Optional, 5% bonus): analyze the correctness of a simple network functionality using existing analysis tools
- The assignments are quite light
- The main purpose is for you to just install and use the tools, specially since P4 is used/mentioned in many papers.

Project

- Individually or in groups of two.
- Original research projects related to programmable networks.
- Run your project idea by the instructor before submitting the proposal.
- **One-Page Proposal (Jan 31)**
  - problem statement, context and motivation, and a high-level overview of related work
- **Two-Page Progress Report (March 2)**
- **Presentation (Last week of March)**
- **Final Project Report (April 15)**
  - 6-page conference-style paper
  - problem statement and motivation, design, evaluation, related work, and future research directions

Final Remarks

- Seminar courses are only as good as the discussions we have.
- Be active, ask questions, and voice your opinion.
- There are no bad ideas, and I mean it 🙂
- If you have a hard time speaking up, let me know and I'll make sure to provide space for you to voice your opinion.
- Be mindful of others in discussions.