UNIVERSITY OF WATERLOO

CS 856: Programmable Networks

Mina Tahmasbi Arashloo

Winter 2025

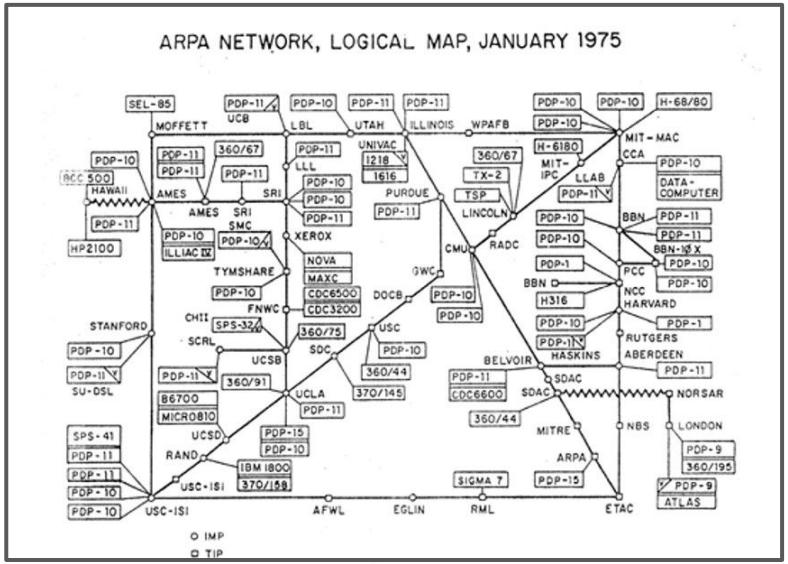
Networks when they started (1970s)

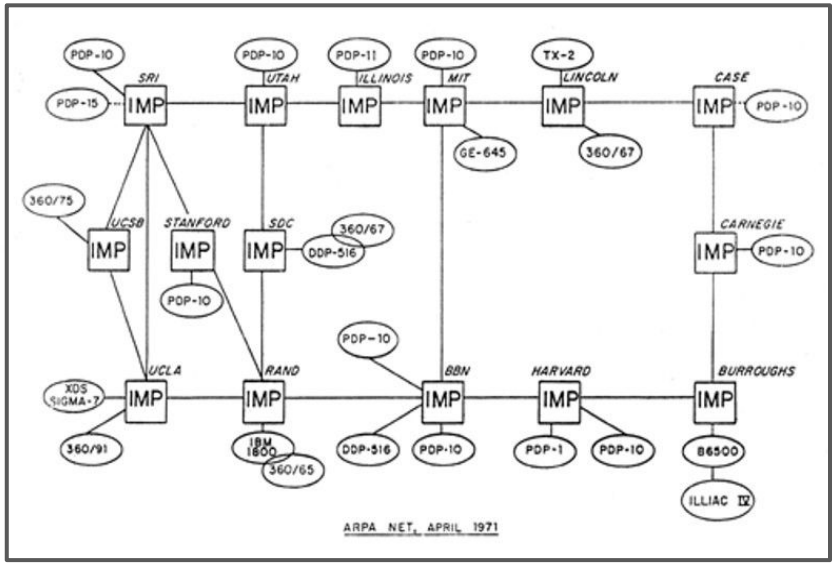
- Small and simple

Networks when they started (1970s)

- Small and simple

Tens of nodes





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

Networks when they started (1970s)

- Small and simple
- A scientific experiment

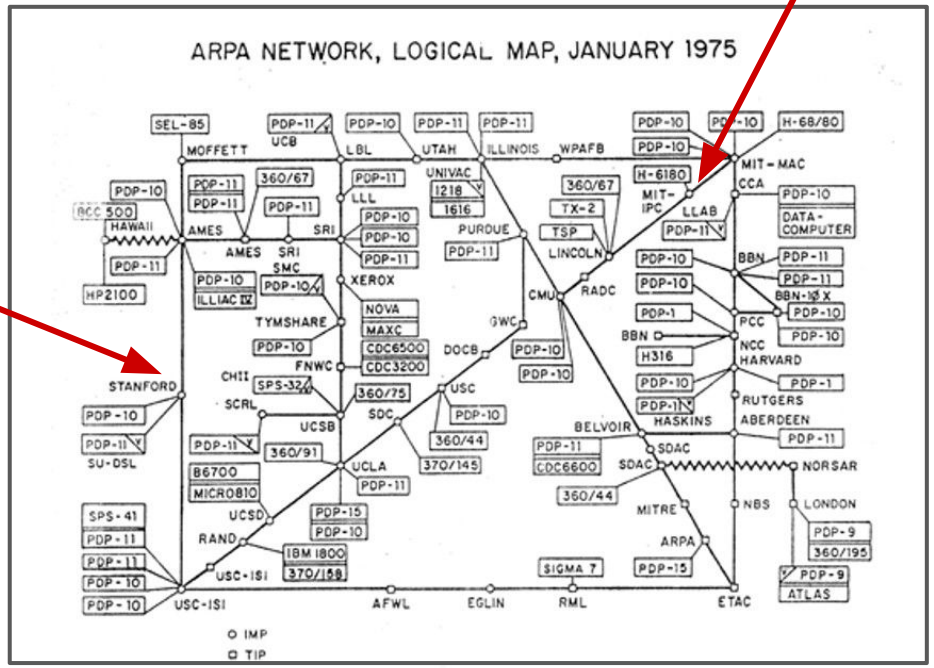
Networks when they started (1970s)

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

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

A

B



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)

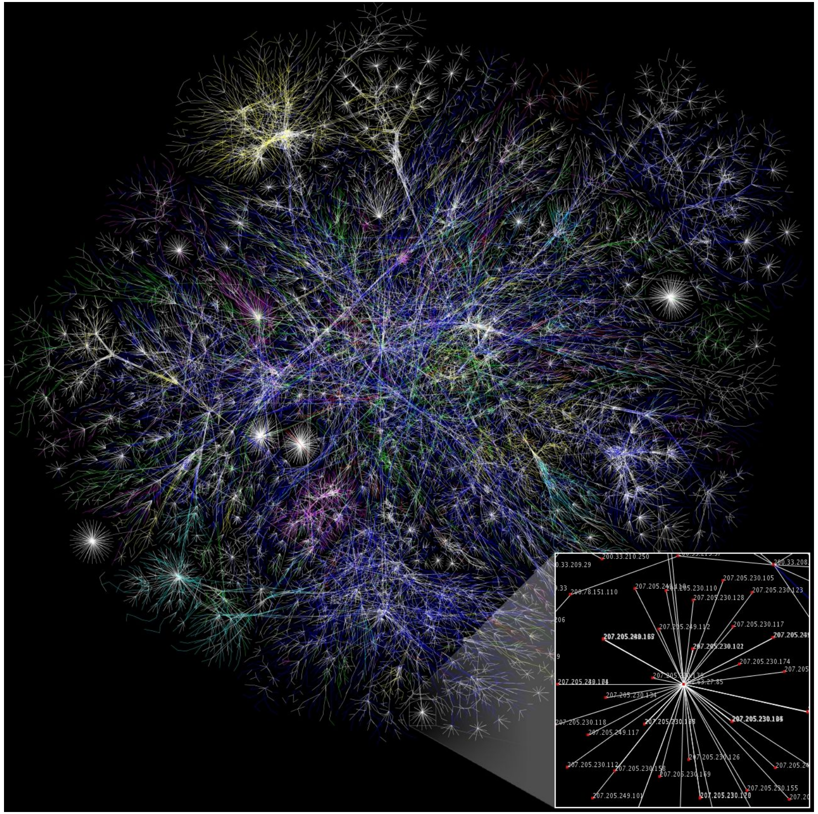
- Small and simple
- A scientific experiment
- Few simple requirements

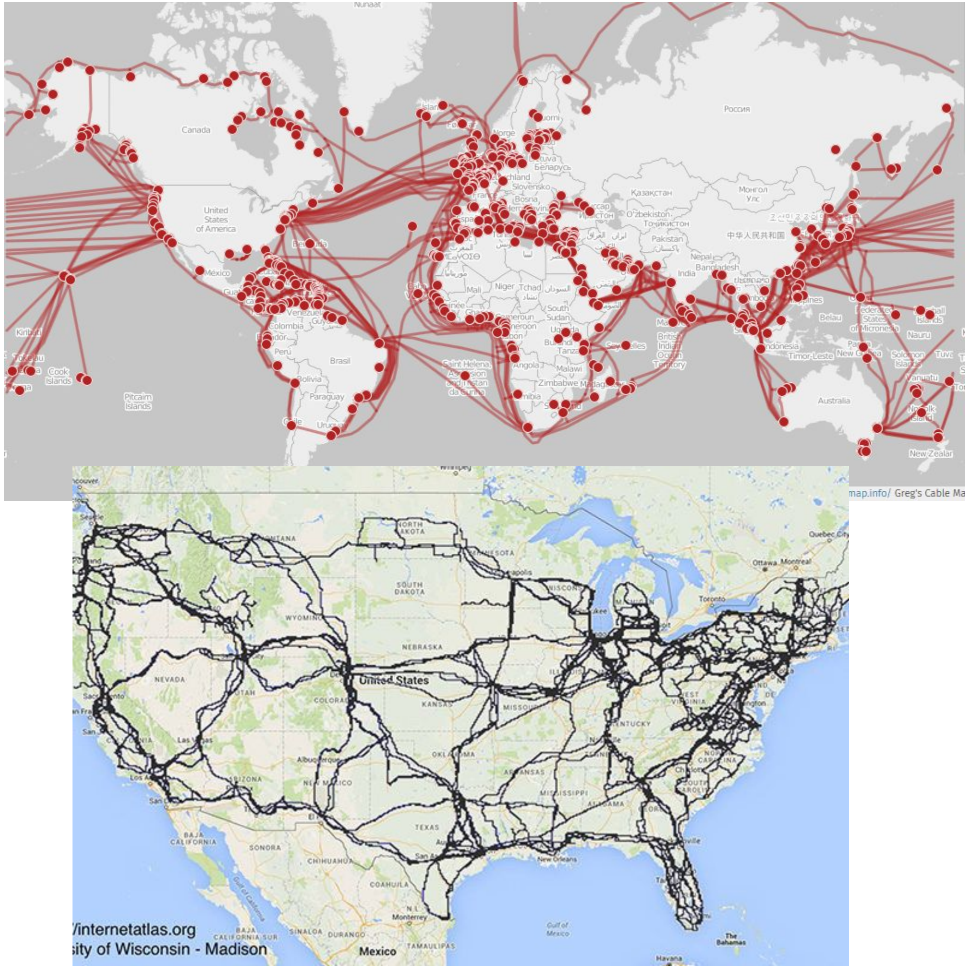
Networks today (2020s)

- 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

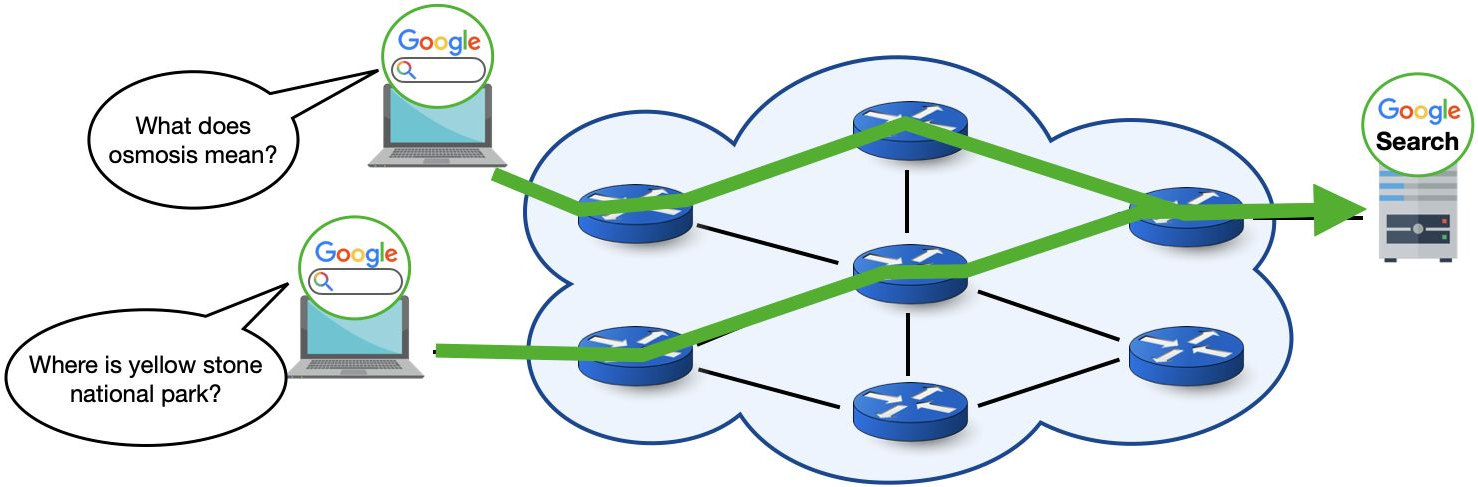



map.info/ Greg's Cable Ma

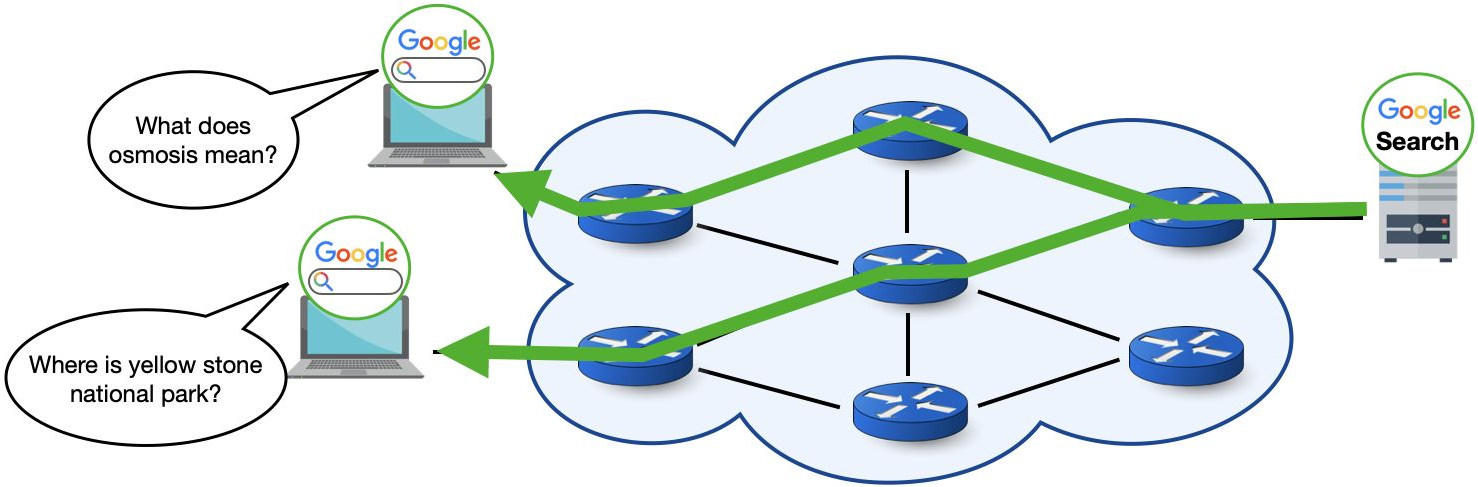
/internetatlas.org
sity of Wisconsin - Madison

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

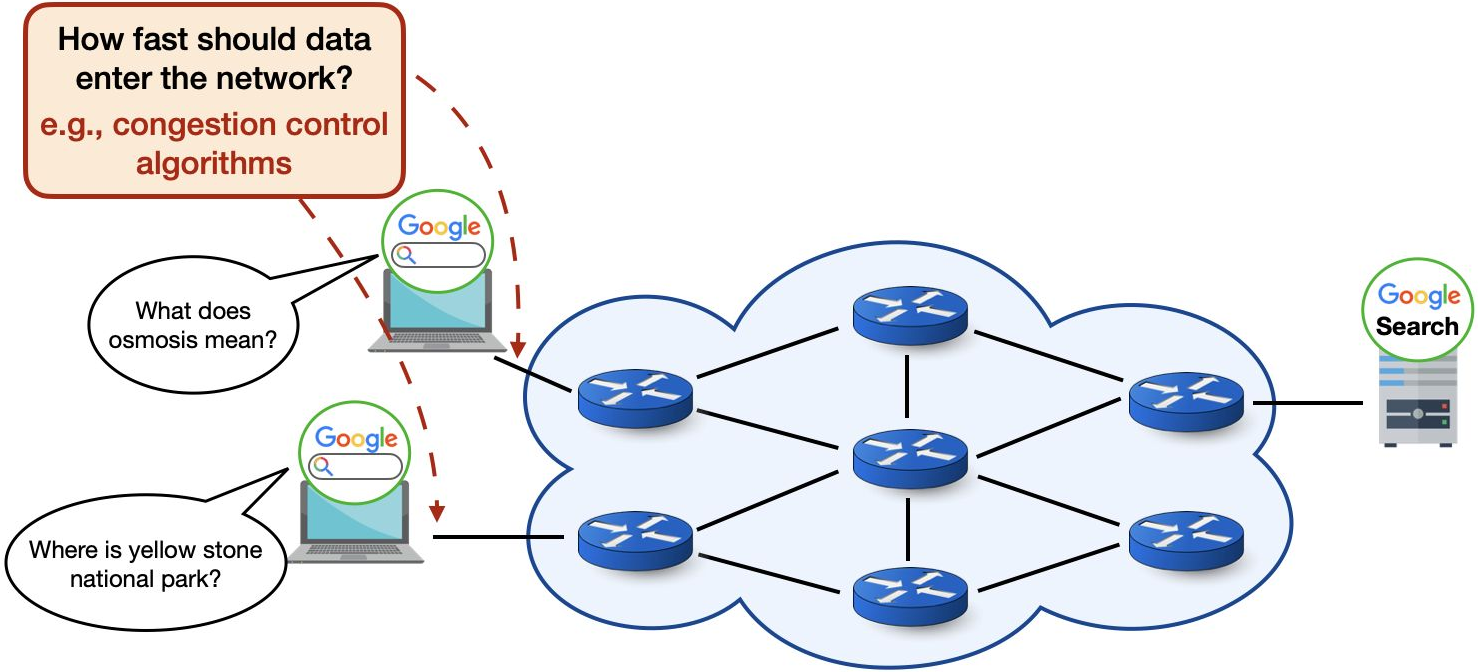
Example Network



Example Network



Example Algorithms and Protocols



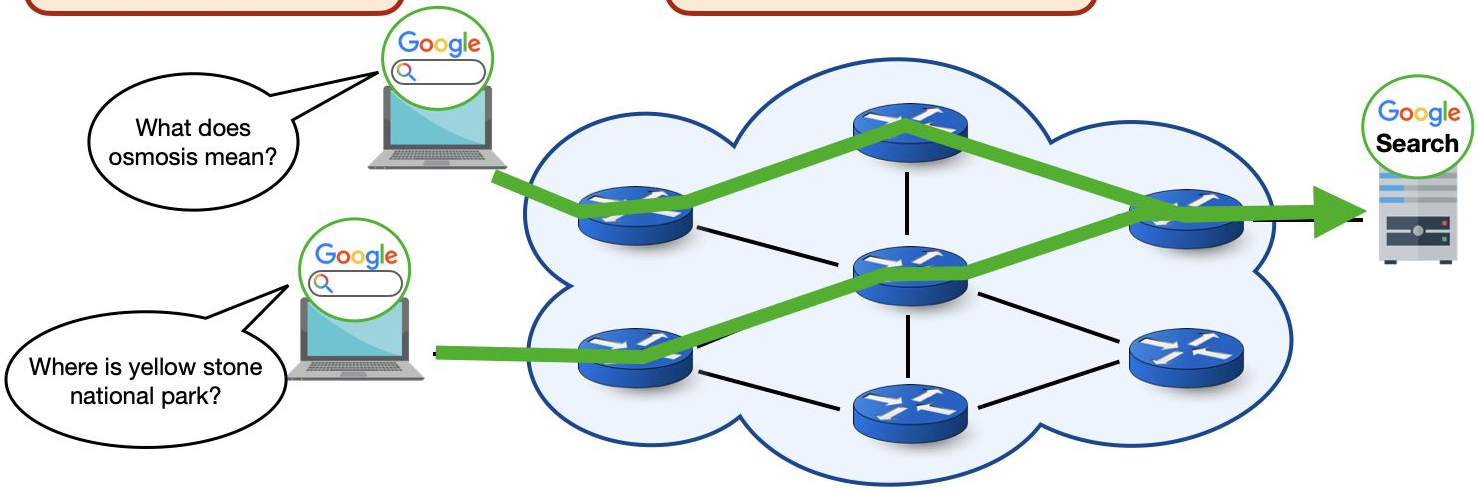
Example Algorithms and Protocols

How fast should data enter the network?

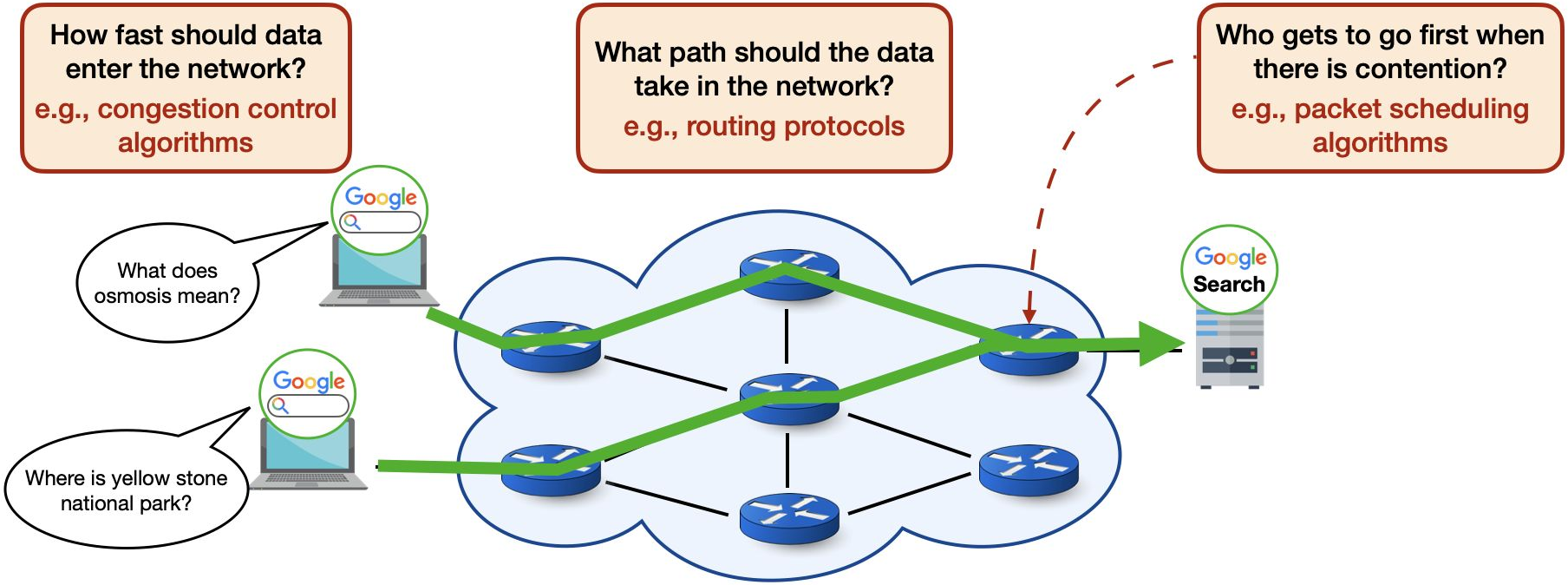
e.g., congestion control algorithms

What path should the data take in the network?

e.g., routing protocols



Example Algorithms and Protocols

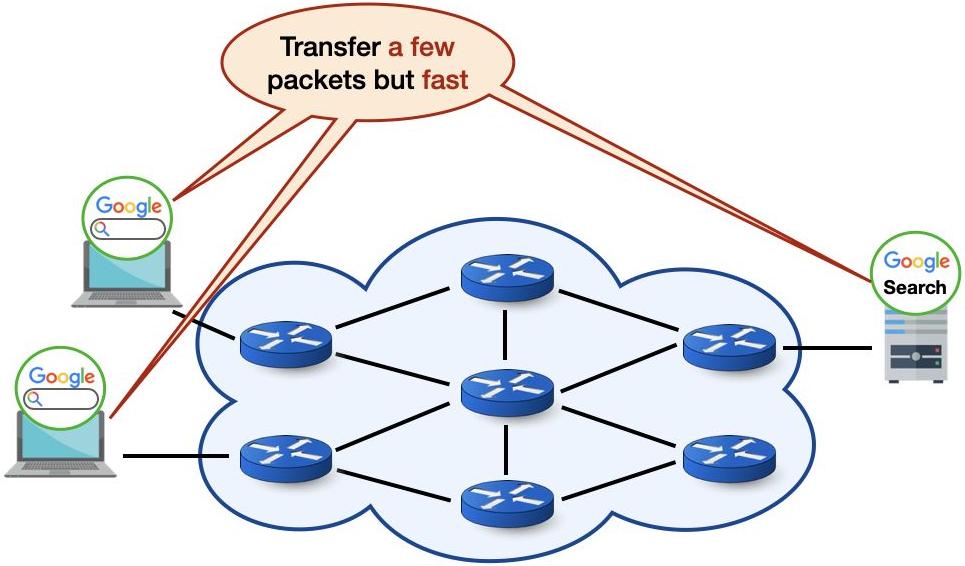


Small Network, One Application, A Few Endpoints

How fast to transmit?

What path to pick?

Who goes first?



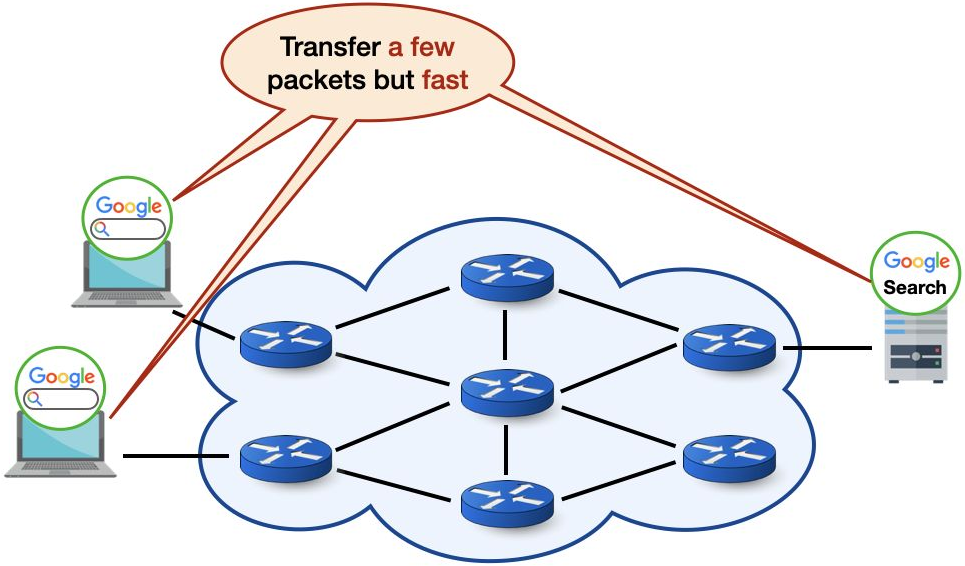
Small Network, One Application, A Few Endpoints

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

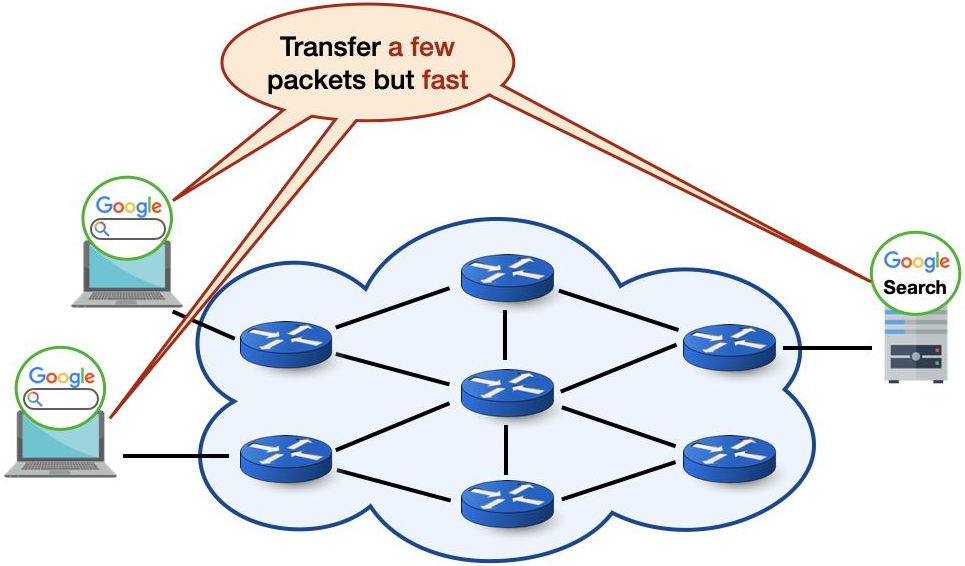
How fast to transmit?

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?

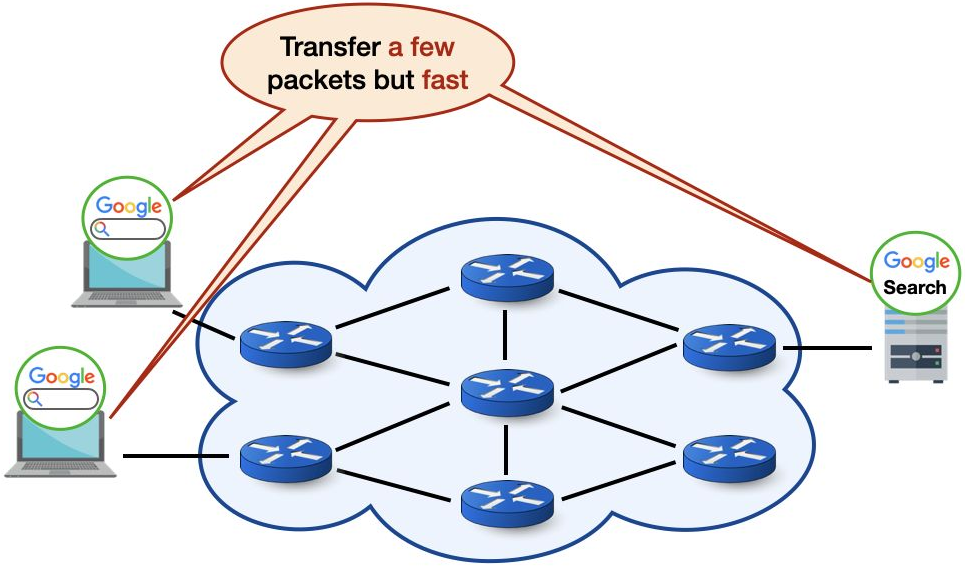
Start fast and back off on loss.

What path to pick?

Pick one of the shortest path at random.

Who goes first?

First come, first serve.

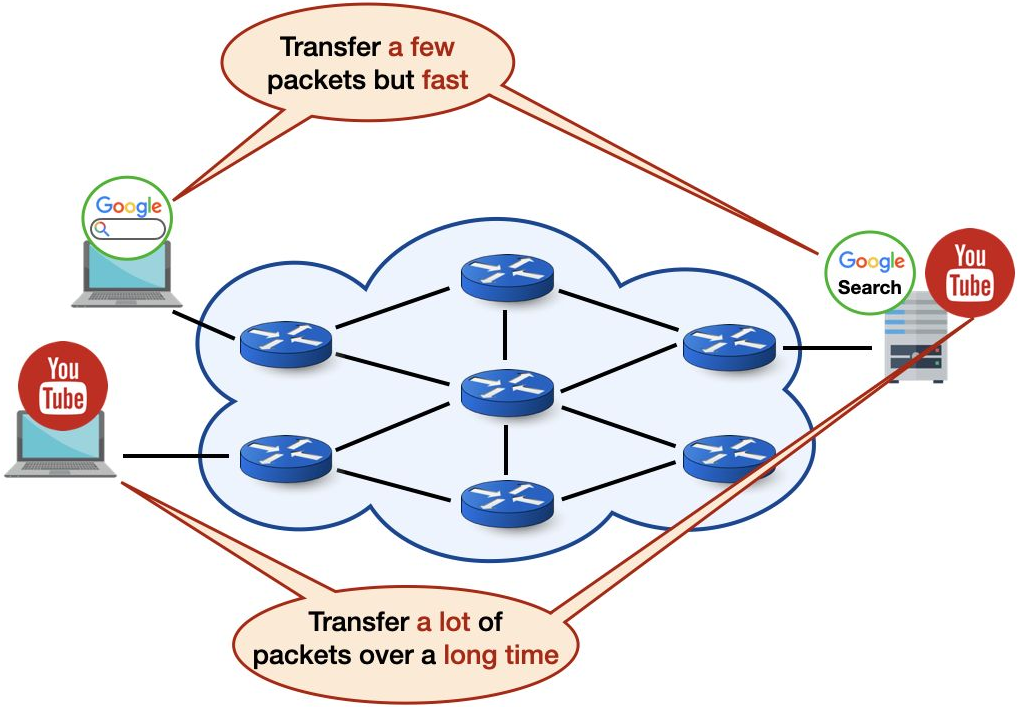


Small Network, More Applications, A Few Endpoints

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.



Small Network, More Applications, A Few Endpoints

How fast to transmit?

Start fast and back off on loss.

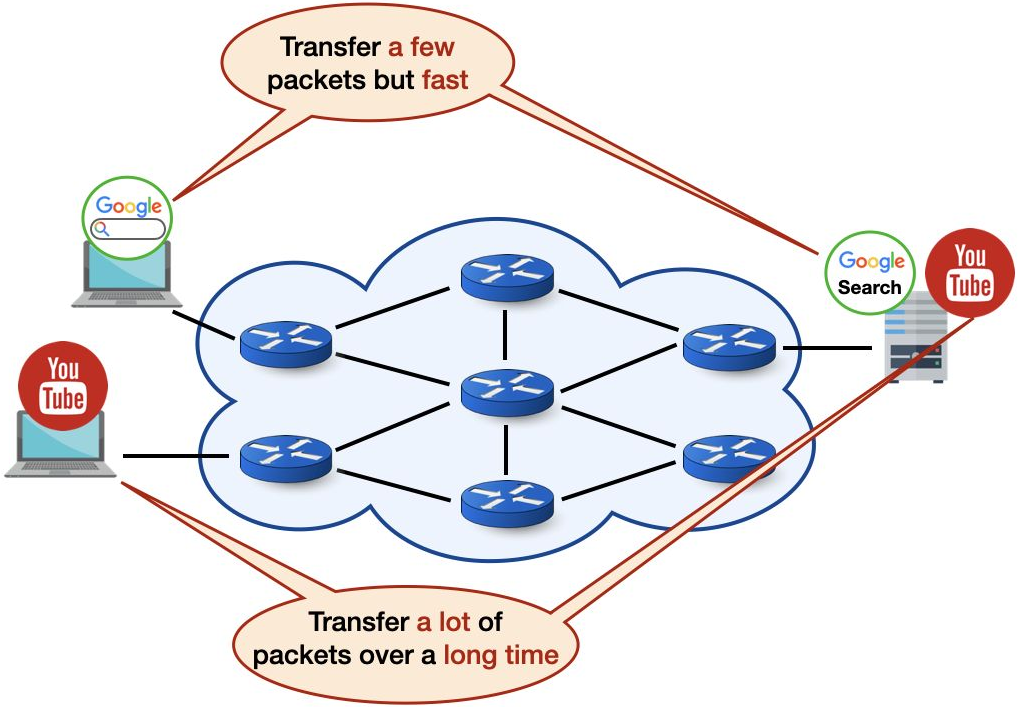
What path to pick?

~~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.



Small Network, More Applications, A Few Endpoints

How fast to transmit?

Start fast and back off on loss.

What path to pick?

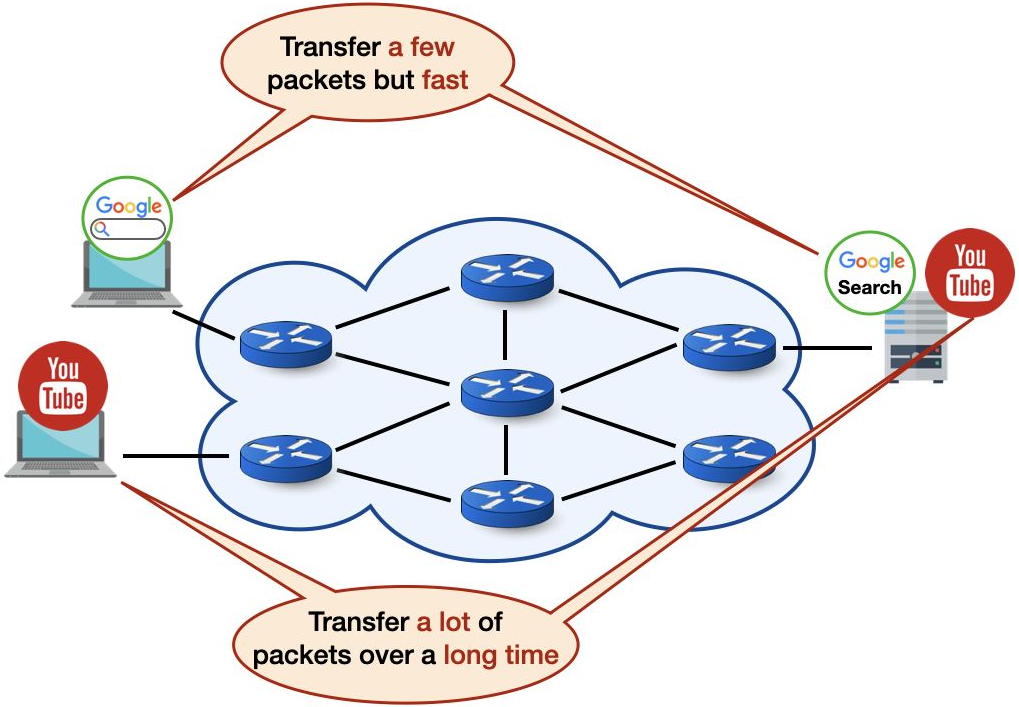
~~Pick one of the shortest path at random.~~

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

Who goes first?

~~First come, first serve.~~

Prioritize search over video.



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.~~

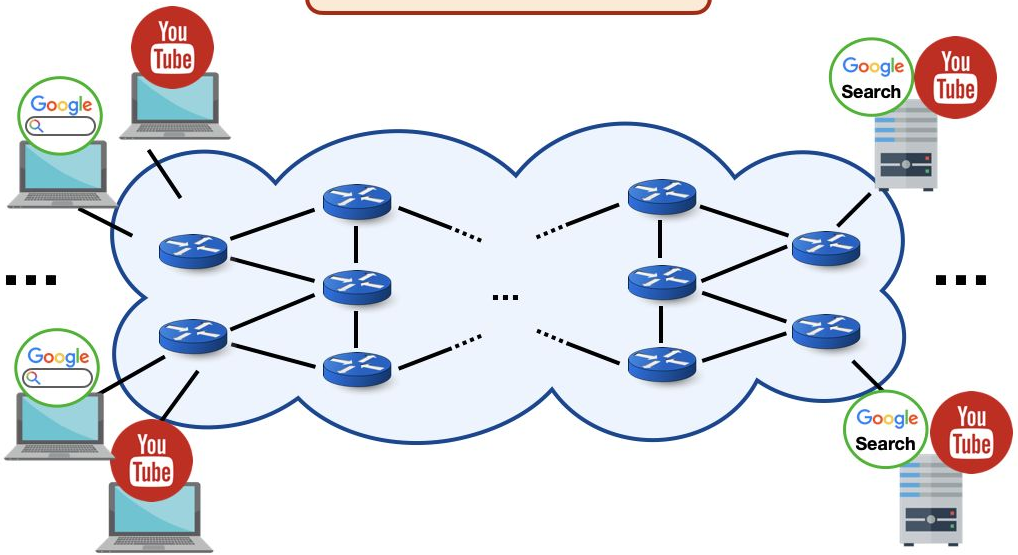
Pick the least loaded path so search traffic avoids video traffic.

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.~~

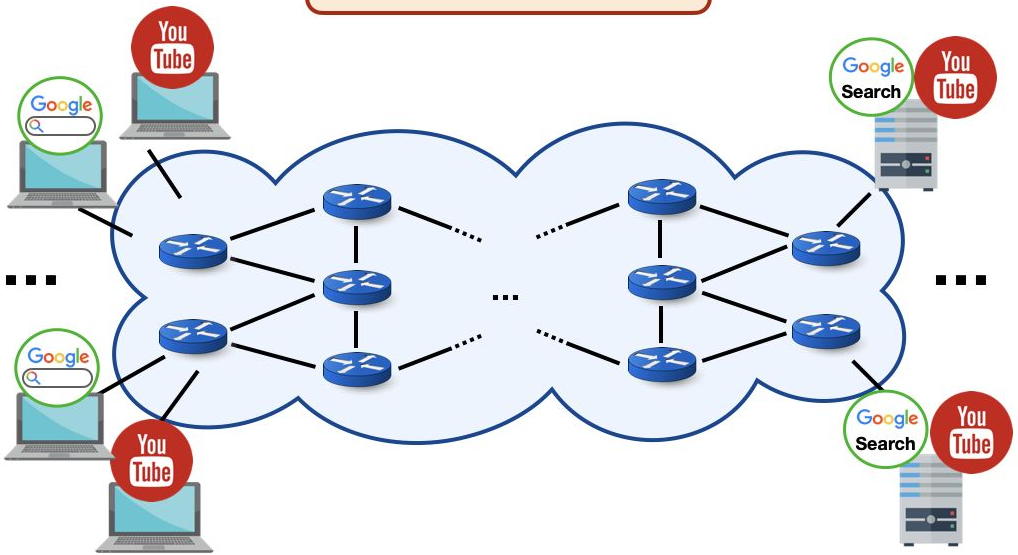
Pick the least loaded path so search traffic avoids video traffic.

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.

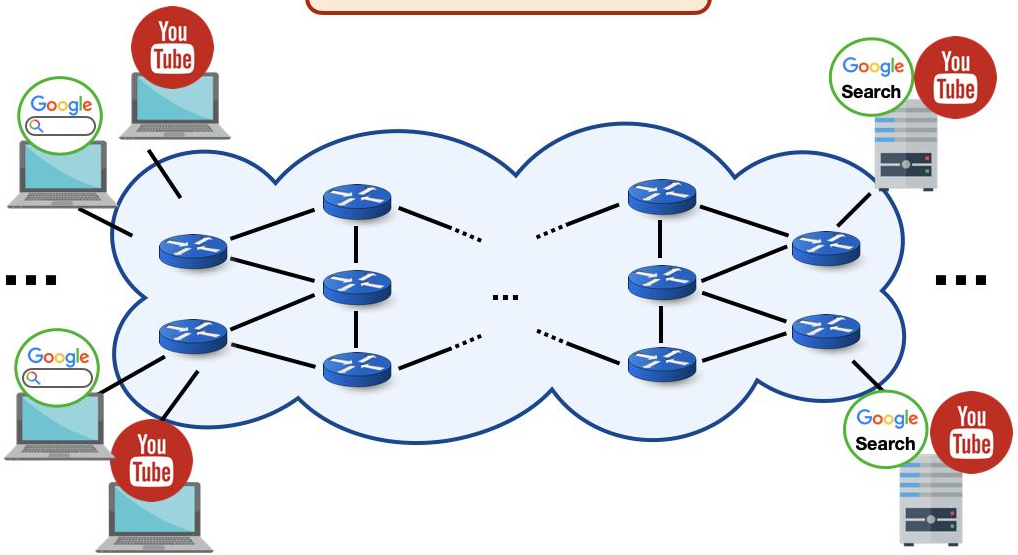
Where do I implement them?

On the edge switches and two of the cores.

How much time do I have?

1μs per packet.

1000x more flows
100x more switches

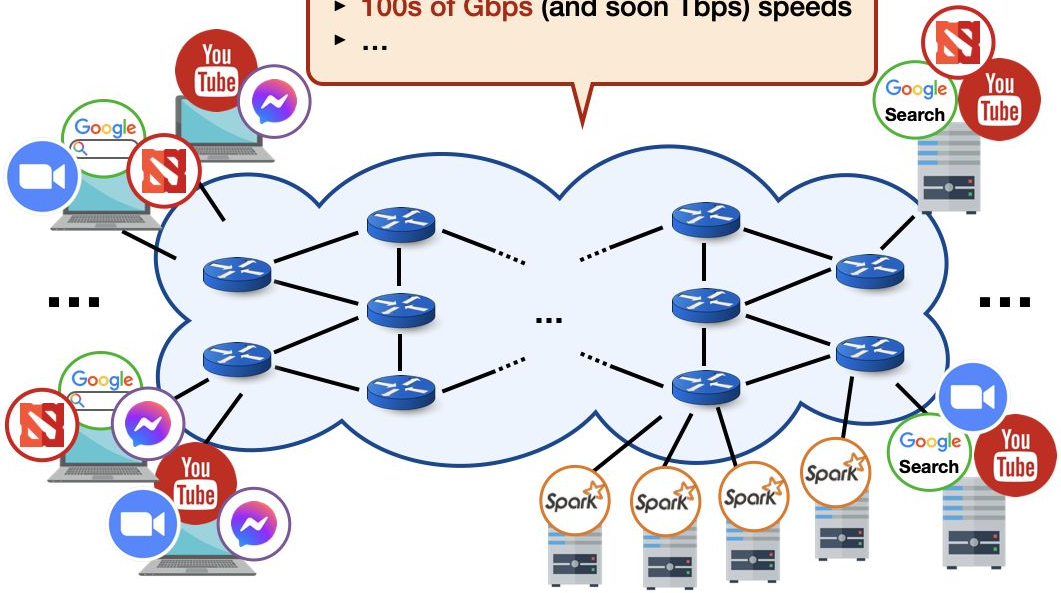


Diverse Applications

Large Scale and High Speed

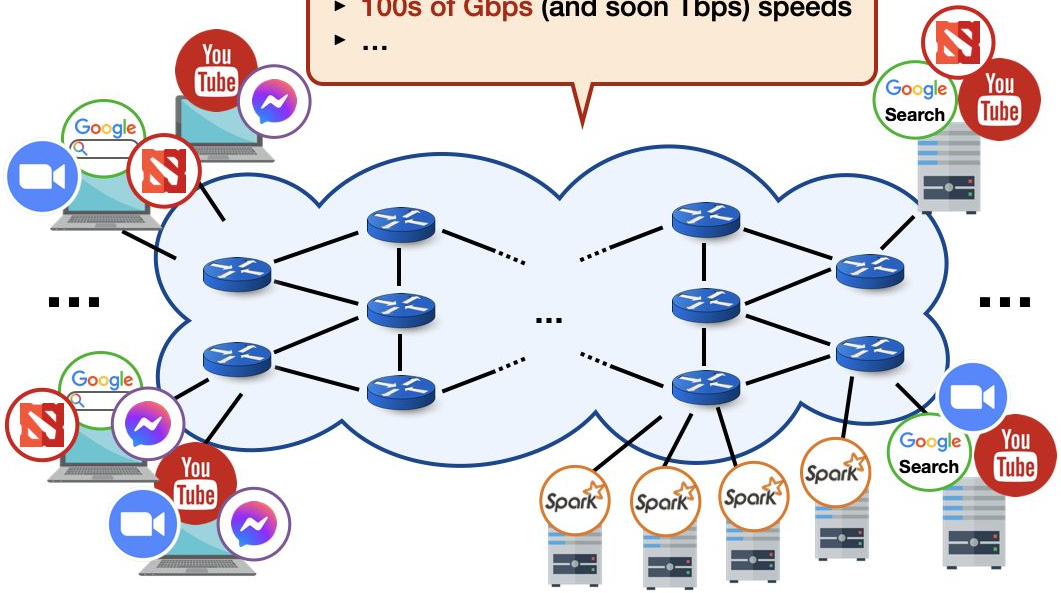
Constantly Evolving Algorithms & Protocols

- 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?

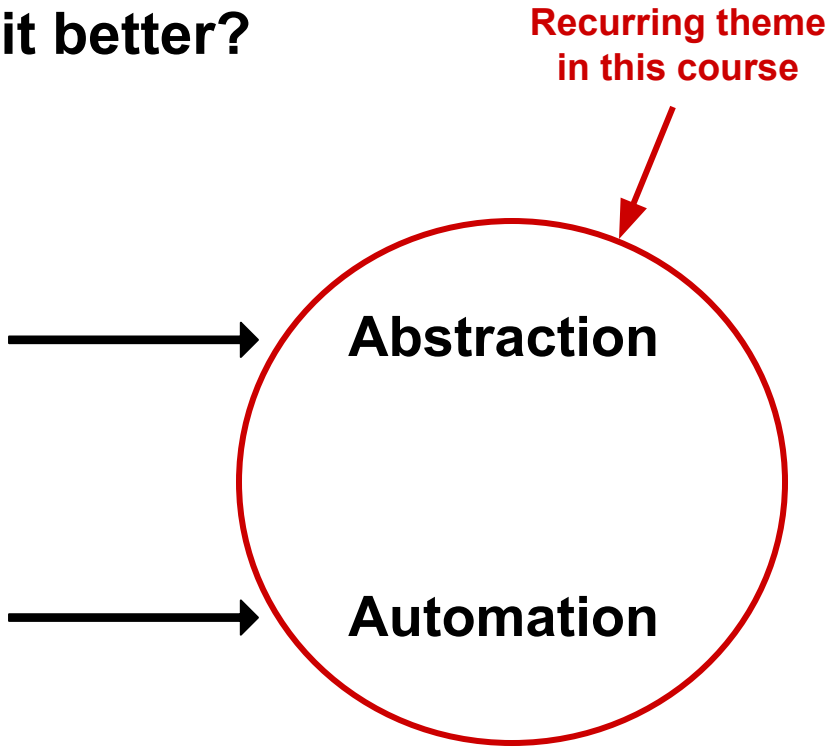
Separate *what* you want the network to do from *how* it is implemented → **Abstraction**

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

How can we make it better?

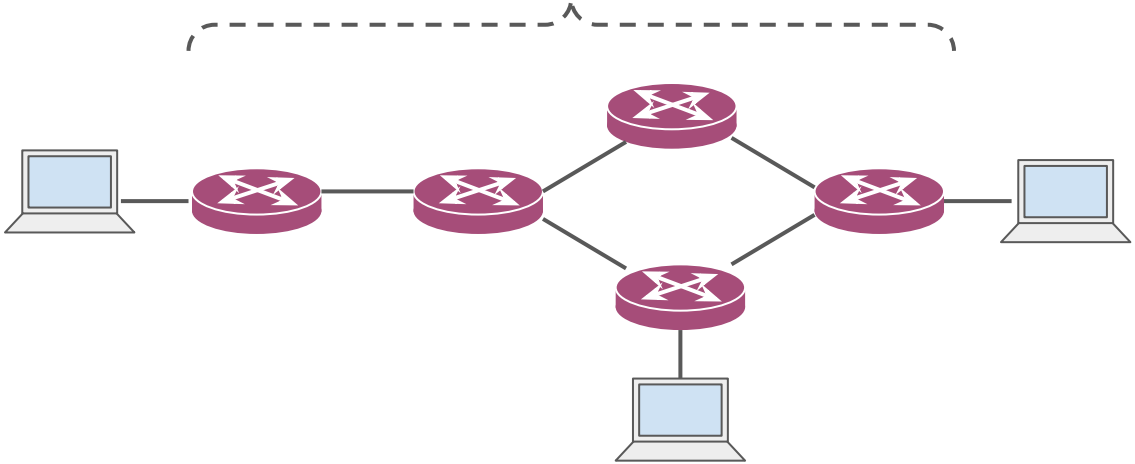
Separate *what* you want the network to do from *how* it is implemented

Don't implement in *manually* 🙂



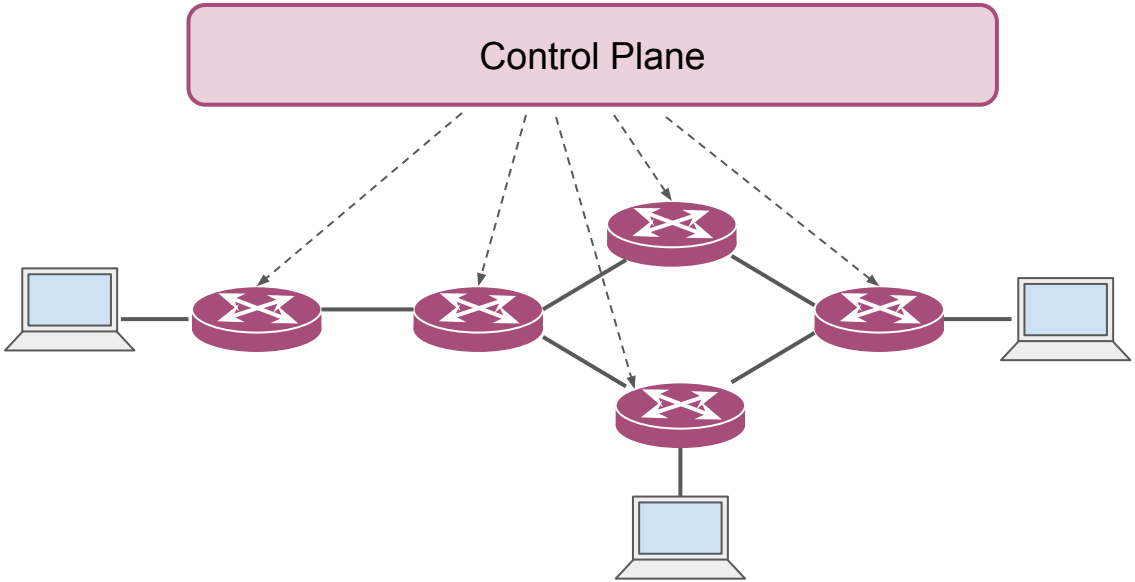
Here are some examples…

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



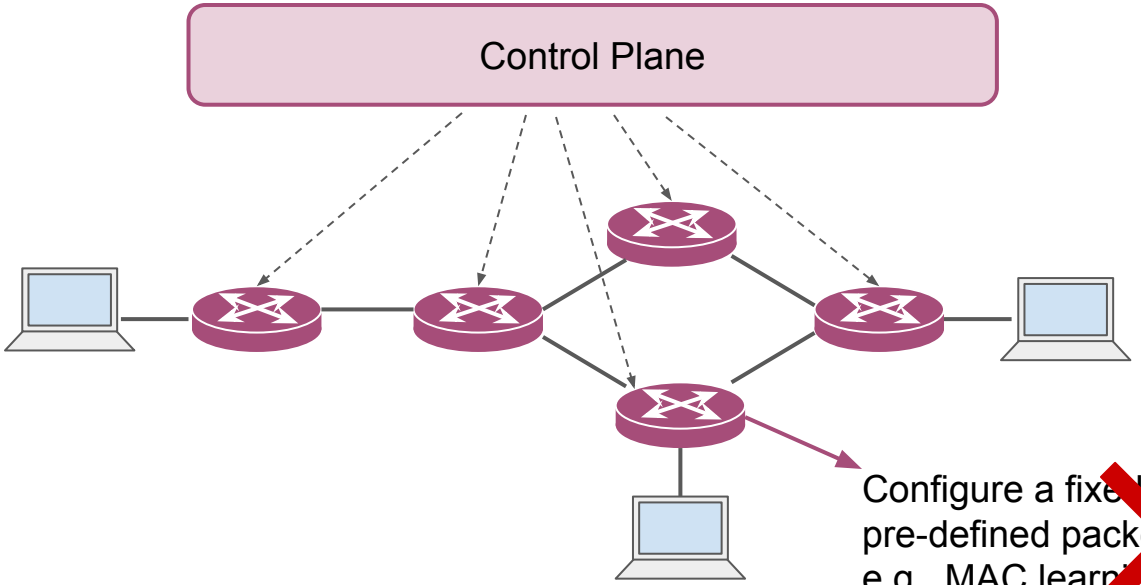
Here are some examples…

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



Here are some examples…

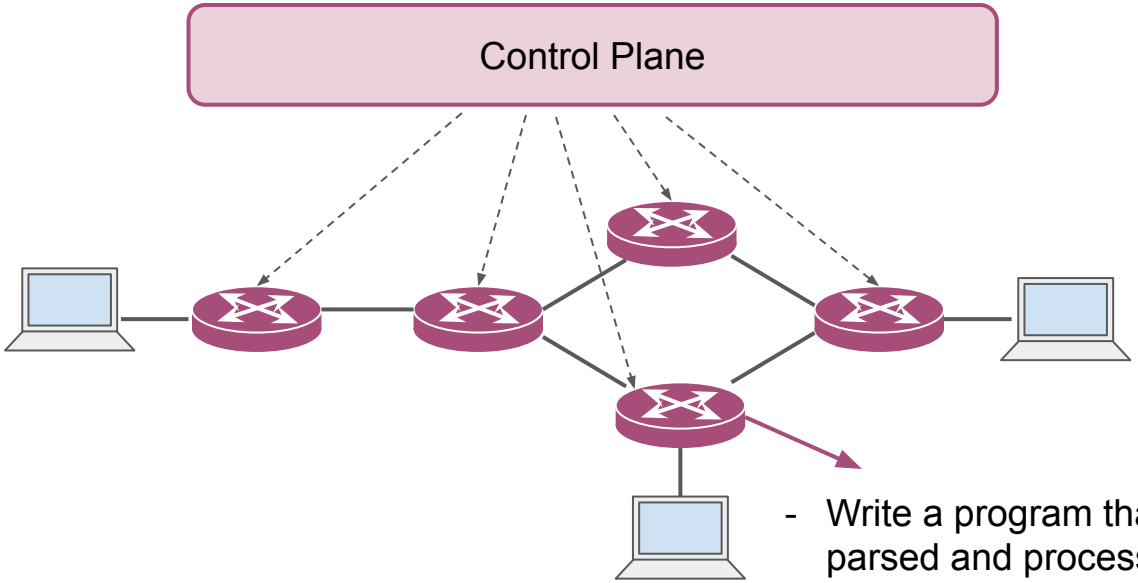
- Write a program that decides the forwarding paths.
- Have a runtime compute and communicate proper configurations to network devices.



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

Here are some examples…

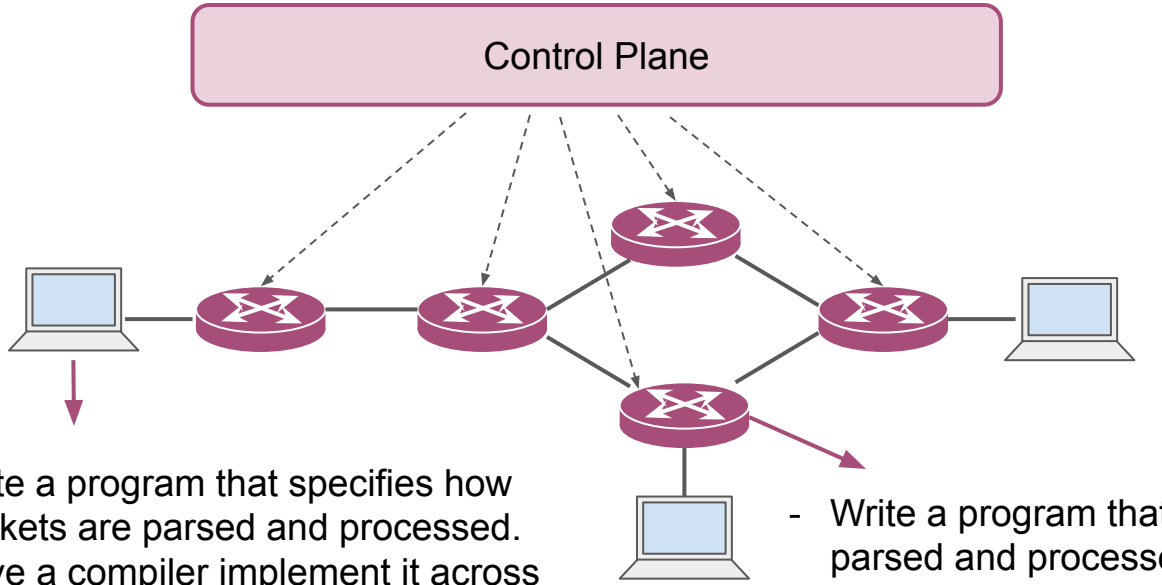
- Write a program that decides the forwarding paths.
- Have a runtime compute and communicate proper configurations to network devices.



- 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.

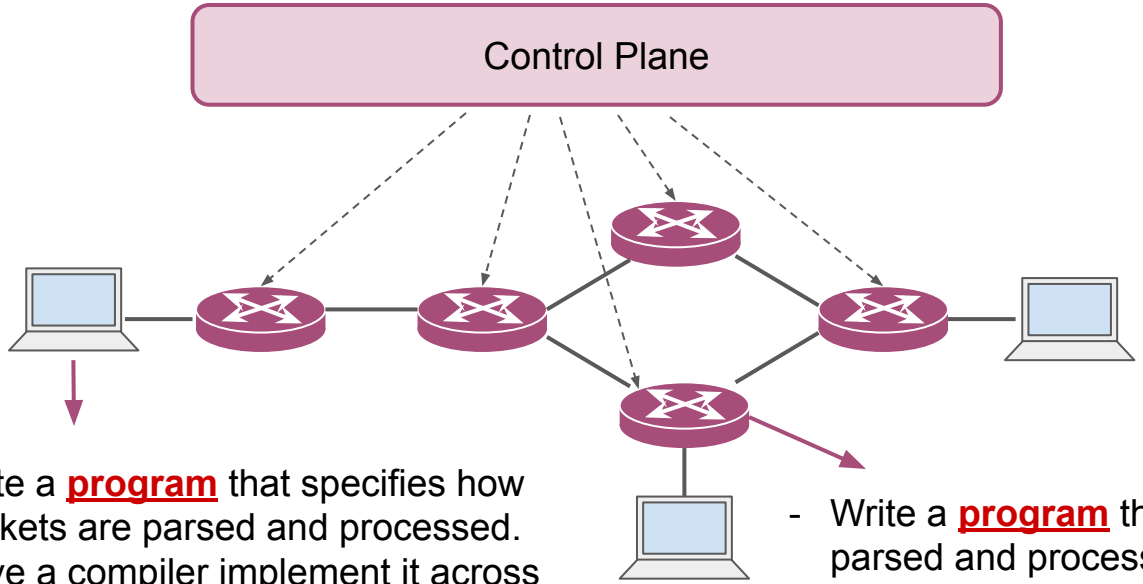


- Write a program that specifies how packets are parsed and processed.
- Have a compiler implement it across user-space, the Kernel, and hardware accelerators.

- 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.



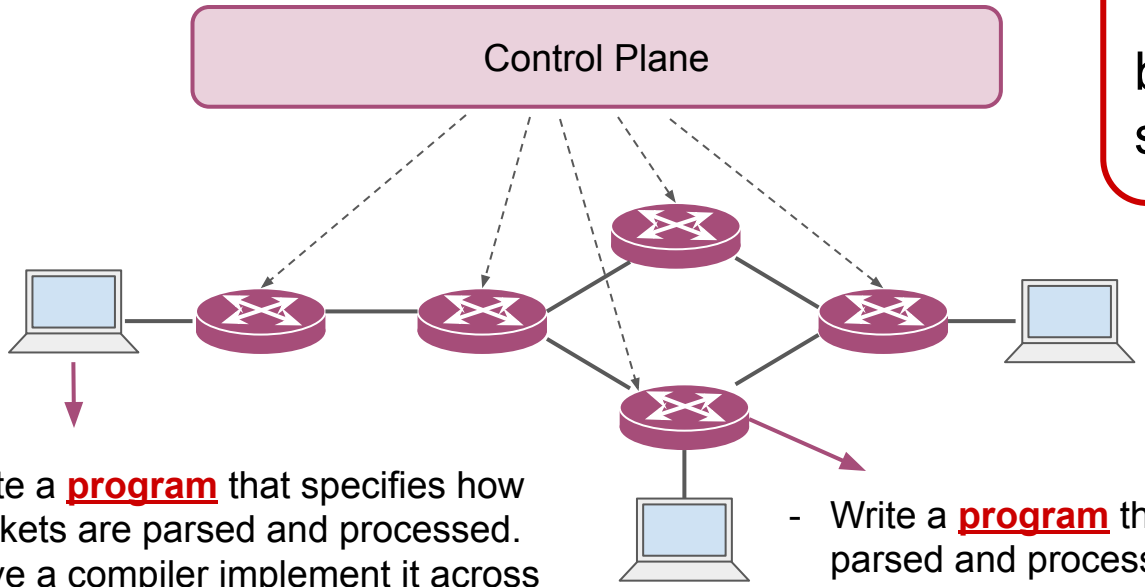
- Write a **program** that specifies how packets are parsed and processed.
- Have a compiler implement it across user-space, the Kernel, and hardware accelerators.

- 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



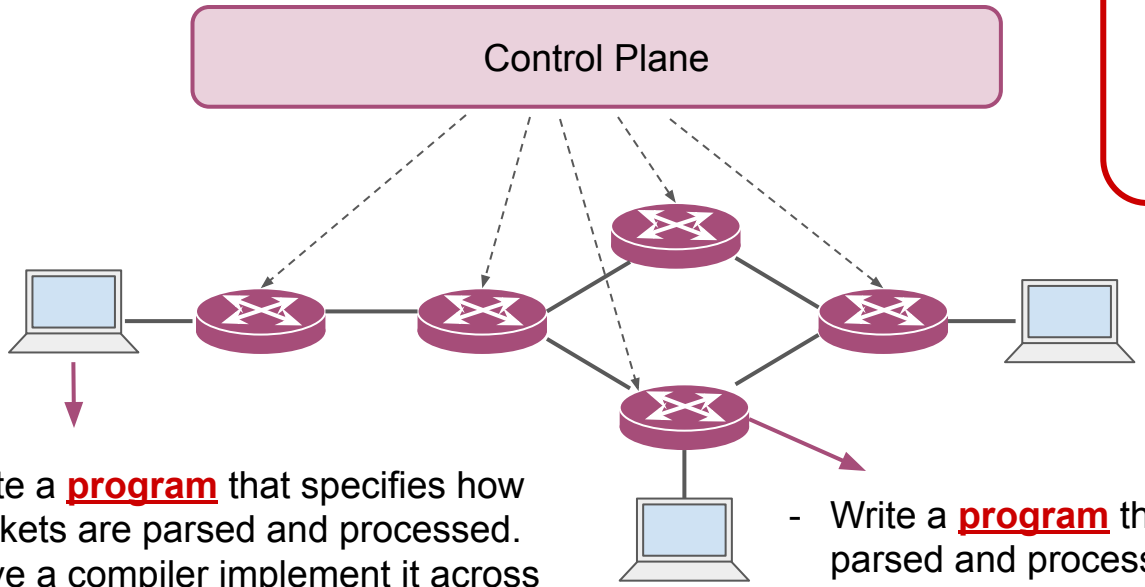
- Write a **program** that specifies how packets are parsed and processed.
- Have a compiler implement it across user-space, the Kernel, and hardware accelerators.

- 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
- Through working on course projects!

Logistics

- Class is Mondays and Wednesday, 11:30pm to 12:50pm.
- 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.
- Grades will be announced through LEARN.

Course Structure

- Reviews? Presentations? Project? Not quite.
- We are doing things a bit differently this term.
- We will learn through working on research projects together
- How would that work?

Course Structure

- The next 1-2 lectures, we will discuss a number of research areas within programmable networks.
- Each student will pick a topic for their course projects
  - We can discuss project ideas during lectures too
- For the first half (i.e., until reading week), each student will learn and teach others about the state-of-the-art in their topic
  - I will happily provide pointers
- For the second half, we will work towards a solution
  - You will think of ideas, we will discuss them in the class
- There is a 6-page final report due at the end of the term (April 24)

How do presentations work?

- <u>One week before</u> your presentation, send me your presentation plan
  - i.e., the paper(s) you want to present, the solution idea you want to discuss
- Be prepared to lead a discussion for ~40 minutes
  - If you are presenting papers, make sure you have read them thoroughly and can navigate it to answer the questions that come up during discussions.
  - If you are presenting a solution idea, make sure you put sufficient thought into it, come up with pros and cons, etc.
- You'll present once every other week.

How do presentations work?

- Don't plan a very long presentation.
- Plan a 10-15 minute overview but be ready with back-up slides or the papers' texts themselves to lead a discussion and answer questions.
- Most importantly, remember that the purpose is for us to learn together.
- So, approach your presentation with the goal of teaching others something you have studied in depth.

How do I find related papers (for the first half)?

- Conference proceedings
  - SIGCOMM, NSDI are our go-to conferences
  - Depending on the topic, OSDI, SOSP, ASPLOS, SIGMETRICS, and others could be relevant too.
  - If you need help/pointers, don't hesitate to reach out.
- References "Chasing"
  - Backwards: look at the related work section of a paper, find related citations
  - Forwards: Use academic search engines like Google Scholar to find relevant papers that have cited a specific paper.
- Ask :)
  - I'll be more than happy to provide some initial pointers.

Changes compared to last time

- No paper reviews
- No written proposal or progress reports
  - Your presentations will basically serve that purpose
- No assignments
- Presentations and in-class discussions are weighted more heavily.
- No changes to the final project report, but hopefully it is easier to write after all the discussions we will have throughout the term.

Grading

- Presentations: 35%
- Participation in discussions: 15%
- Final project report: 50%

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.