UNIVERSITY OF WATERLOO

CS 856: Programmable Networks

Lecture 8: Flexible and Fine-Grained Network Monitoring

Mina Tahmasbi Arashloo

Winter 2023

Logistics

- Project progress report is due **Friday, March 10th**
- Reviews due **Monday, March 13th, at 5pm.**

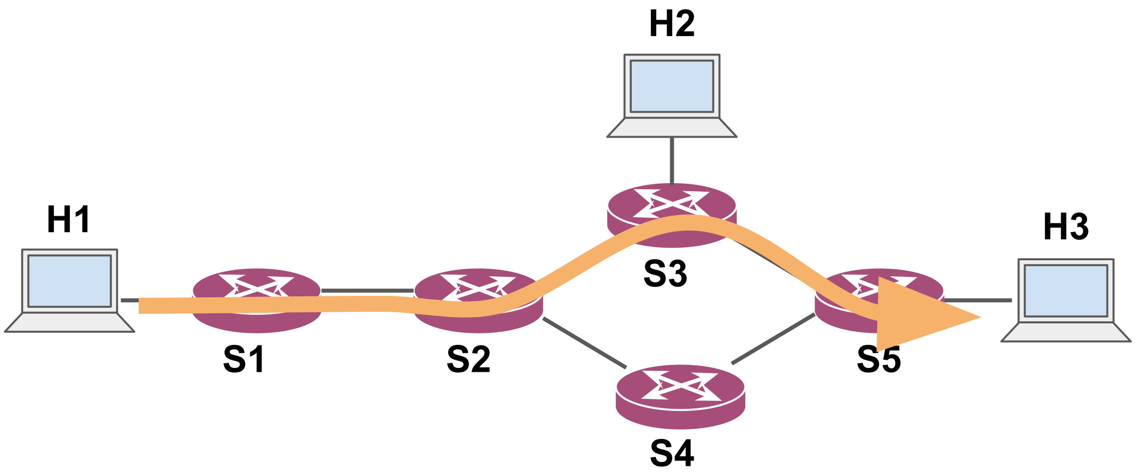
What is network monitoring?

- Understanding what is happening in the network at run-time and in real-time
    - Which links in the network are congested?
    - Which flows are contributing the most to the congestion at link L?
    - Are there flows that experience tail latency larger than X?
    - What are the K largest flows in the network?
    - Are there any unusual traffic patterns to/from certain IP addresses?
    - …

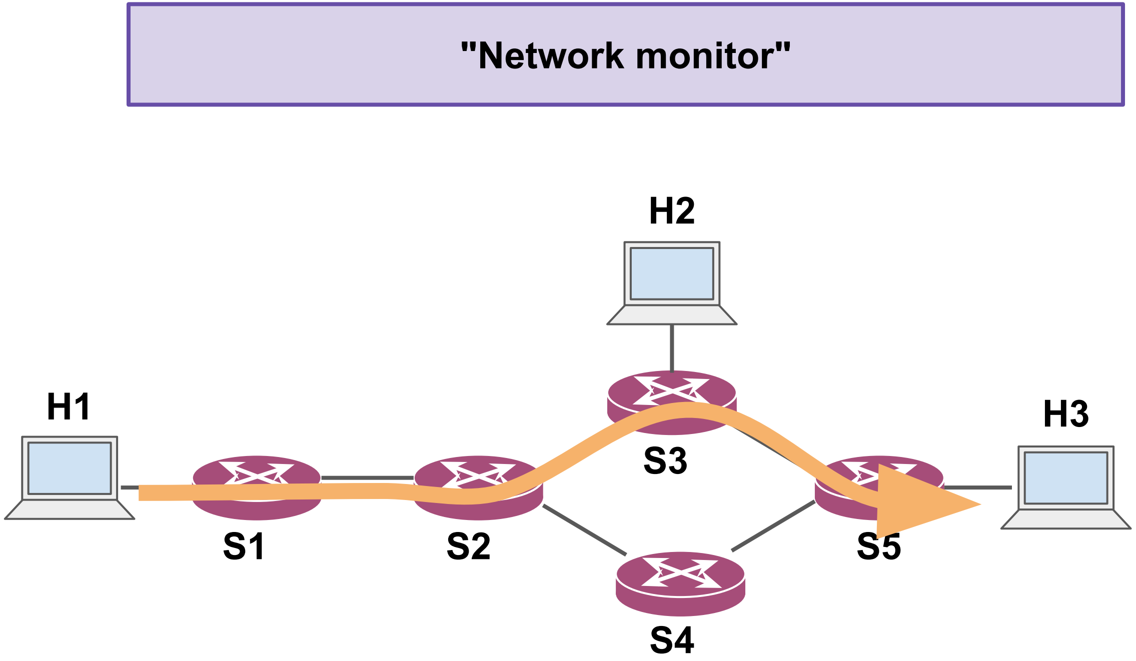
Why is it important?

- Networks are quite dynamic!
    - Traffic patterns change, links and devices fail, …
- More often than not, we need to dynamically re-consider how to network should process traffic in response to changes at run-time.

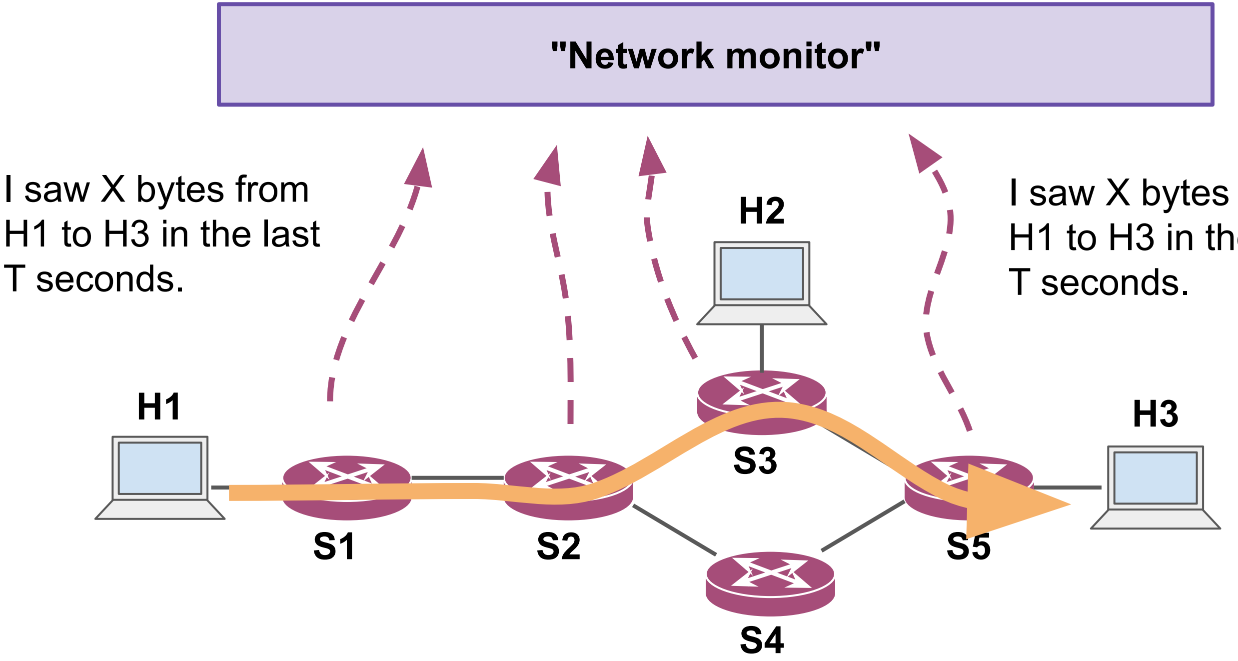
A simple example



A simple example

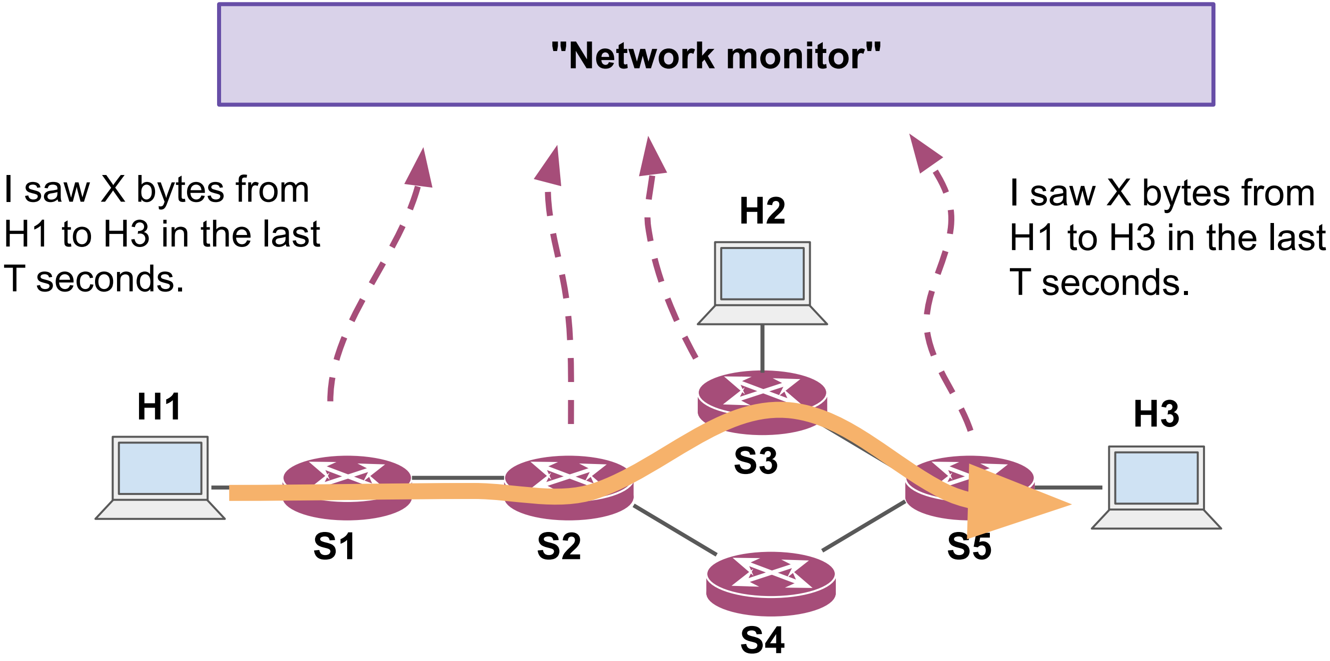


A simple example

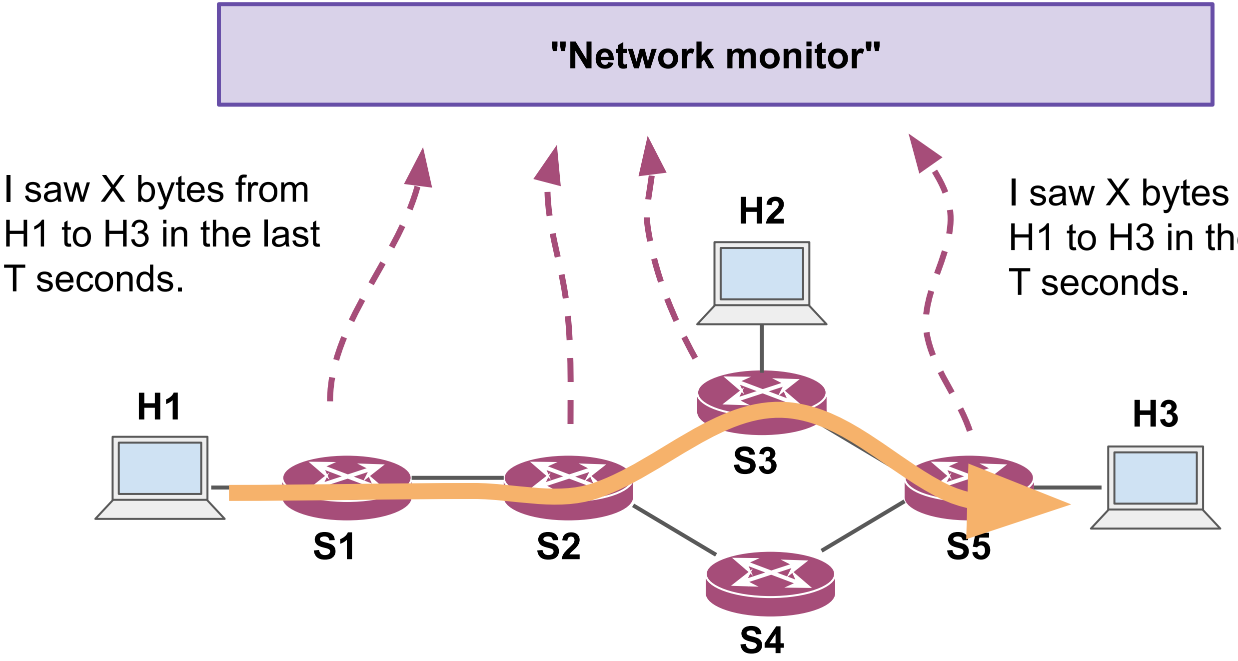


A simple example

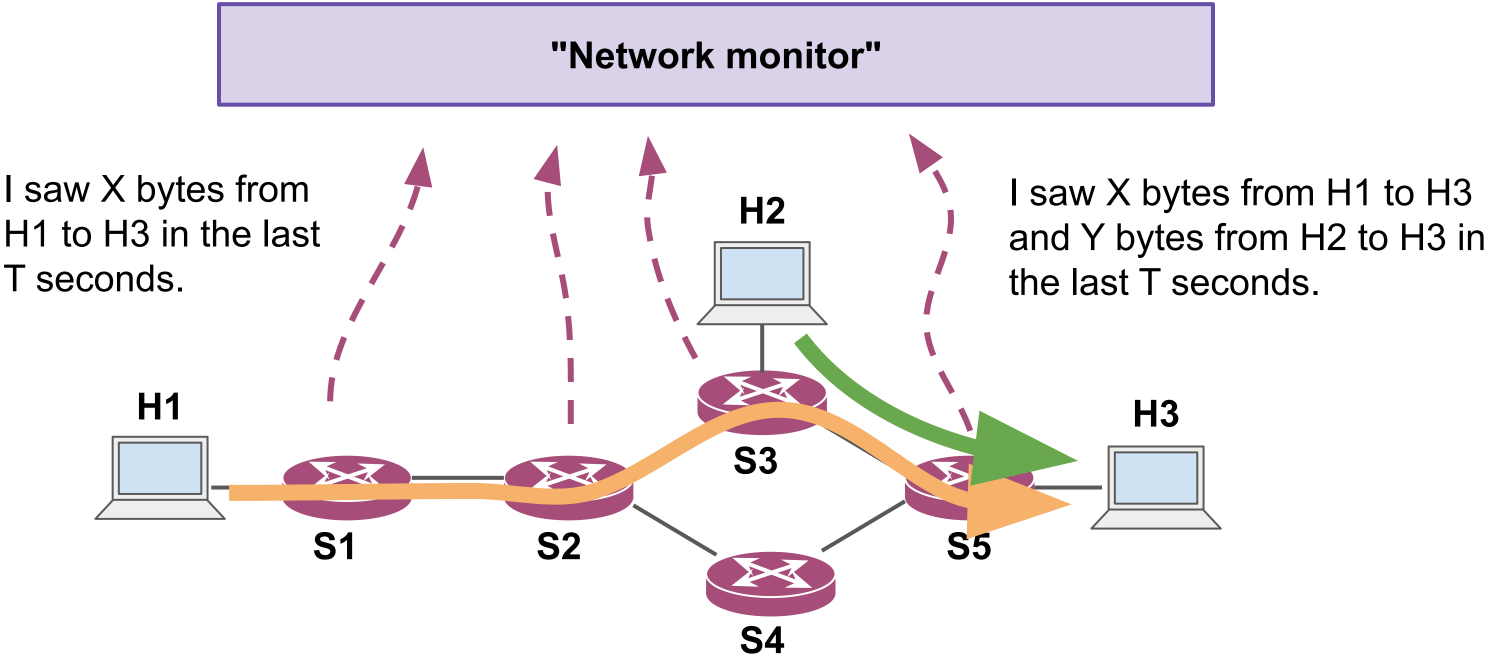
- The monitor can *pull* the monitoring data from the device
- or the device can *push* it to the monitor.



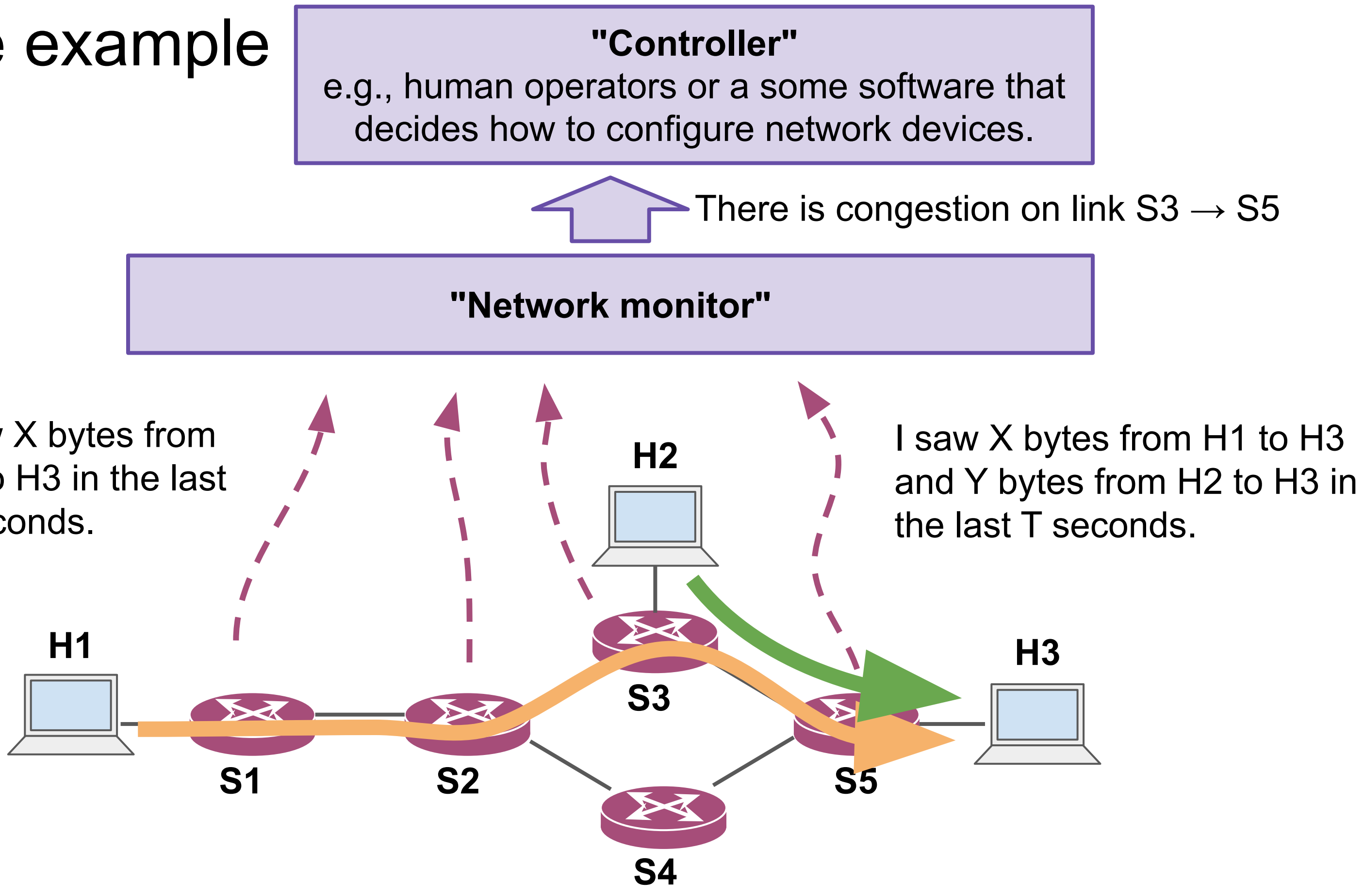
A simple example



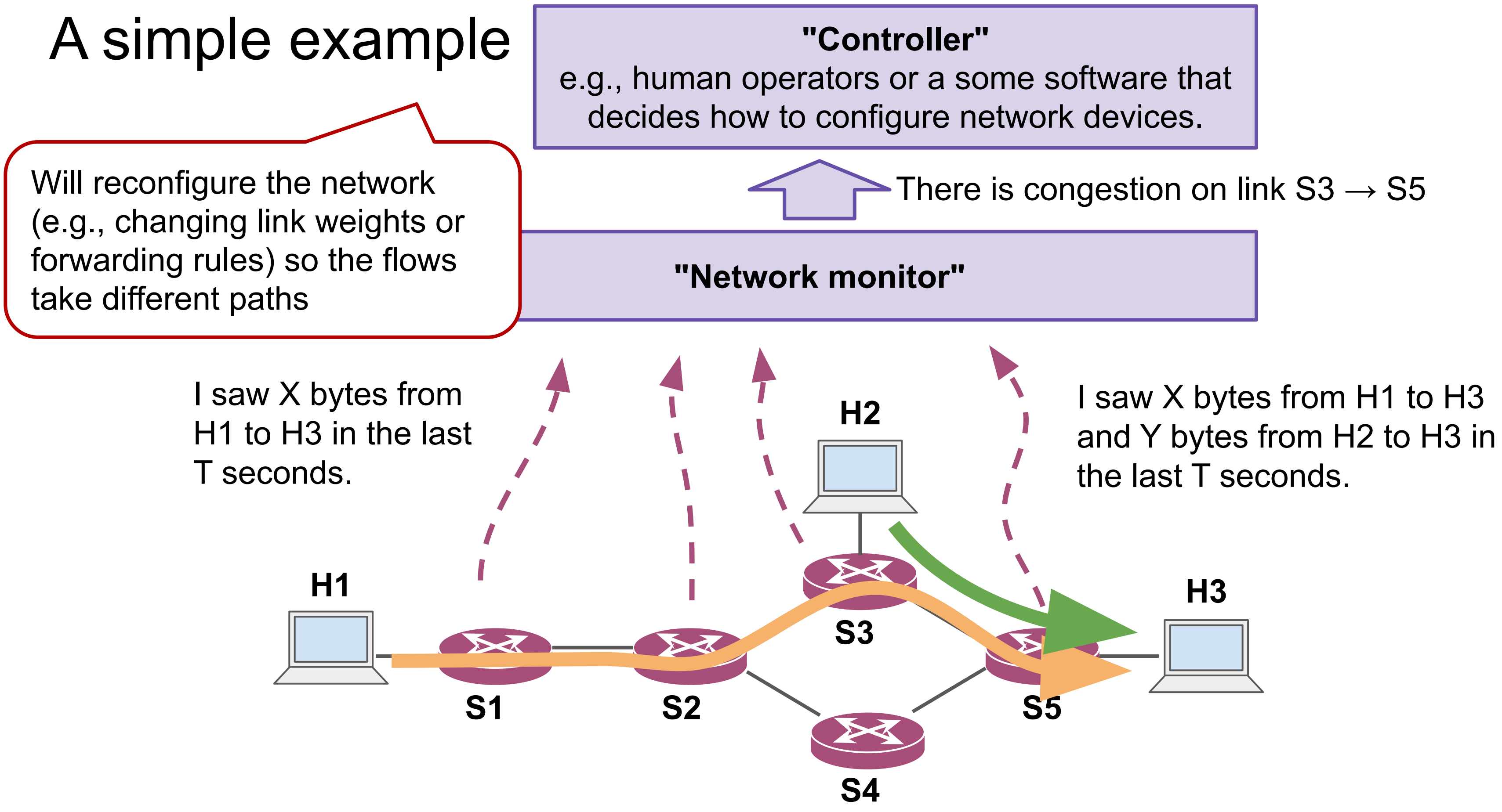
A simple example



A simple example

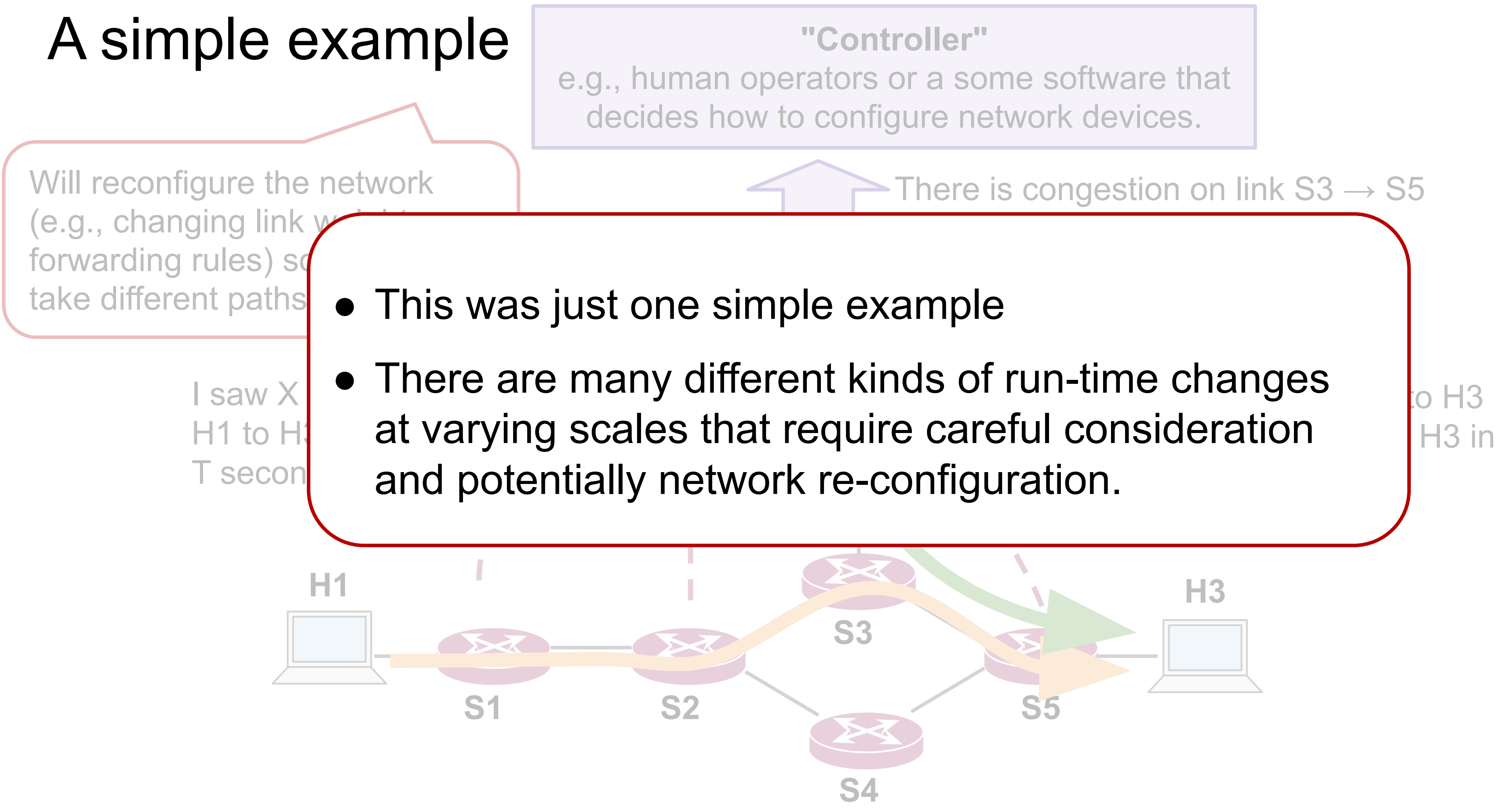


A simple example



A simple example

- This was just one simple example
- There are many different kinds of run-time changes at varying scales that require careful consideration and potentially network re-configuration.



Why is it important?

- Networks are quite dynamic!
    - Traffic patterns change, links and devices fail, …
- More often than not, we need to dynamically re-consider how to network should process traffic in response to changes at run-time.
- An accurate understanding of the state of the network is crucial for making good decisions :)

Why is it important?

- Networks are quite dynamic!
    - Traffic patterns change, links and devices fail, …
- More often than not, we need to dynamically re-consider how to network should process traffic in response to changes at run-time.
- An accurate understanding of the state of the network is crucial for making good decisions :)

obtained through network monitoring

Why is it challenging?

We have to observe network traffic and analyze it in real-time.

Why is it challenging?

We have to observe network traffic and analyze it in real-time.

- Terabits of traffic per second on a single switch
- 100s or 1000s of switches in a network

Why is it challenging?

- Many different statistics and properties to monitor
- What you need to monitor can change over time.

We have to observe network traffic and analyze it in real-time.

- Terabits of traffic per second on a single switch
- 100s or 1000s of switches in a network

Why is it challenging?

- Many different statistics and properties to monitor
- What you need to monitor can change over time.

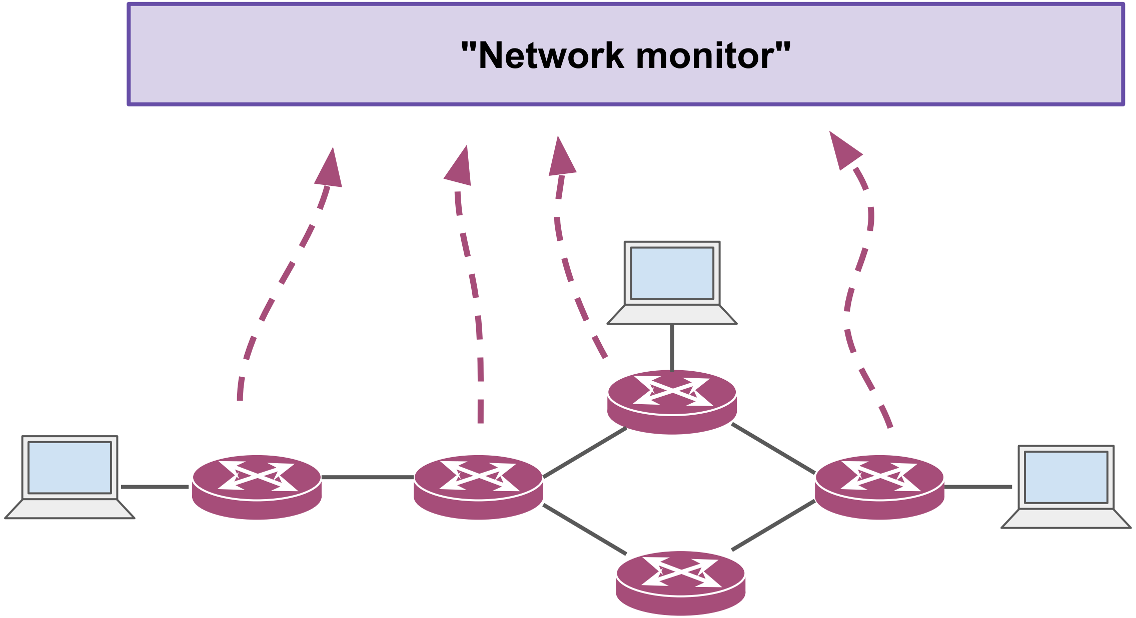
We have to observe network traffic and analyze it in real-time.

- Terabits of traffic per second on a single switch
- 100s or 1000s of switches in a network

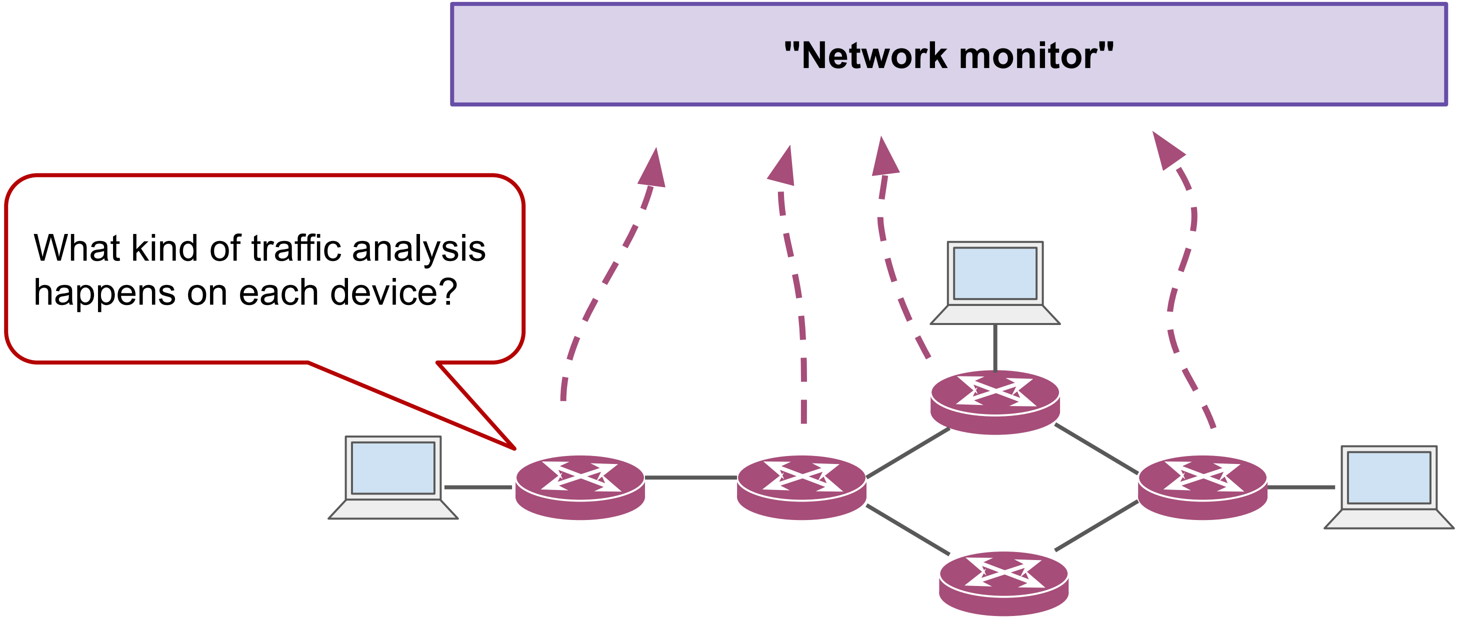
Have to observe and analyze traffic quite fast

- Data plane: fast but has limited resources.
- Control plane: more resources but slower.

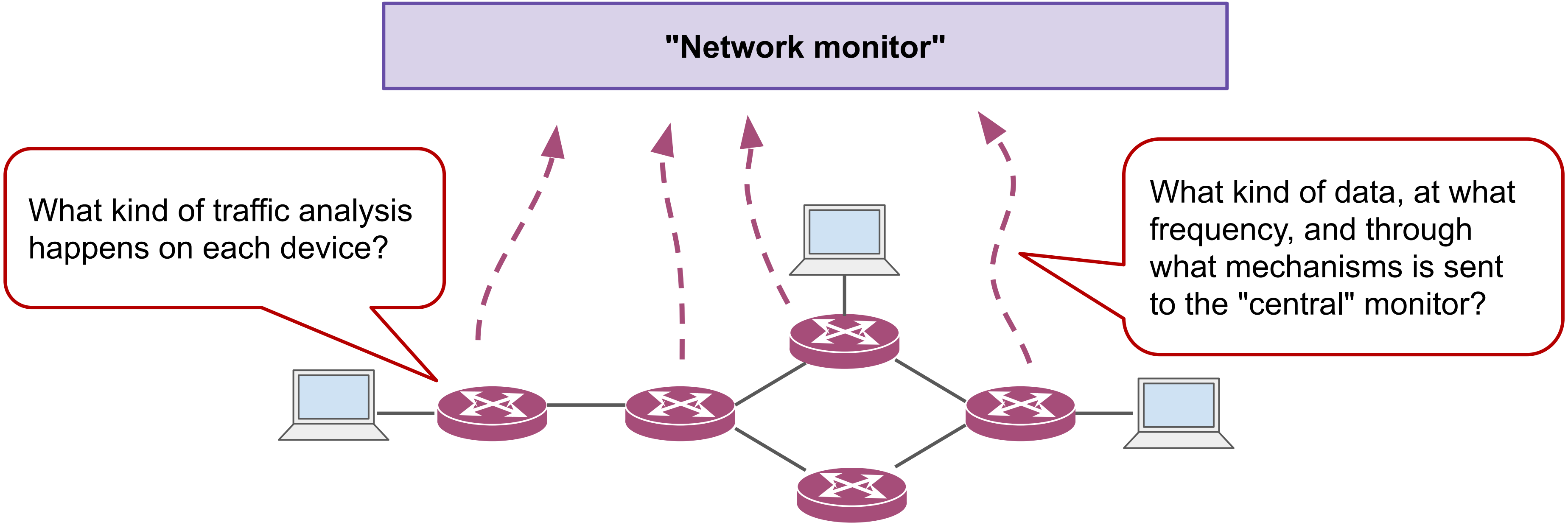
The design space



The design space



The design space



A single "query": K largest flows

- Suppose you want to know which K flows are the largest in your network.
  - largest flow → has sent the largest amount of traffic
  - known as heavy-hitters.
- There are a number of ways you can go about doing that…

A single "query": K largest flows

- You can send the flow identifier and size of each packet to the central network monitor
  - The monitor can count #bytes in each flow, sort the flows, and find the top K
  - Significant communication overhead, but very accurate results

A single "query": K largest flows

- You can count how many bytes of each flow you see in the switch and send a report of that to the network monitor every X seconds.
  - The monitor can merge the information to find flow sizes and the K largest flows.
  - Lower communication overhead, accuracy can depend on reporting frequency and how in-sync the reports are.

A single "query": K largest flows

- You can keep track of only the heavy hitters (as opposed to every flow) on each device and have the network monitor pull the info when needed.
    - What is the communication overhead?
    - How accurate are the results? hint: packets of the same flow may traverse different devices in the network.

Multiple queries

- Do you collect some generic statistics from each device at the network monitor and use that to answer as many queries as you can?
  - What information do you collect? flow sizes? queue sizes?
  - How granular? per-packet? per-flow?
  - How much information do you aggregate on each device and over how long of a time window before analyzing it at the monitor?
- Or, do you tailor what you measure to the kind of queries you are asked?

Monitoring in "traditional" networks

- Not a lot of flexibility to try different points in the design space
- It is up to the vendors what kind of monitoring data they collect on the switches and how it can be reported to a monitoring server.
  - Typically limited to coarse-grained information every few seconds.
  - e.g., NetFlow
- Sounds familiar? :)

Network programmability → Flexible and fine-grained monitoring

- Program the data plane to gather the data that you want
- Program the data plane (and the run-time) to have the data pushed to/pulled form a central monitor when you want.
- Create top-down programmable monitoring frameworks:
    - Users specify the information they are interested as queries
    - The compiler and runtime figure out how to configure each device to collect and report information according to the query.

Network programmability → Flexible and fine-grained monitoring

- Program the data plane to gather the data that you want
- Program ... ed to/pulled ...
- Create to...
  - Users...
  - The compiler and runtime figure out how to configure each device to collect and report information according to the query.

Monitoring is one of the "killer" apps for programmable data planes

Network programmability → Flexible and fine-grained monitoring

- Program the data plane to gather the data that you want
- Program the data plane (and the run-time) to have the data pushed to/pulled form a central monitor when you want.
- Create top-down programmable monitoring frameworks:
  - Users specify the information they are interested as queries
  - The compiler and runtime figure out how to configure each device to collect and report information according to the query.

Network programmability → Flexible and fine-grained monitoring

- Program the data plane to gather the data that you want
- Program the data plane (and the run-time) … e the data pushed to/pulled form a central monitor when …
- Create top-down programmable moni…
  - Users specify the information they are interested as queries
  - The compiler and runtime figure out how to configure each device to collect and report information according to the query.

- Sketches
- In-Band Network Telemetry (INT)
- …

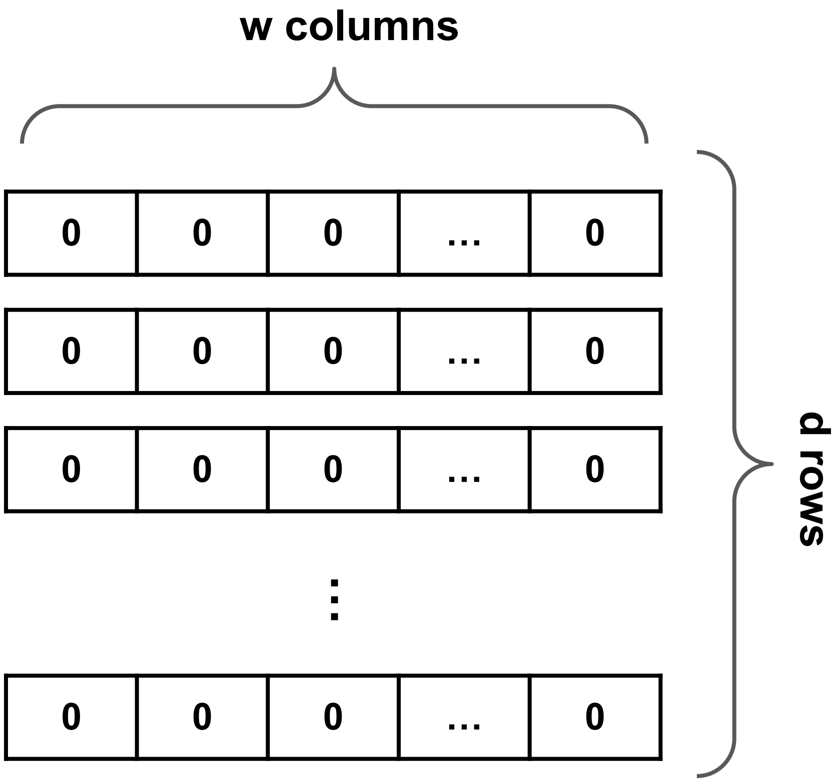
Sketches

- Modern high-speed switches can observe terabits of traffic every second.
- and have limited computational resources
  - Specially memory, which is essential for monitoring purposes, e.g., to keep track of statistics
- Typically, if we have N flows going through a switch, we don't have O(N) memory in the switch to keep information about them.
- So, what do we do?

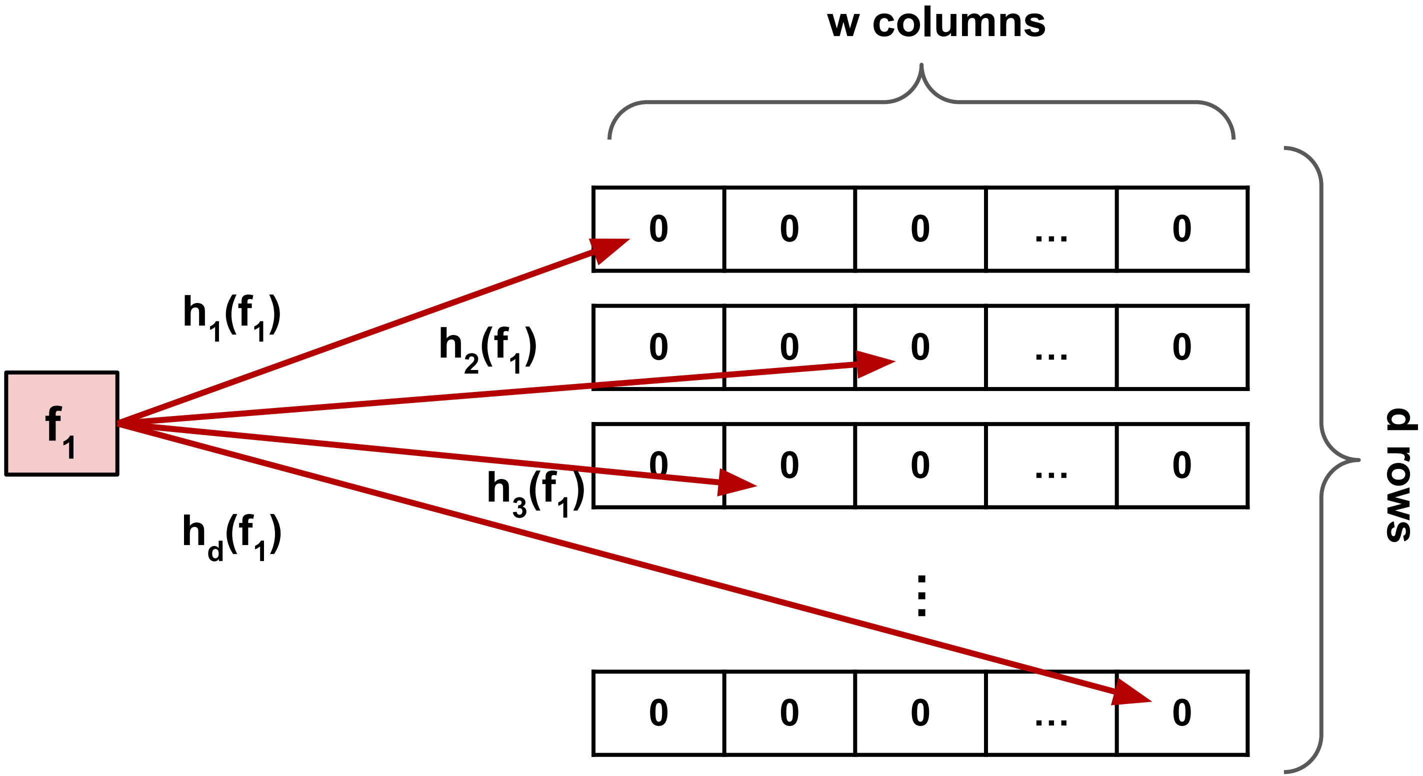
Sketches

- Sketches are approximate data structures that
    - keep information about a large amount of data in a substantially smaller amount of space
    - and can answer certain queries about it in an approximate way.
- They typically provide a trade-off between resource usage and accuracy.
- If you give them more space, they'll provide a more accurate result.

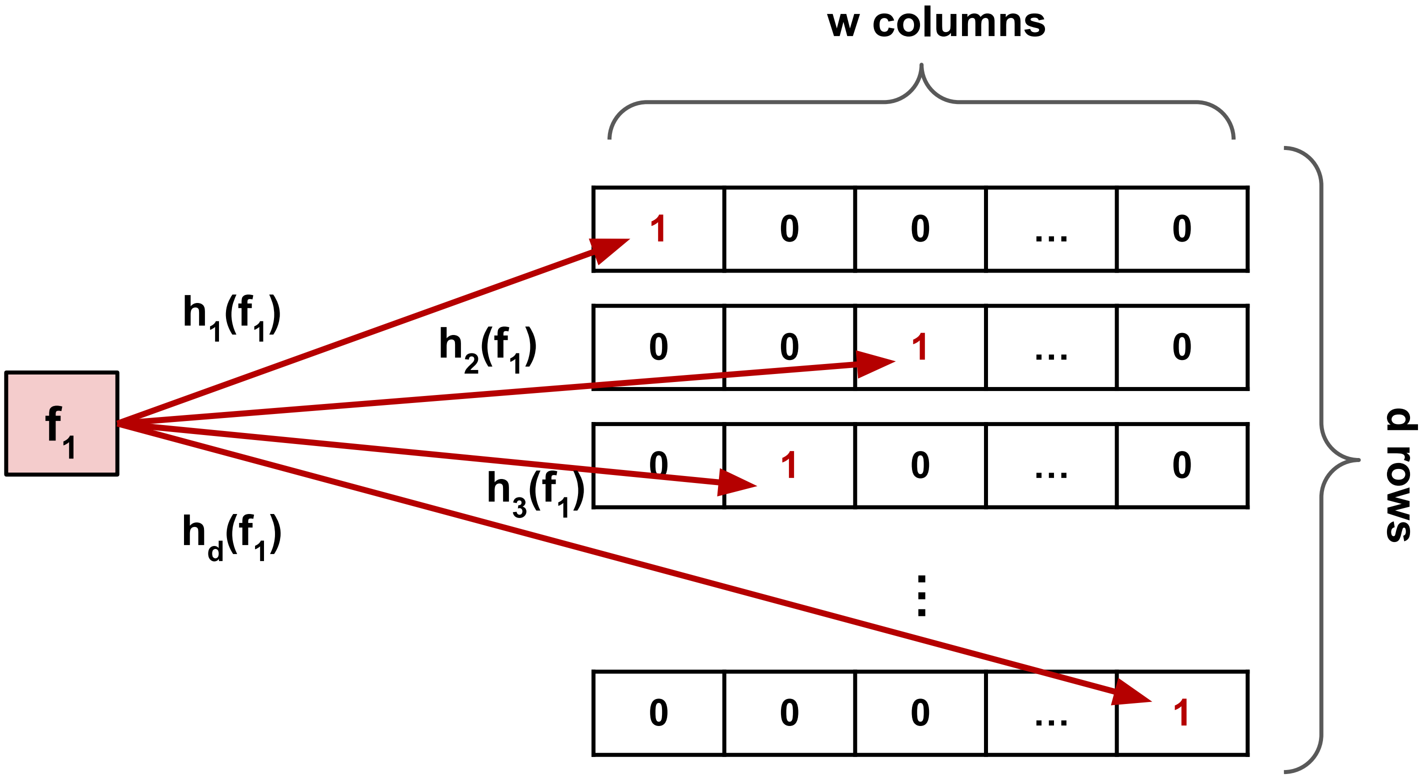
Example: Count-Min Sketch



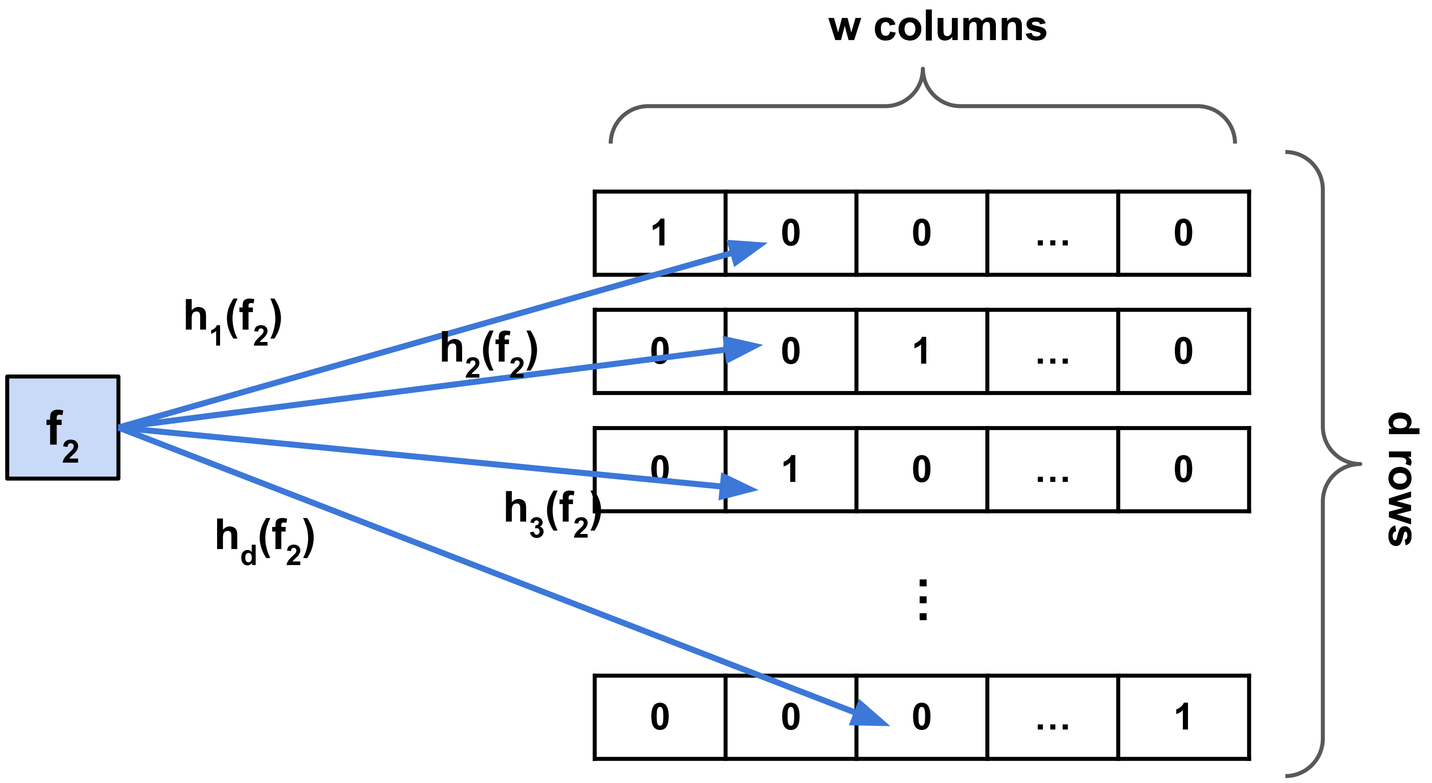
Example: Count-Min Sketch



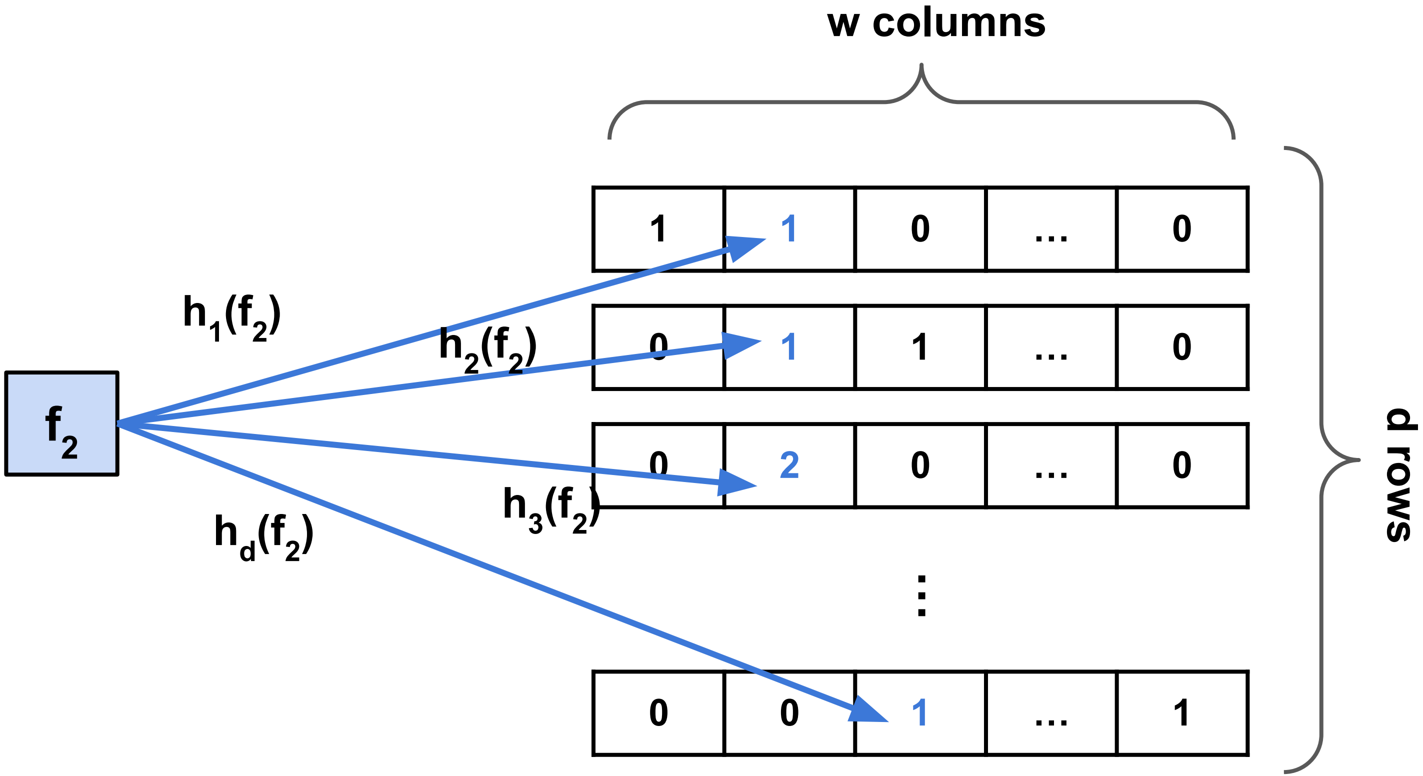
Example: Count-Min Sketch



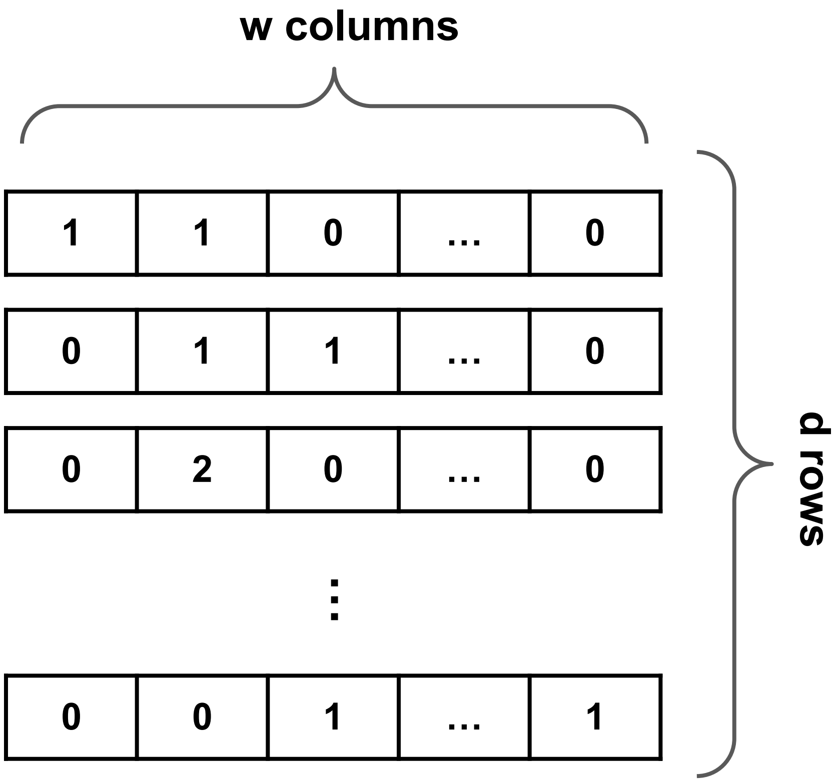
Example: Count-Min Sketch



Example: Count-Min Sketch



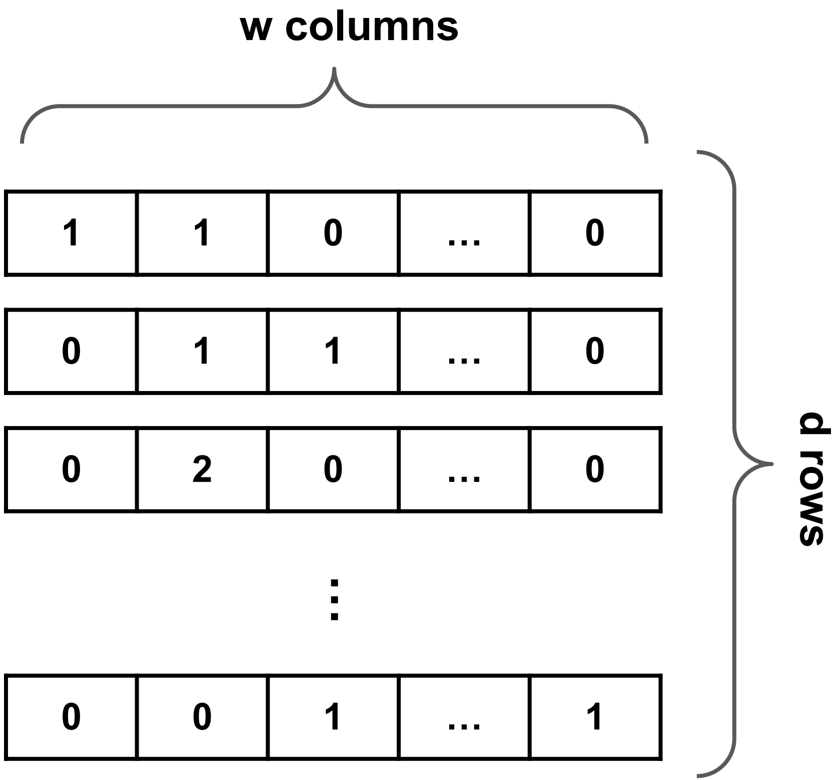
Example: Count-Min Sketch



Example: Count-Min Sketch

How many f_1 packets did you see?

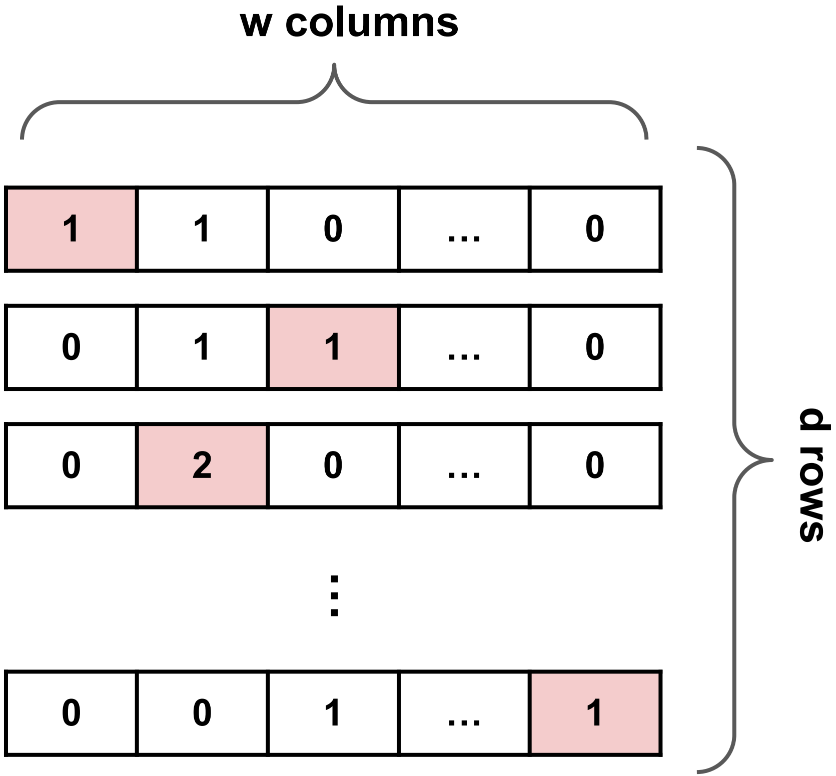
- Get the entries at $h_1(f_1)$, $h_2(f_1)$, …, $h_d(f_1)$



Example: Count-Min Sketch

How many f_1 packets did you see?

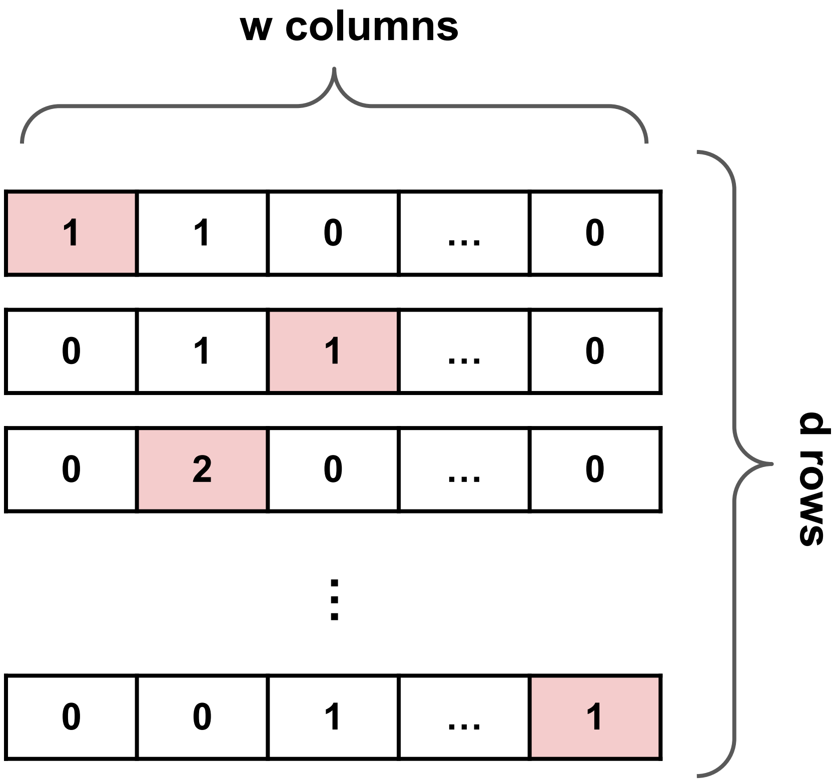
- Get the entries at $h_1(f_1)$, $h_2(f_1)$, …, $h_d(f_1)$



Example: Count-Min Sketch

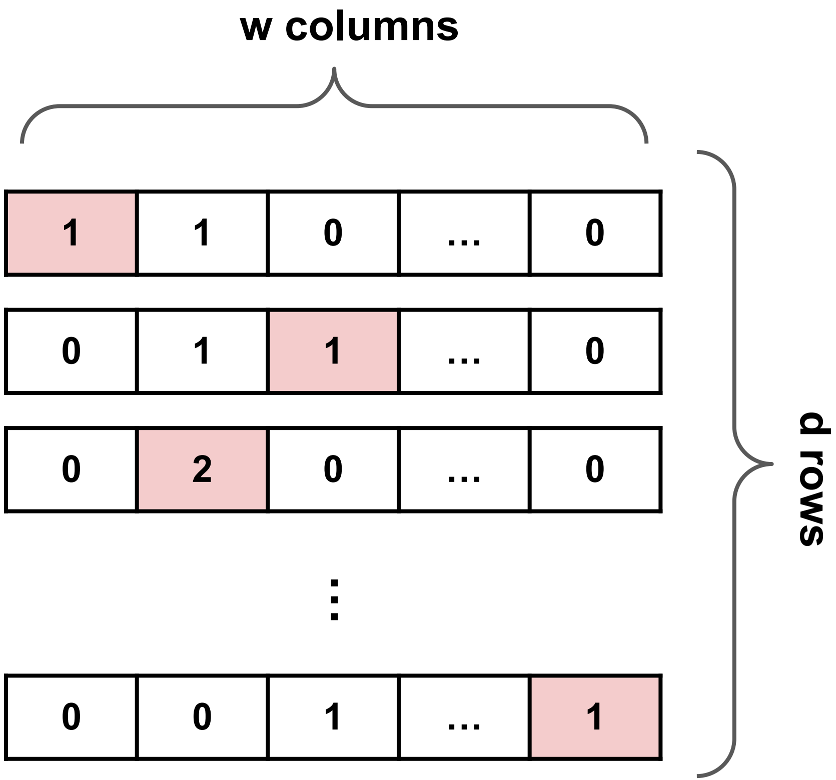
How many f_1 packets did you see?

- Get the entries at $h_1(f_1)$, $h_2(f_1)$, …, $h_d(f_1)$
- Return the minimum
  - i.e., 1



Example: Count-Min Sketch

- Approximation comes from hash collisions
- The larger w and d, the more accurate the result.



Connections to streaming algorithms

- There are lots of synergies between network monitoring and streaming algorithms.
- Algorithms in the streaming setting have more constraint than "regular" algorithms
    - They see the input as a sequence of items examined in a few passes, typically one → packets passing through the switch
    - Operate within limited memory (sublinear) and sometimes limited processing per item → computational constraints of programmable switches

In-Band Network Telemetry (INT)

- In programmable data planes, you can define custom headers and process them however you want.
- INT proposes to add a "telemetry" header and have switches populate it with information that will help network monitoring
  - How long did the packet spend in the switch? How much of it was waiting in a queue?
  - switch id, to help figure out what paths packets take in the network
  - …

In-Band Network Telemetry (INT)

- Once the packet gets to its destination, the information in the INT header can be analyzed there and/or sent to a central monitor.
- Having fine-grained information about what happened to the packet as it traverses a network is extremely useful.
- Specially for detecting and debugging transient and subtle problems.
- There is no free lunch!
    - If every switch adds a large INT header to the packet, that can create throughput overheads.

Network programmability → Flexible and fine-grained monitoring

- Program the data plane to gather the data that you want
- Program the data plane (and the run-time) to have the data pushed to/pulled form a central monitor when you want.
- Create top-down programmable monitoring frameworks:
  - Users specify the information they are interested as queries
  - The compiler and runtime figure out how to configure each device to collect and report information according to the query.

Network programmability → Flexible and fine-grained monitoring

- Program the data plane to gather the data that you want
- Program the data plane (and the run-time) to have the data pushed to/pulled form a central monitor when you want.
- Create top-down programmable monitoring frameworks:
  - Users specify the information they are interested as queries
  - The compiler and runtime figure out how to configure each device to collect and report information according to the query.

Connections to network verification

- It is likely that we can't model *everything* in the network in detail and analyze it proactively.
- To ensure our networks satisfy our desired properties, we need
  - scalable proactive analysis to catch as many violating scenarios as possible before production
  - flexible and fine-grained run-time monitoring to continuously watch for property violations at runtime.

Paper 1: Sonata: Query-Driven Streaming Network Telemetry

- A programmable monitoring platform inspired by stream processing platforms (e.g., Spark)
- Users specify their queries over streams of packets in a map-reduce-style interface
- The compiler and runtime figure out which parts of the query to run in the data plane and which parts in a stream processor in the control plane

Paper 2: One Sketch to Rule Them All: Rethinking Network Flow Monitoring with UnivMon

- Before UnivMon, sketches proposed for network monitoring were customized to specific monitoring tasks.
- Inspired by work on universal streaming, UnivMon proposes data-plane sketching primitives that are can be used to compute several statistics.

Additional Resources

- Network Telemetry: Towards A Top-Down Approach
  - a white paper on how do design flexible, fine-grained monitoring platforms, also taking advantage of programmable networks
- A few more papers on data-plane monitoring primitives and programmable monitoring platforms