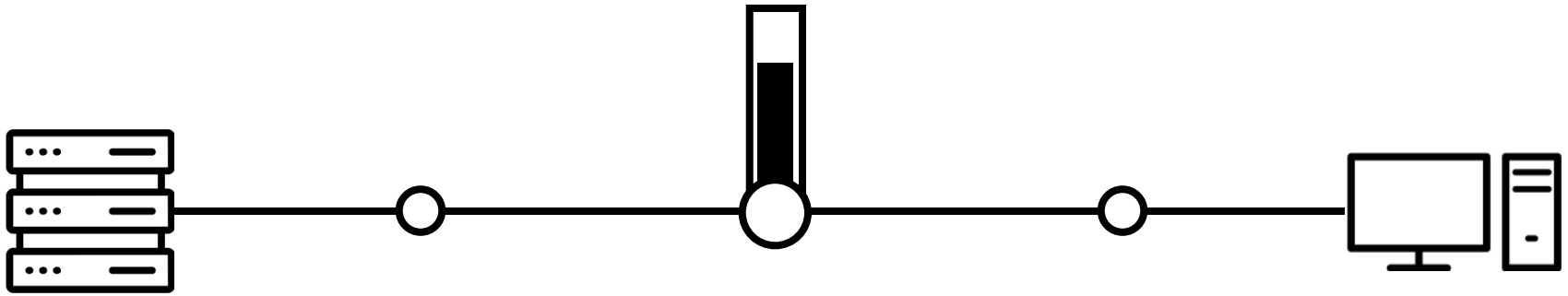
IETF 126 – ICCRG
Vienna, Austria
22nd July 2026

Managing Congestion Control Heterogeneity on the Internet with *Approximate Performance Isolation*

Ayush Mishra[1], Archit Bhatnagar[2], Yixuan Zhang[3], Ben Leong[4], Gao Ya[5], Raj Joshi[6,7]

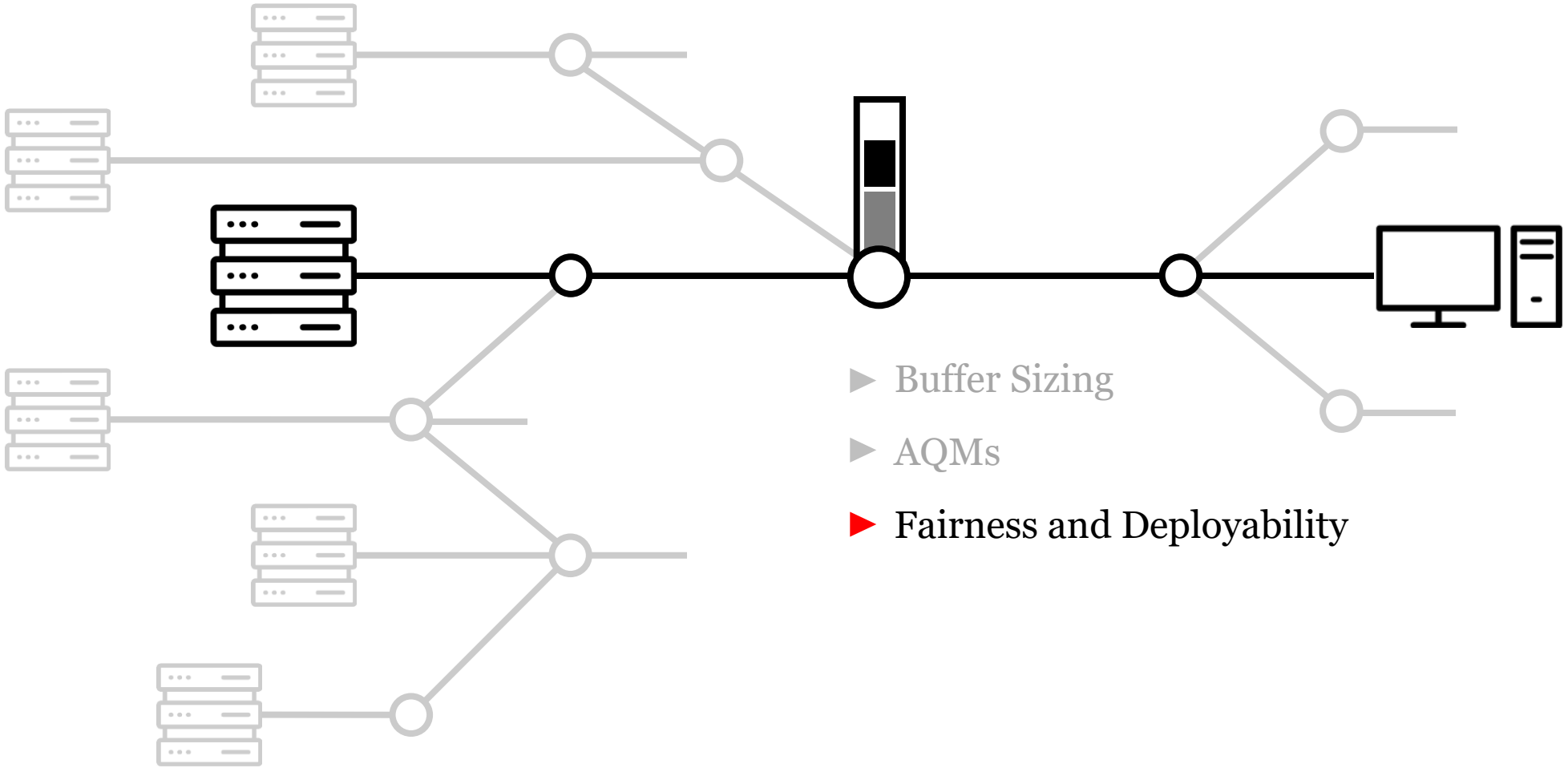
[1]ETH Zurich, [2]University of Michigan, [3]Tsinghua University
[4]National University of Singapore, [5]Wuxi Institute of Technology
[6]Harvard University, [7]Red Hat AI

Congestion Control Algorithms (CCs) are critical to how we run and design our networks



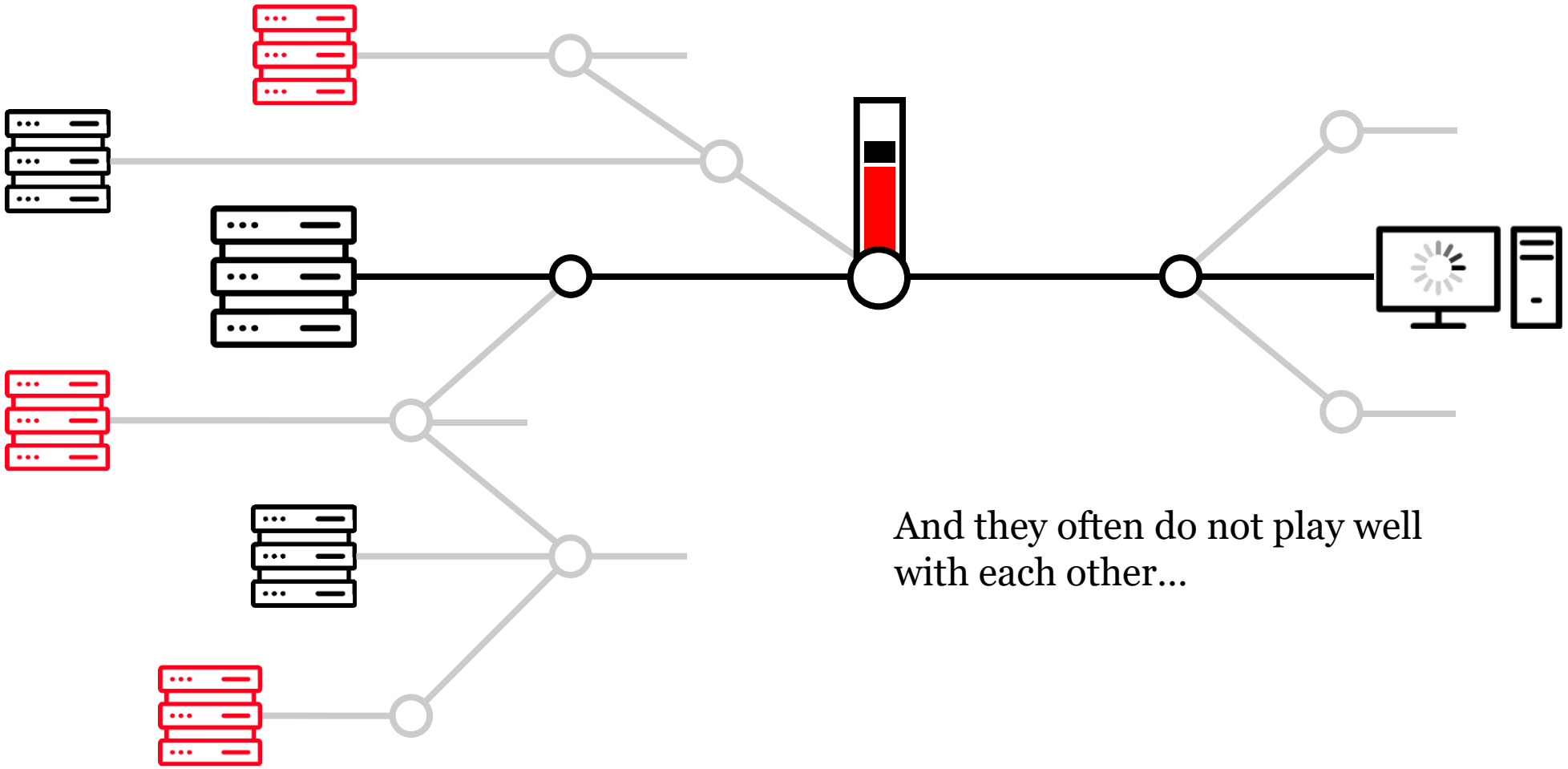
- Buffer Sizing
- AQMs
- Fairness and Deployability

Congestion Control Algorithms (CCs) are critical to how we run and design our networks



- Buffer Sizing
- AQMs
- Fairness and Deployability

The current mix of CCs on the Internet is heterogeneous



And they often do not play well with each other…

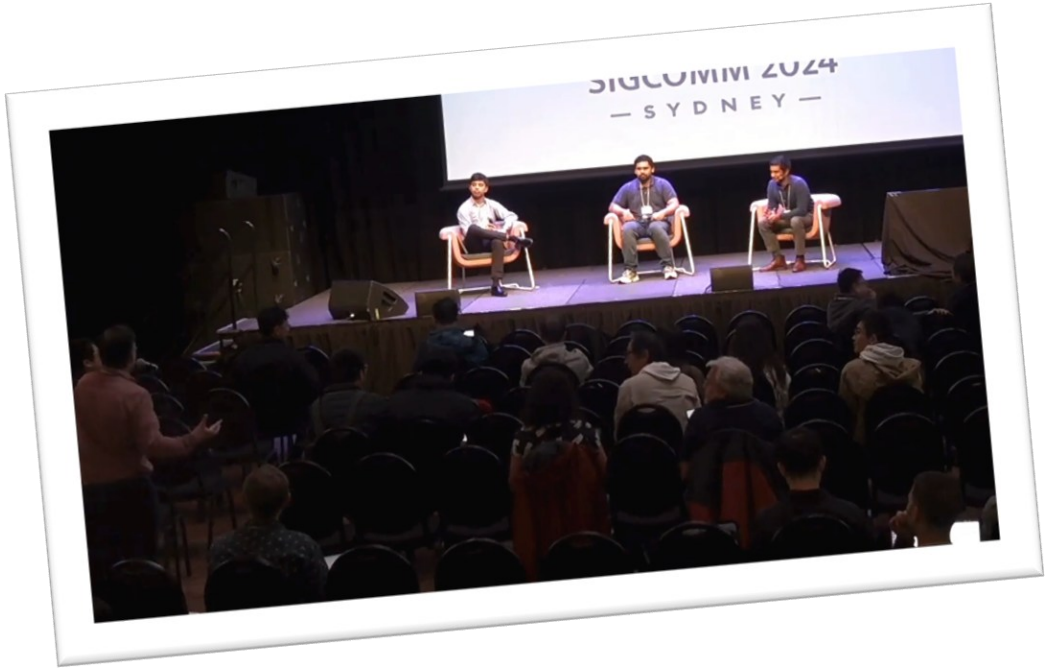
There are significant concerns about fairness

"...observed a single BBR flow consuming a fixed 35-40% of link capacity when competing with as many as 16 CUBIC flows"

Modelling BBR's Interaction with Loss-based Congestion Control
Ware et al.

IMC '19

There are significant concerns about fairness



Community Deployability Standards for Internet Congestion Control

SIGCOMM Debates ‘24

“...observed a single BBR flow consuming a fixed 35-40% of link capacity when competing with as many as 16 CUBIC flows”

Several calls for changing *unfair* CCs

Reviewing the deployability standards on the Internet

Modelling BBR’s Interaction with Loss-based Congestion Control
Ware et al.

IMC ‘19

We have been doing congestion control on the Internet for decades

Why is this suddenly becoming a problem?

Where does this heterogeneity come from?

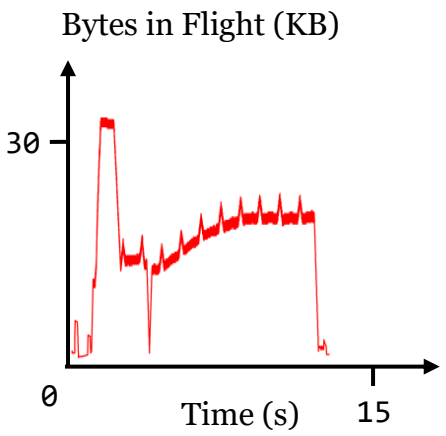
Website	Share	CC
youtube.com, google	13.9%	BBRv3
netflix.com	13.7%	Reno
facebook.com	6.5%	CUBIC
apple.com	4.6%	unknown
amazon.com	4.2%	BBRv1
hulu.com	2.4%	unknown

- Deployment differs between websites

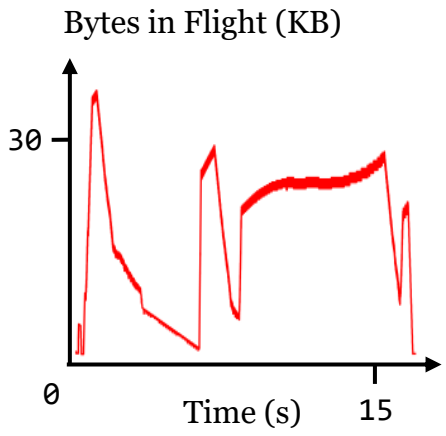
Keeping an Eye on Congestion Control in the Wild with *Nebby*
Mishra et al.

SIGCOMM '24

Where does this heterogeneity come from?



`amazon.com` in Ohio



`amazon.com` in Mumbai

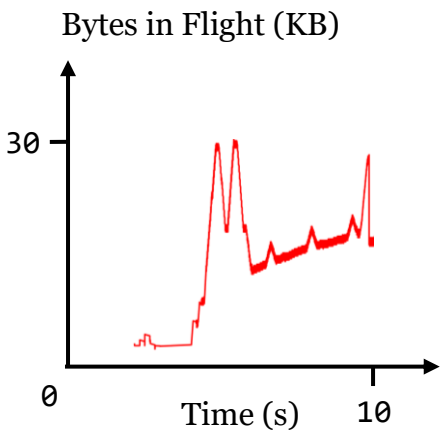
Deployment differs between websites

- ... across geographic regions

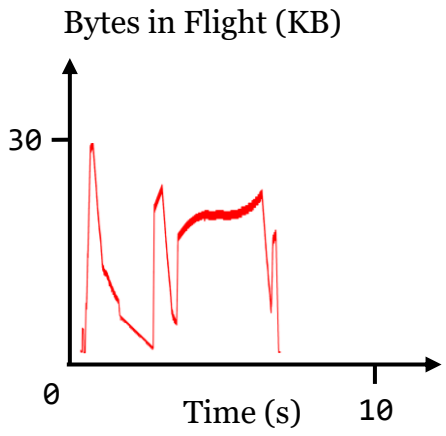
Keeping an Eye on Congestion Control in the Wild with *Nebby*
Mishra et al.

SIGCOMM '24

Where does this heterogeneity come from?



`twitch.com`, video



`twitch.com`, images

Deployment differs between websites

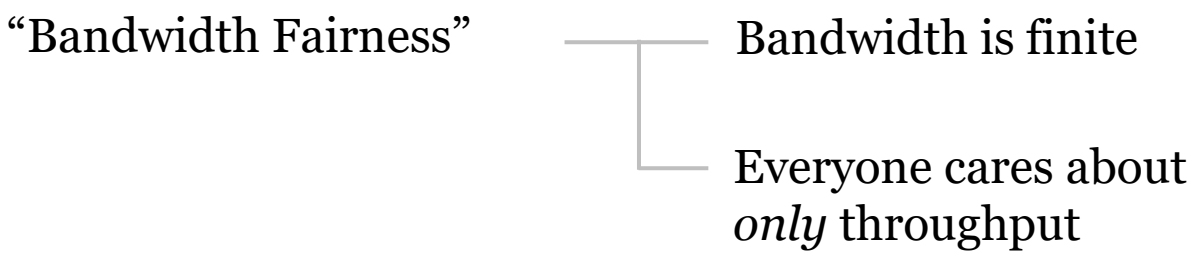
… across geographic regions

- … and differs by asset type

Keeping an Eye on Congestion Control in the Wild with *Nebby*
Mishra et al.

SIGCOMM '24

We need to go beyond bandwidth fairness



We need to go beyond bandwidth fairness

"Bandwidth Fairness"

Bandwidth is finite

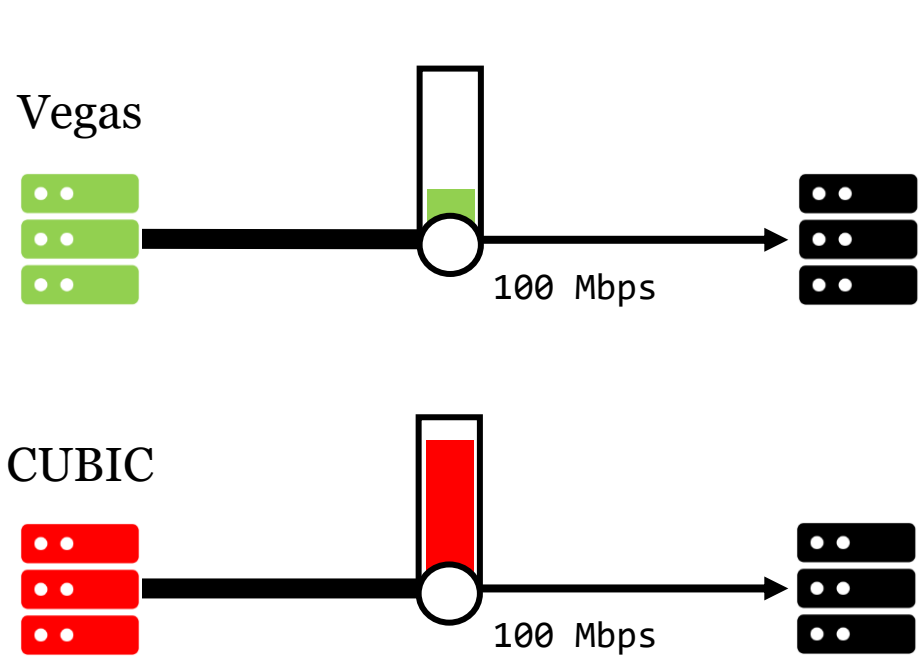
Everyone cares about *only* throughput

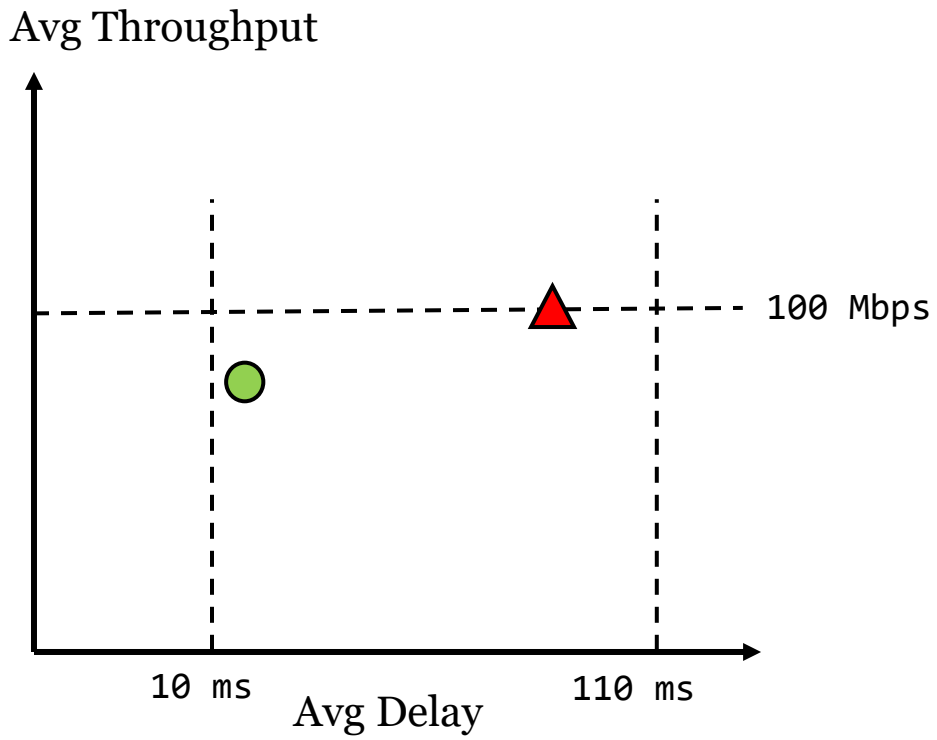
- Why doesn't everyone use the same CC then?

CCs represent an app's desired throughput-delay trade-offs

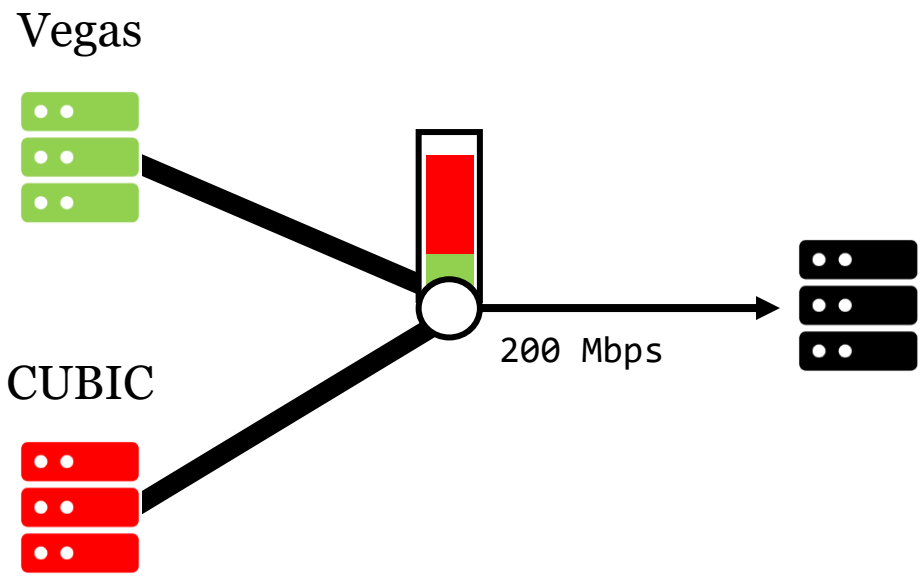
We can see this from how CC heterogeneity manifests on the Internet

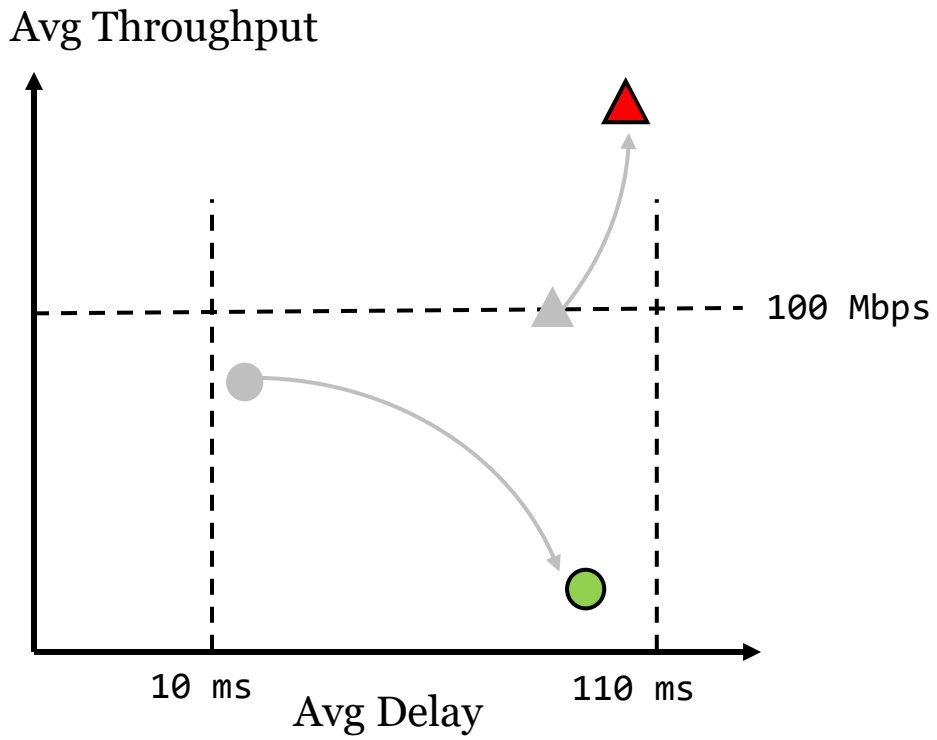
Different CCs make different trade-offs



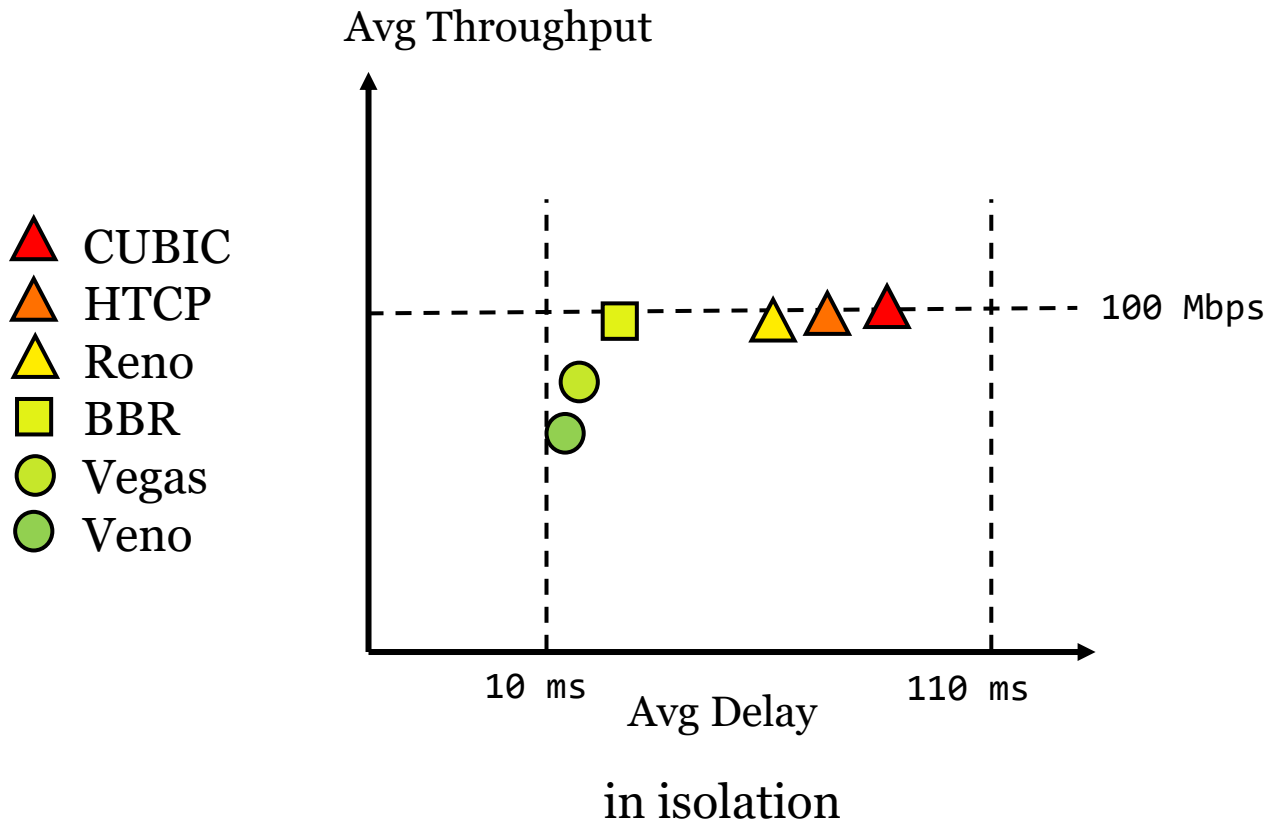


Different CCs make different trade-offs



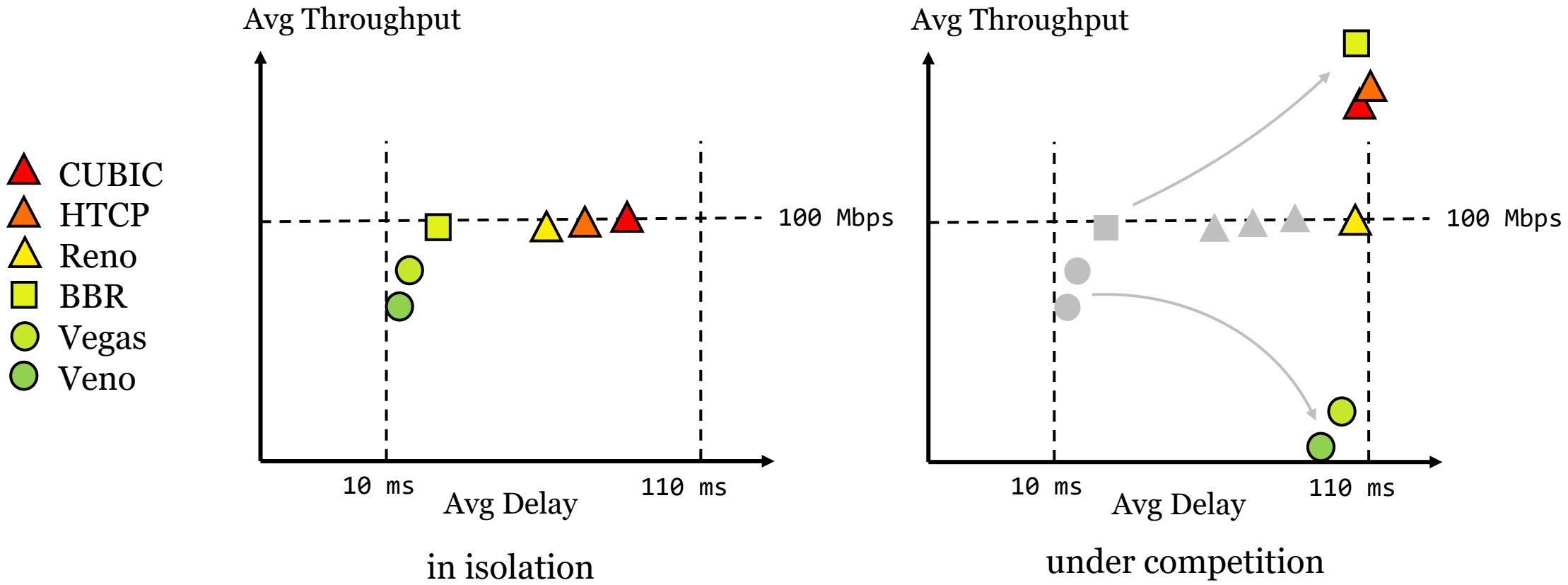


This is true for most CC algorithms

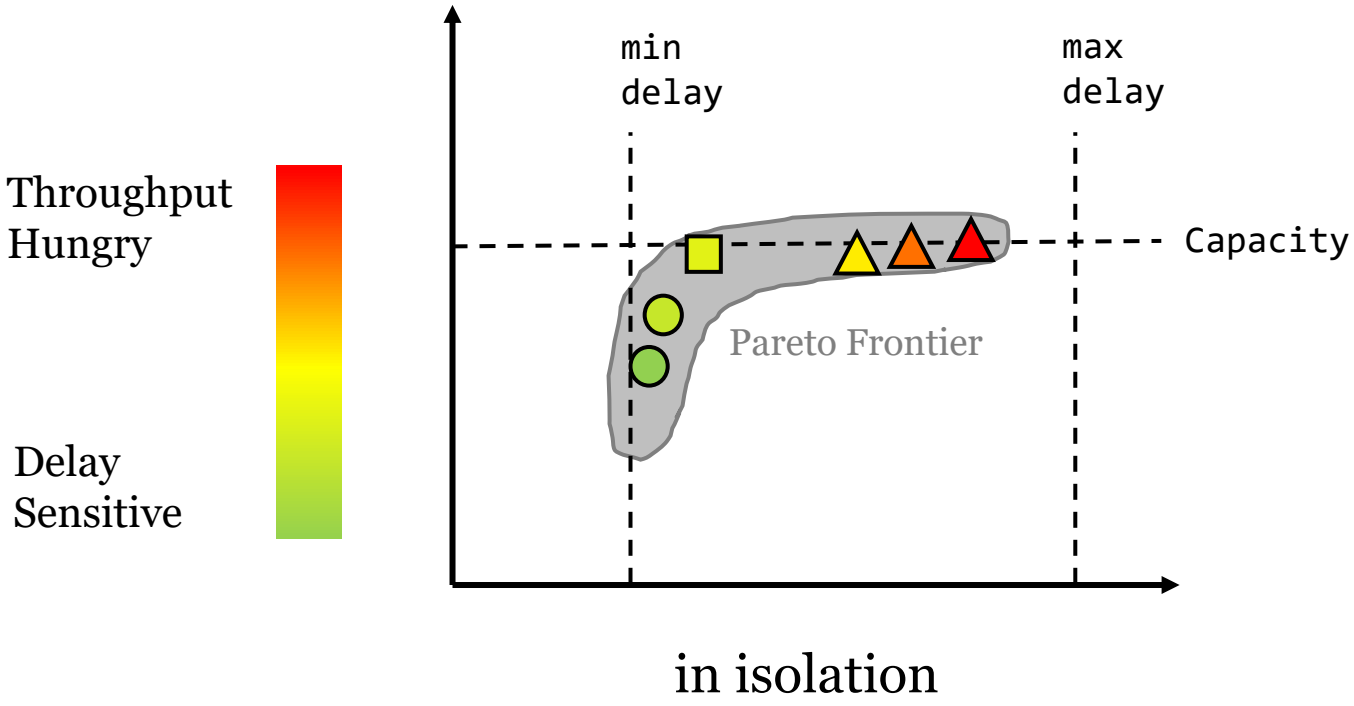


in isolation

This is true for most CC algorithms



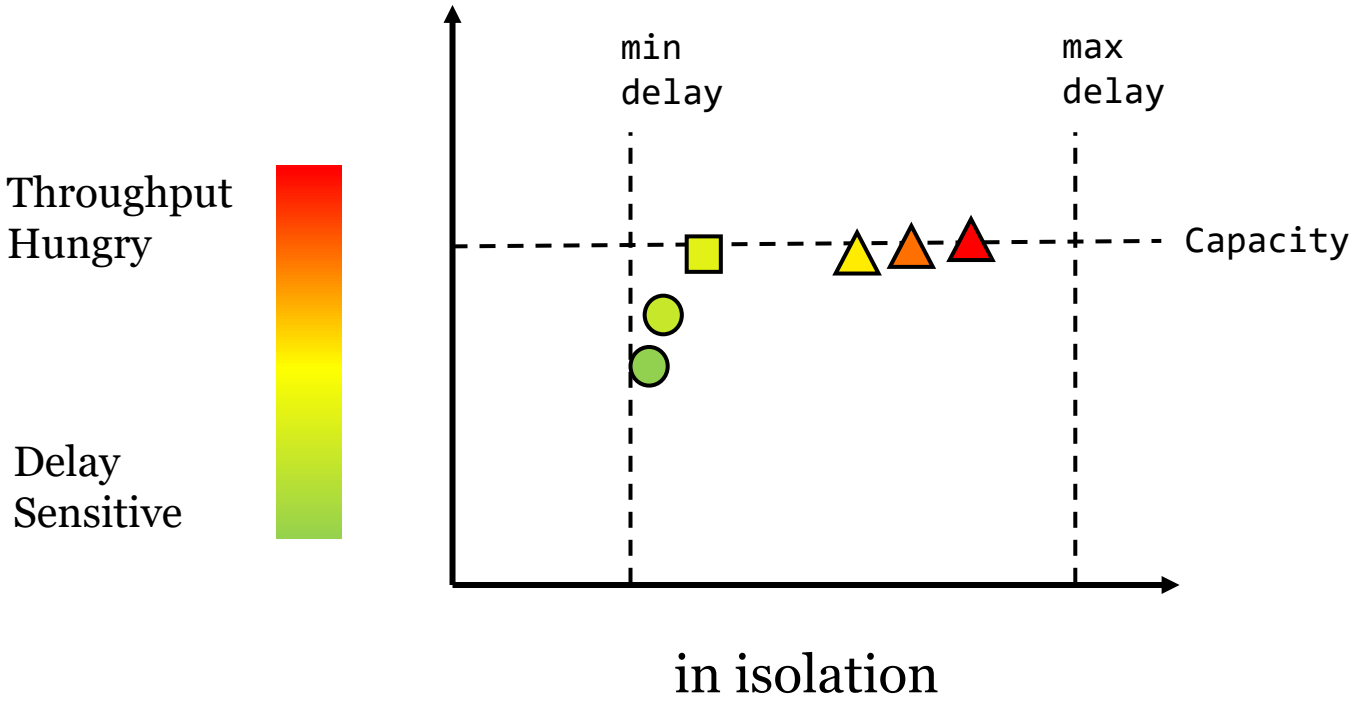
Most CCs organize themselves on a Pareto Frontier



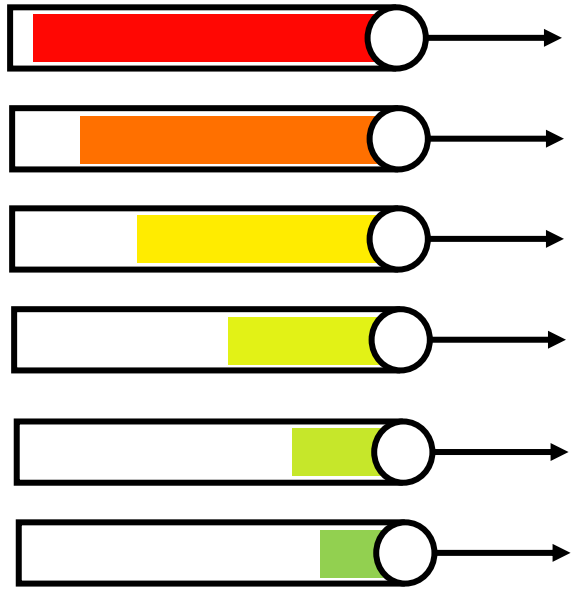
in isolation

How do we create isolation between them when they compete?

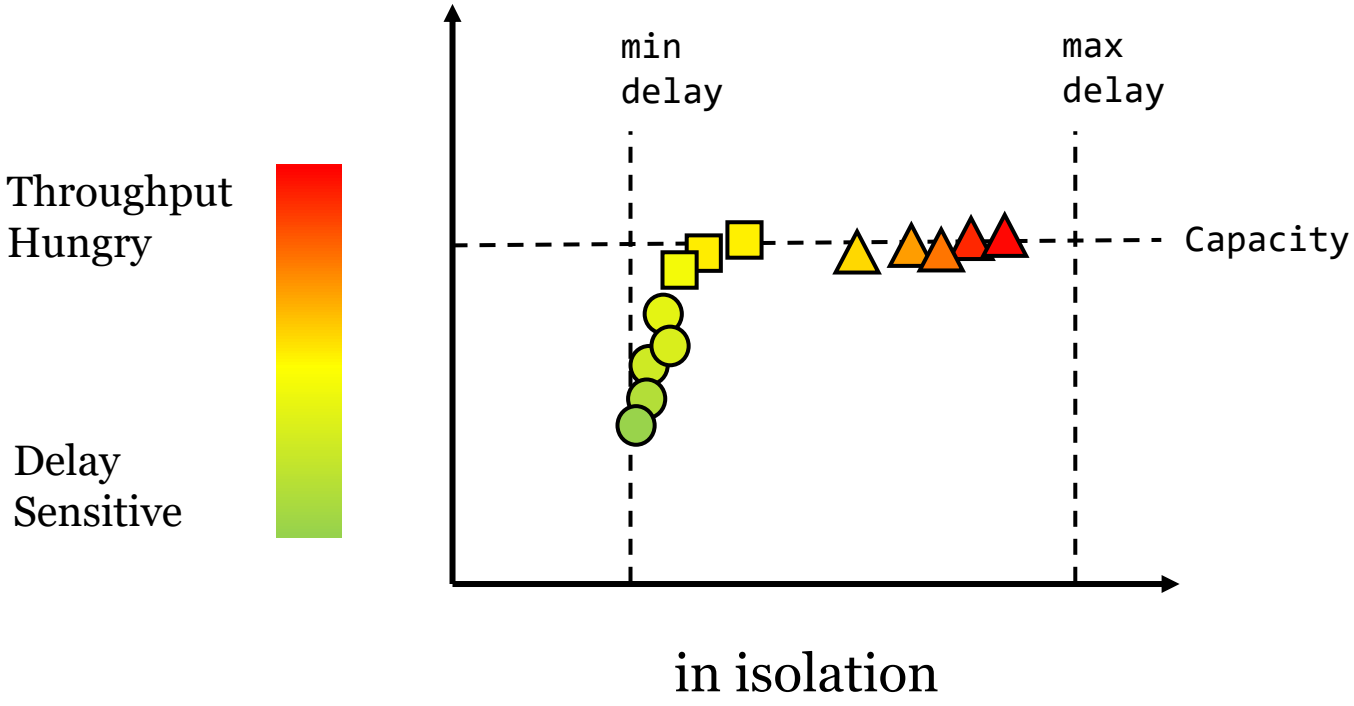
Fair Queuing is the ideal solution



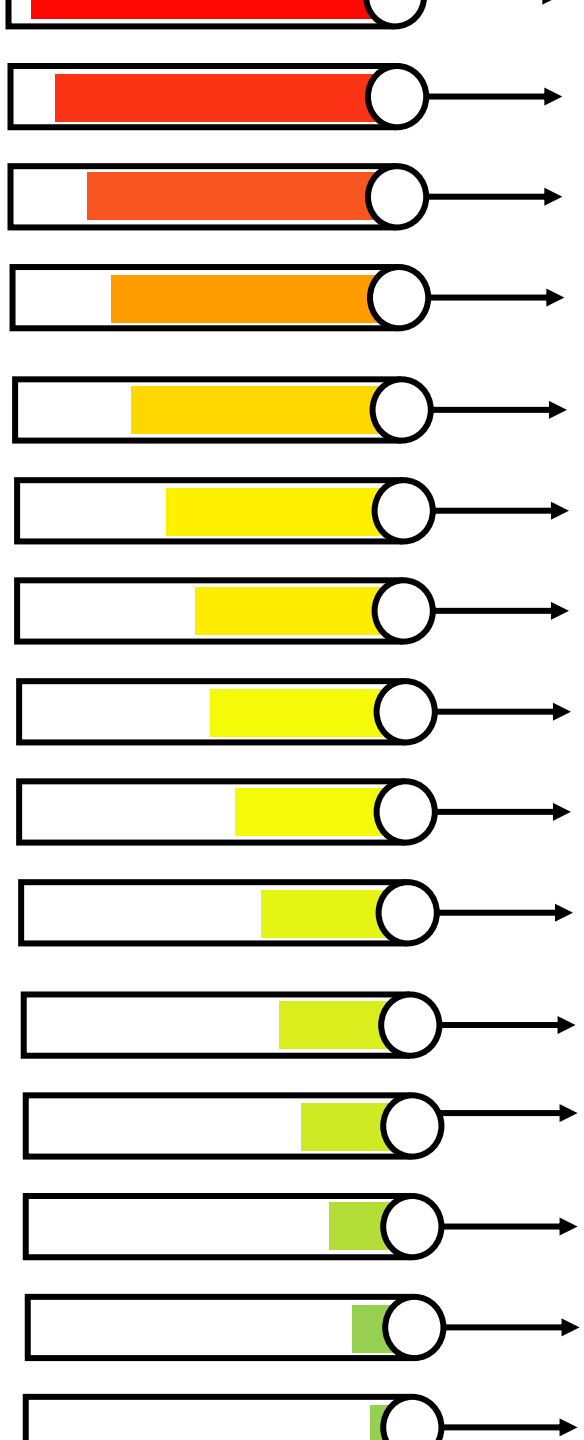
in isolation



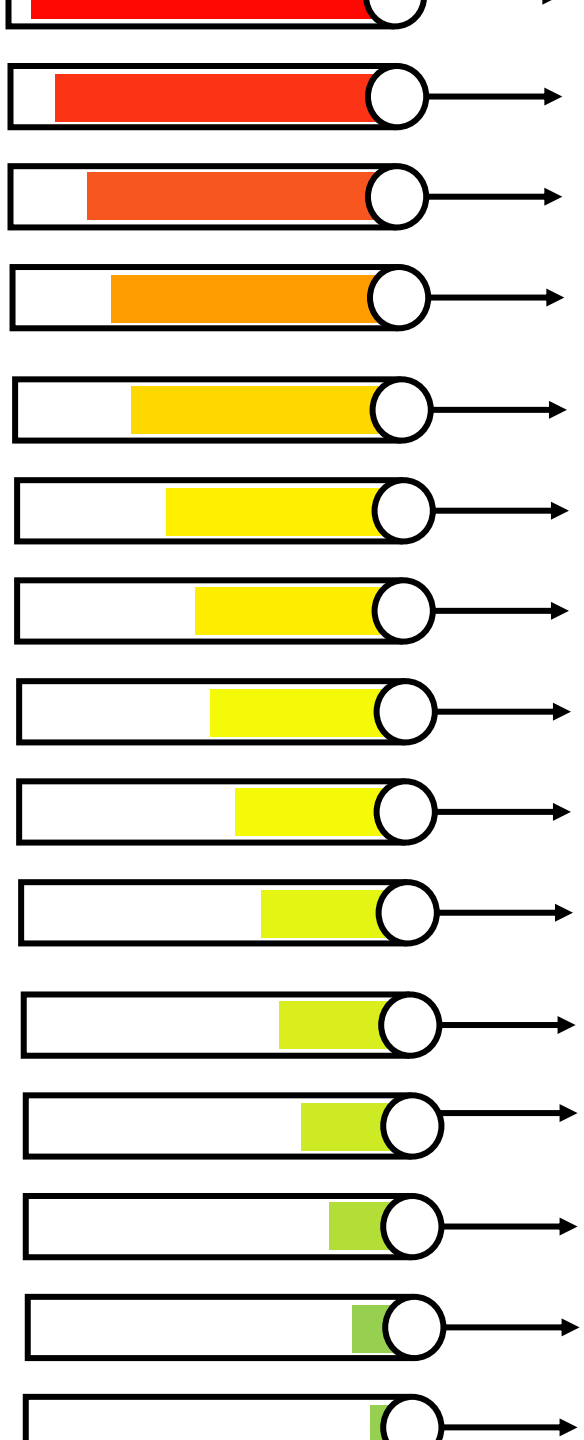
The Internet has too many flows and too few queues



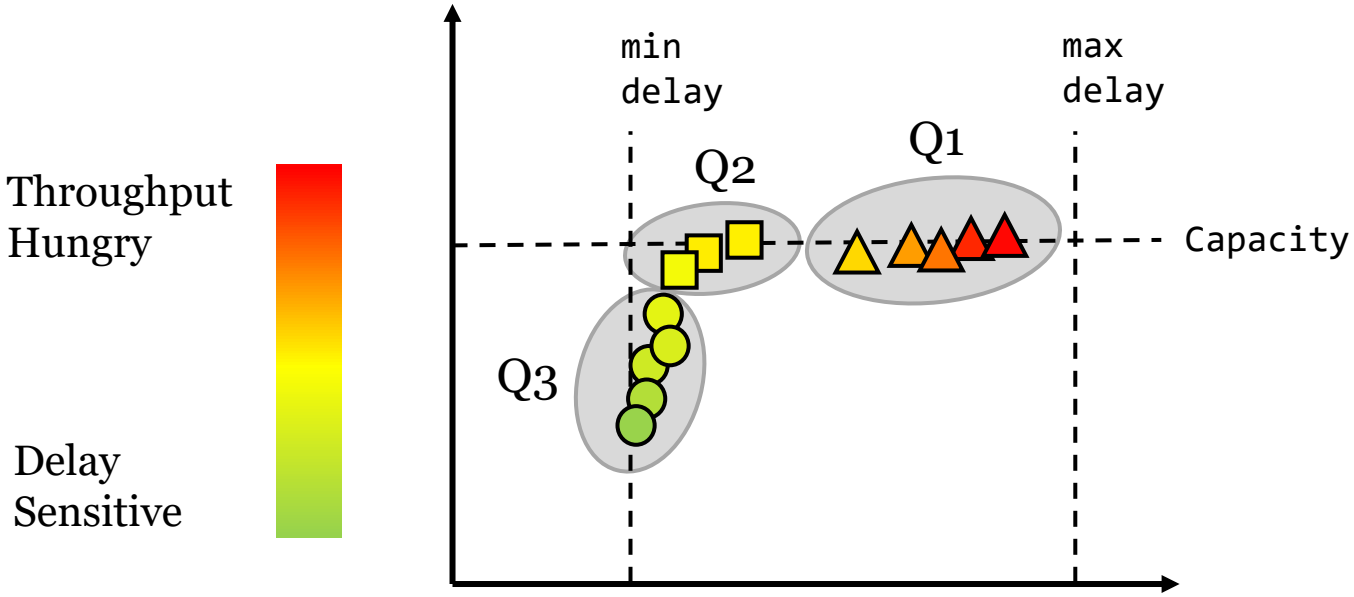
in isolation



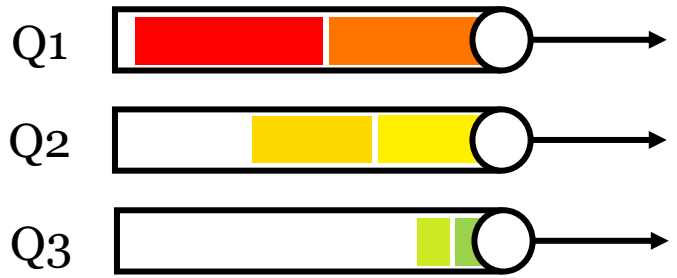
How can we provide Performance Isolation with a handful of queues?



Approximate Performance Isolation

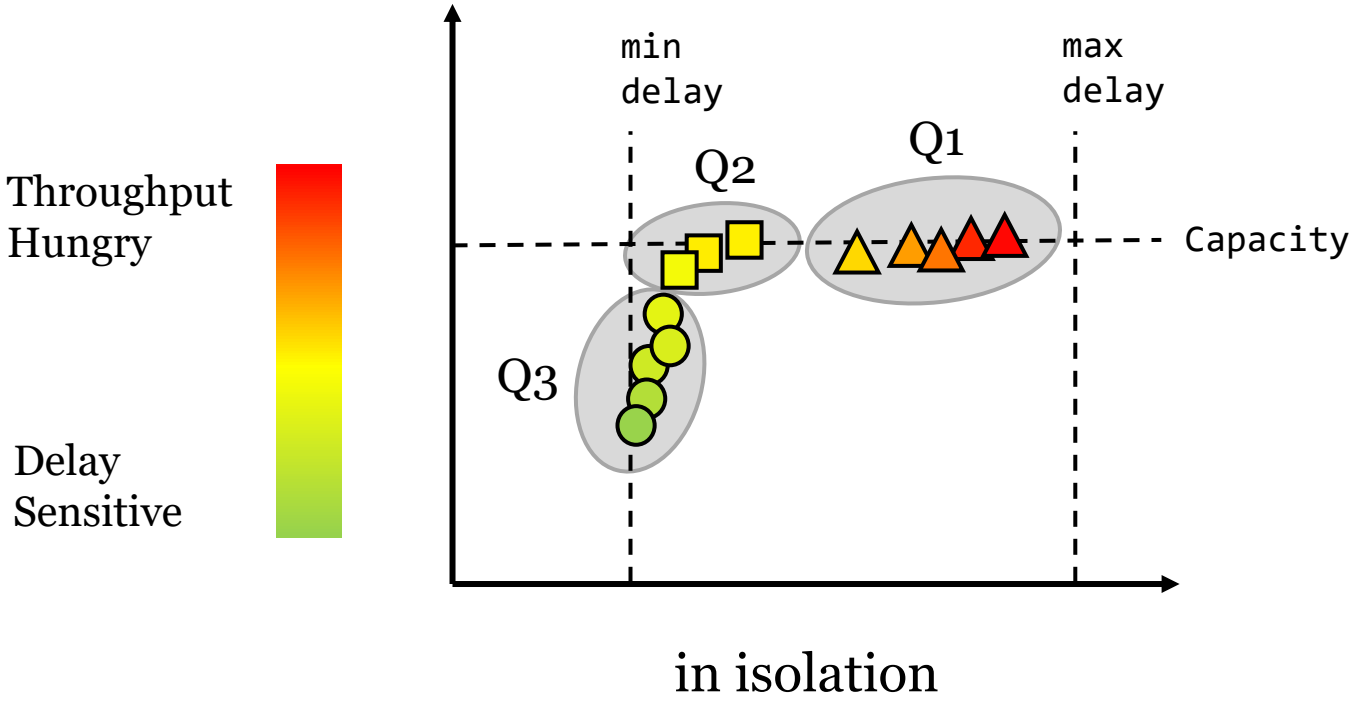


in isolation

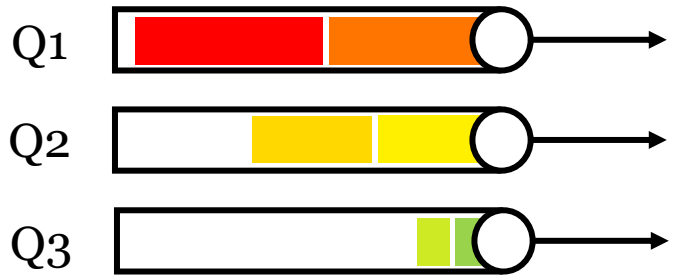


Flows that want similar trade-offs are likely to achieve them together

How do we infer these desired trade-offs?



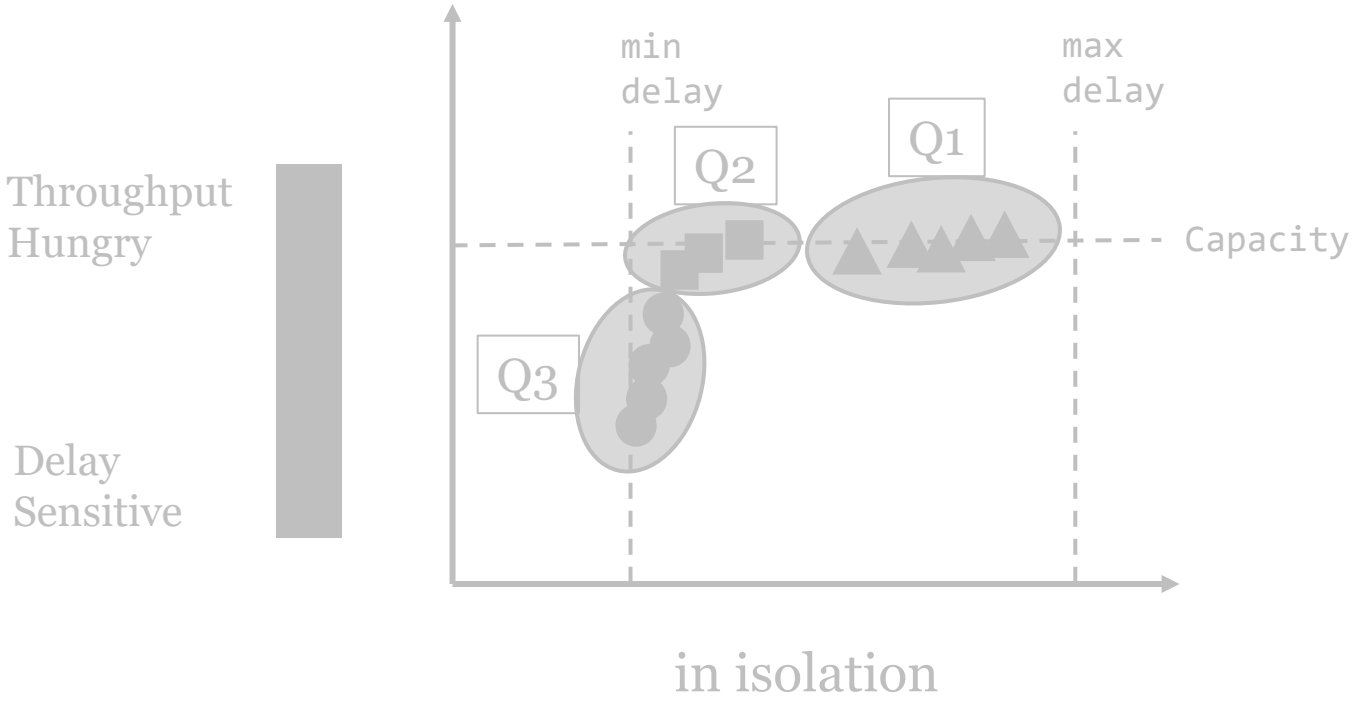
in isolation

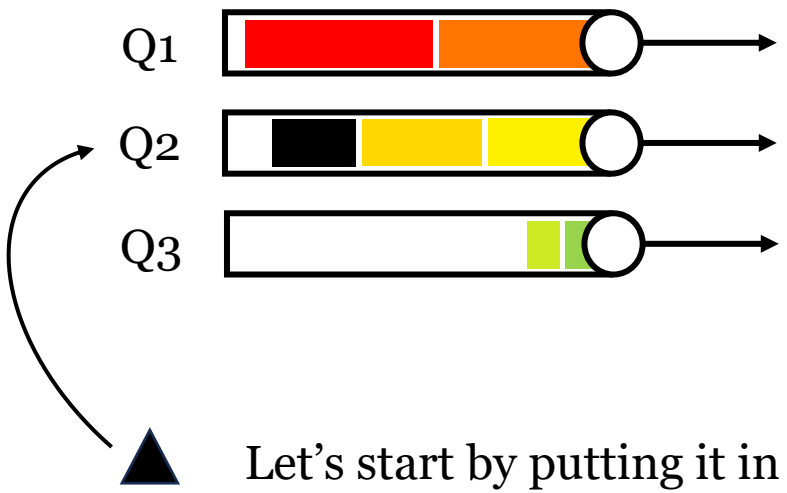


A new flow enters

But we don't know what it wants

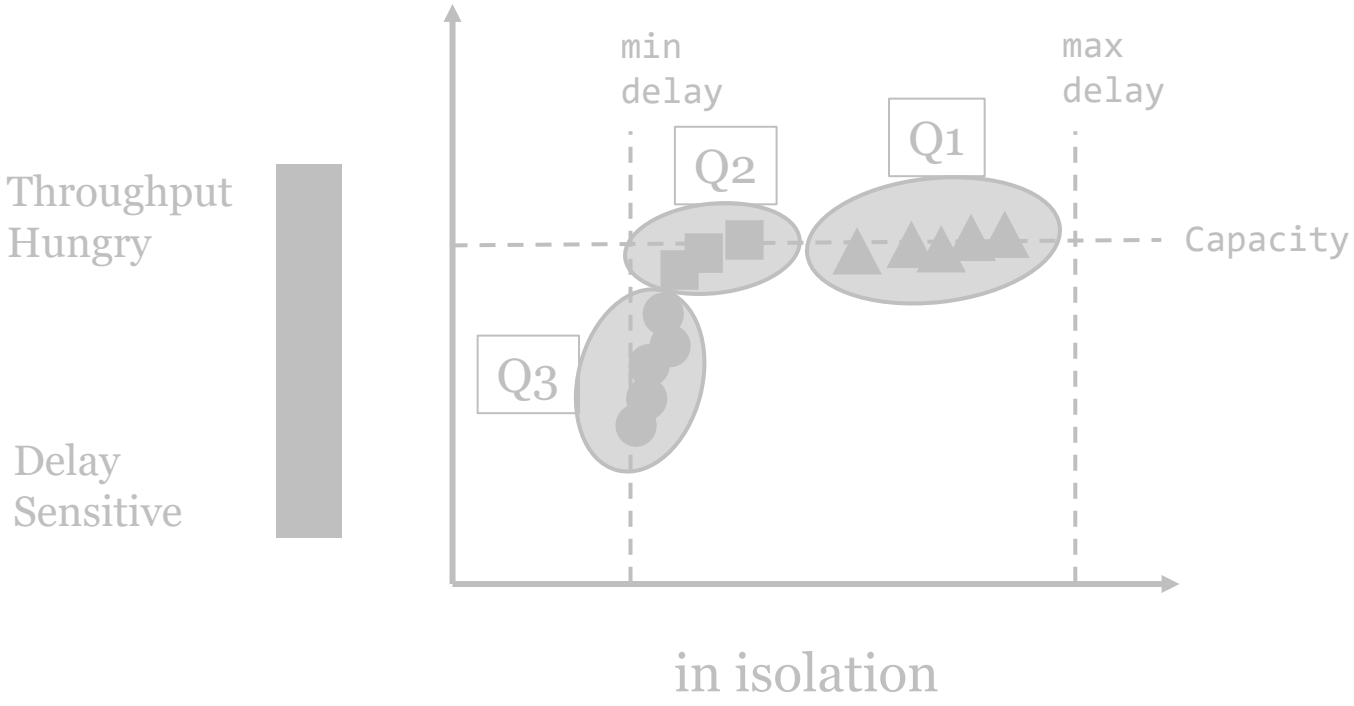
How do we infer these desired trade-offs?

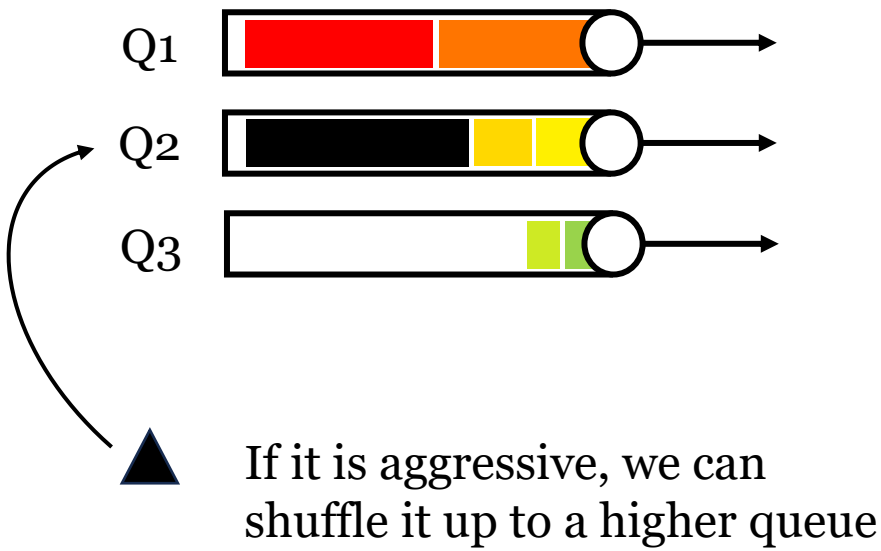




Let's start by putting it in a random queue

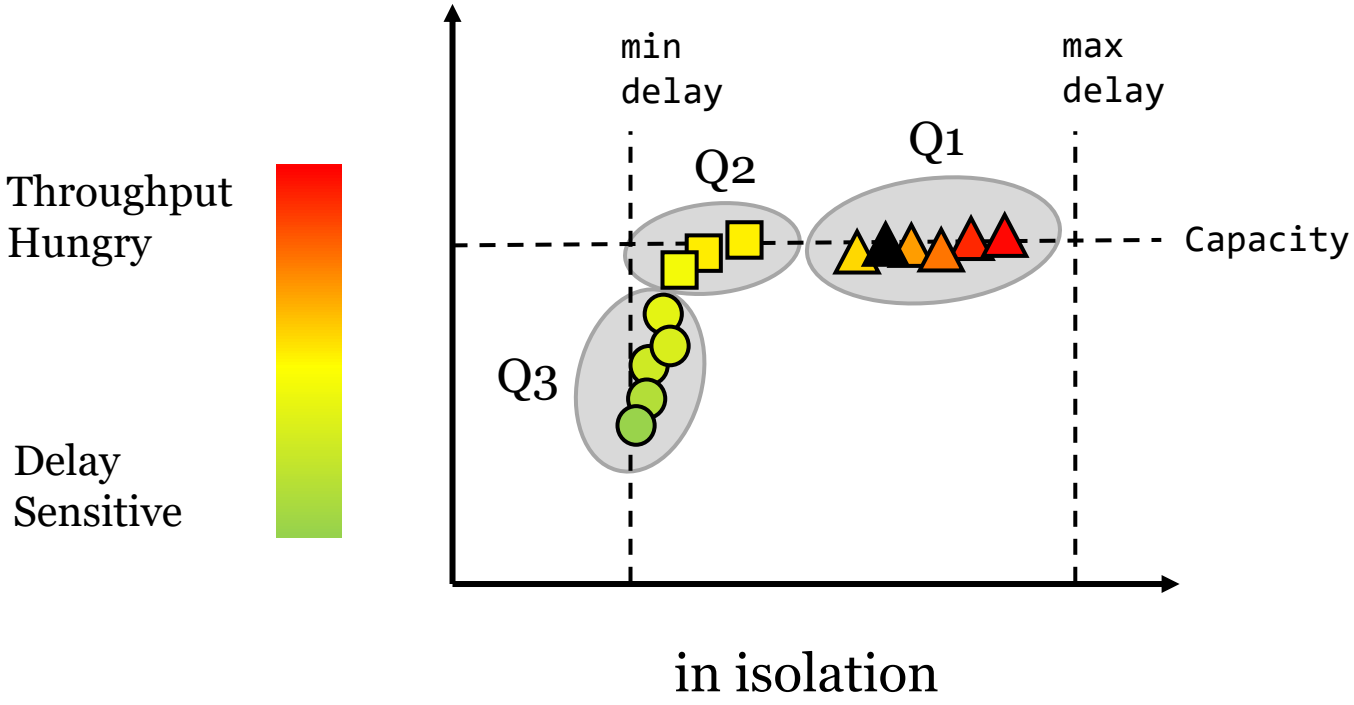
How do we infer these desired trade-offs?



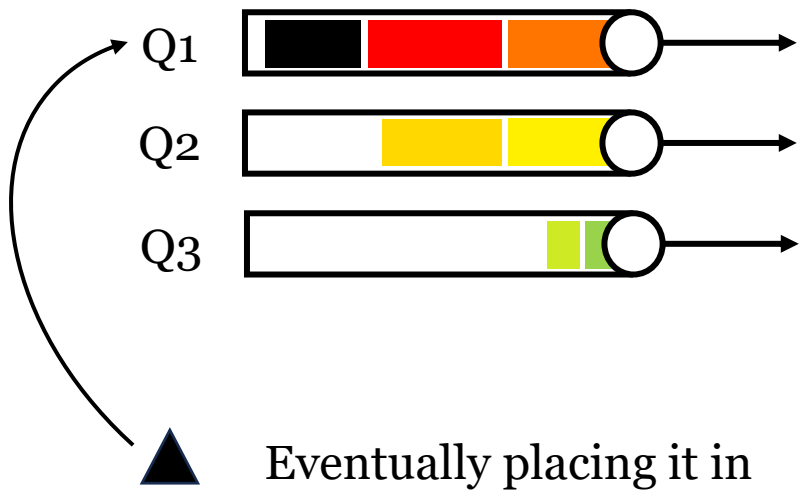


If it is aggressive, we can shuffle it up to a higher queue

How do we infer these desired trade-offs?

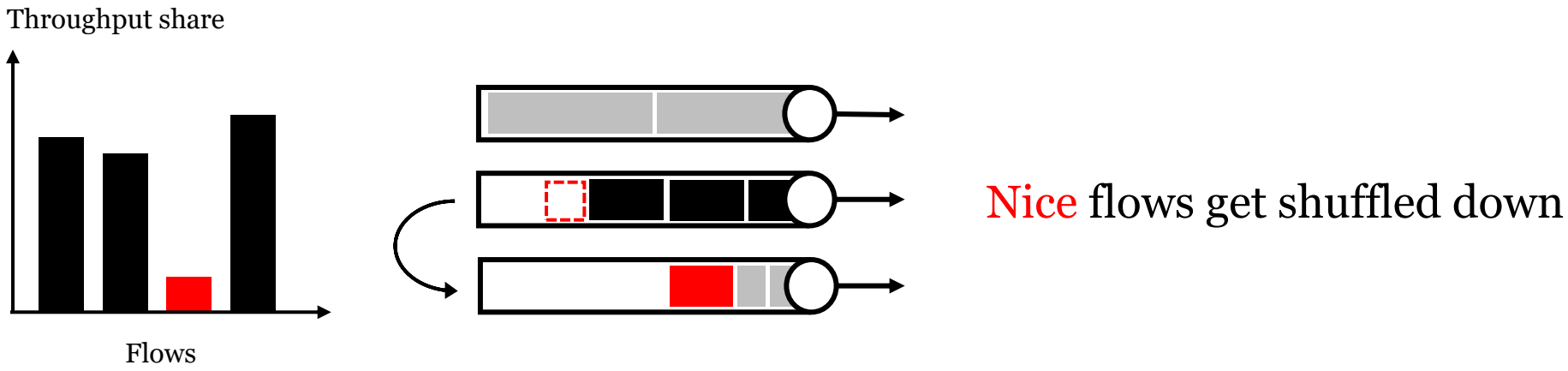


in isolation



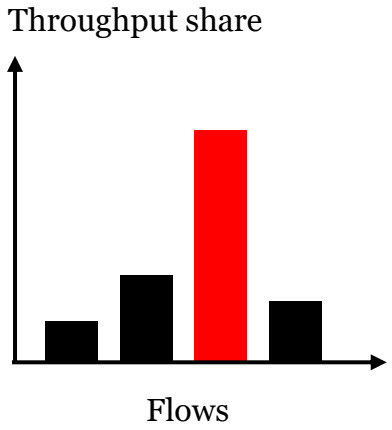
Eventually placing it in the right queue

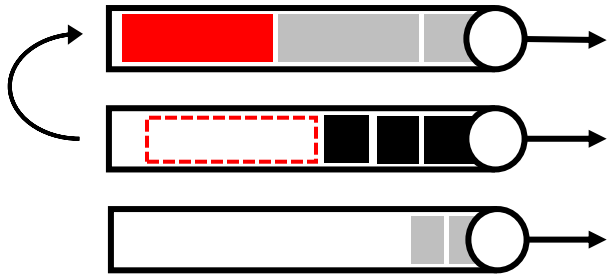
We can perform these shuffles in rounds over all available n queues



Nice flows get shuffled down

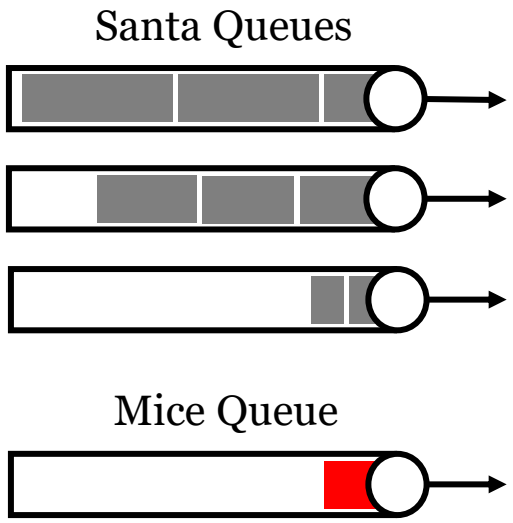
We can perform these shuffles in rounds over all available n queues





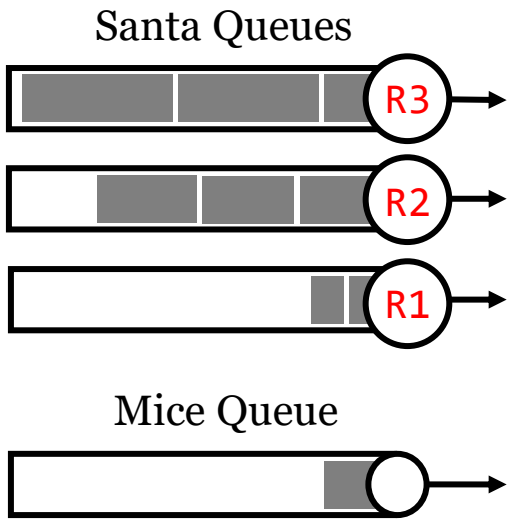
Naughty flows get shuffled up

We implemented Santa on a p4 switch



Dedicated Mice Queue for short flows

We implemented Santa on a p4 switch

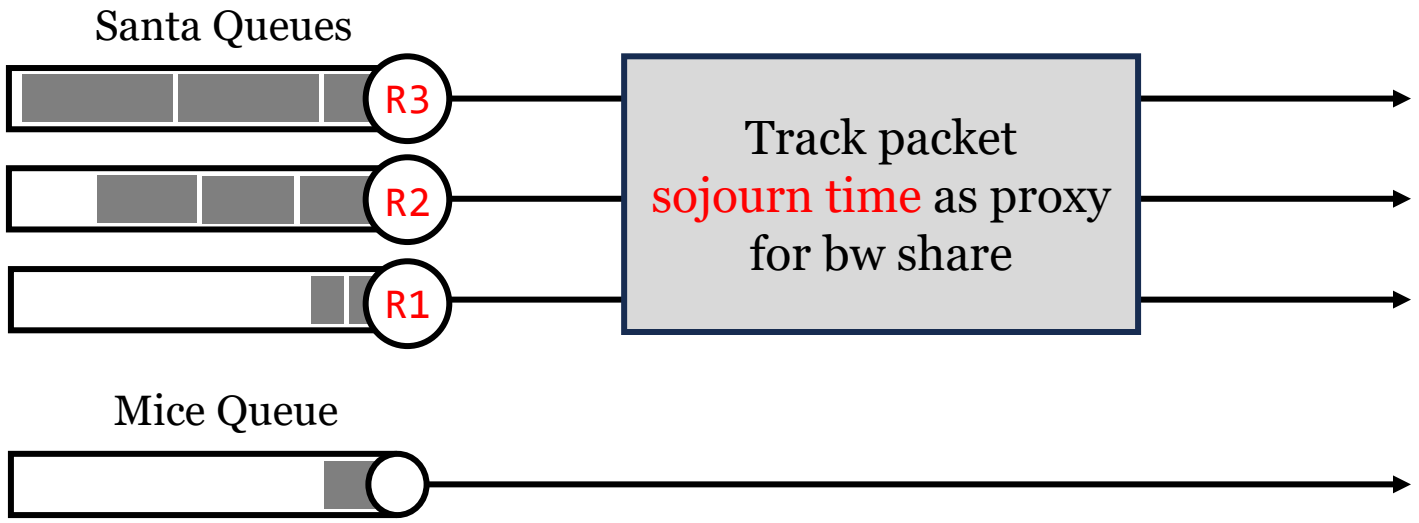


Santa queues dequeue at rates proportional to their number of flows

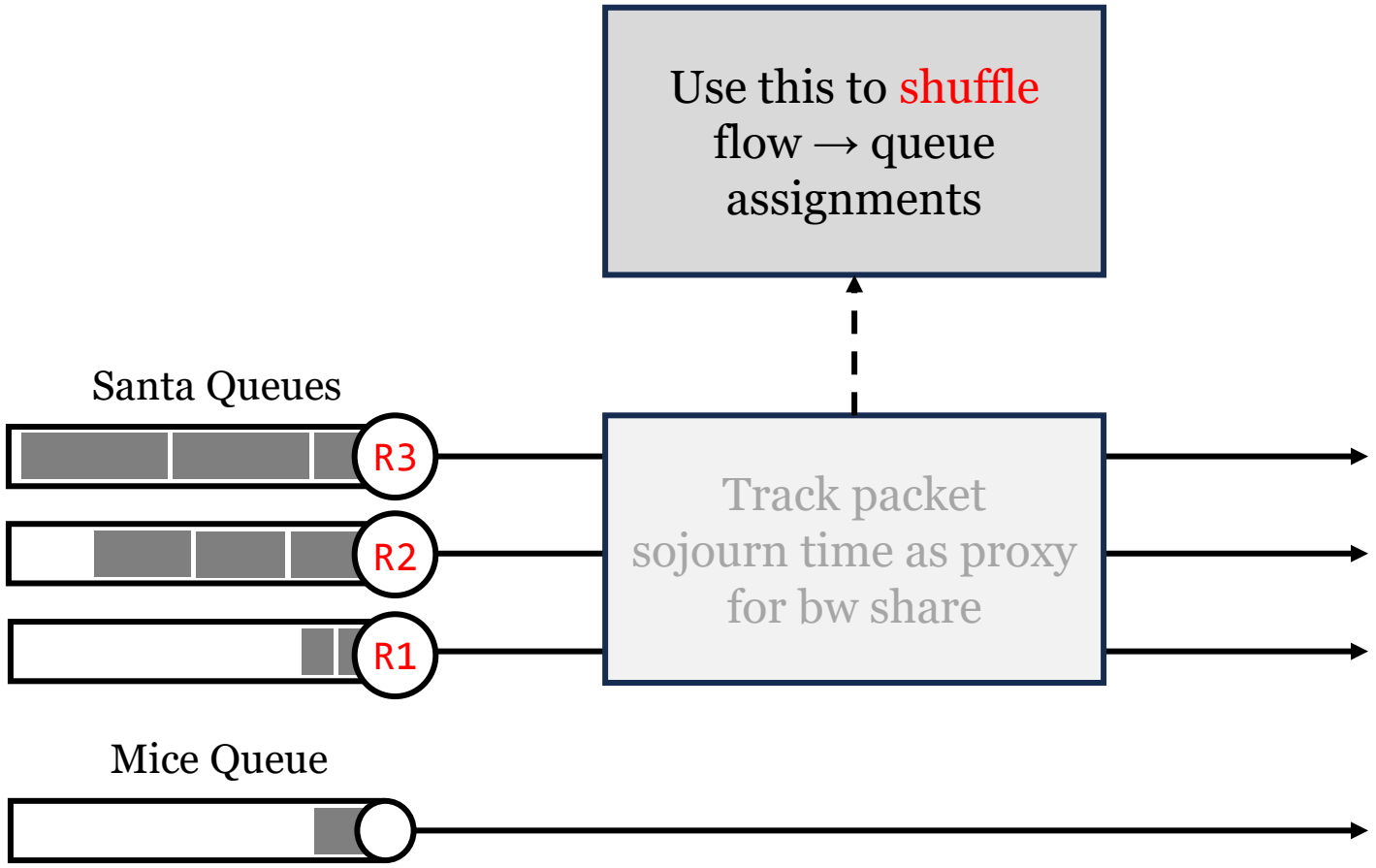
(configurable)

Mice queue gets strict priority

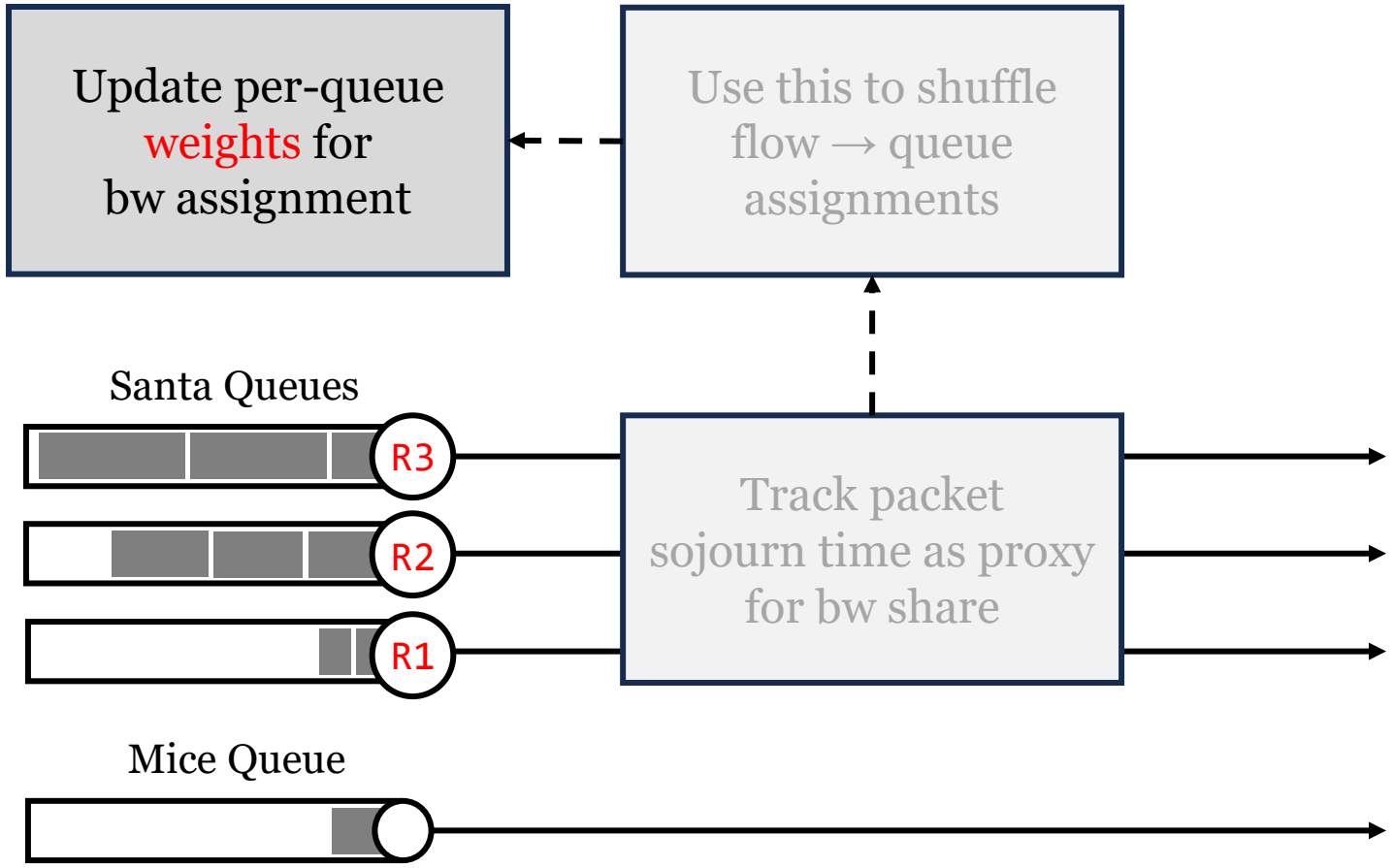
We implemented Santa on a p4 switch



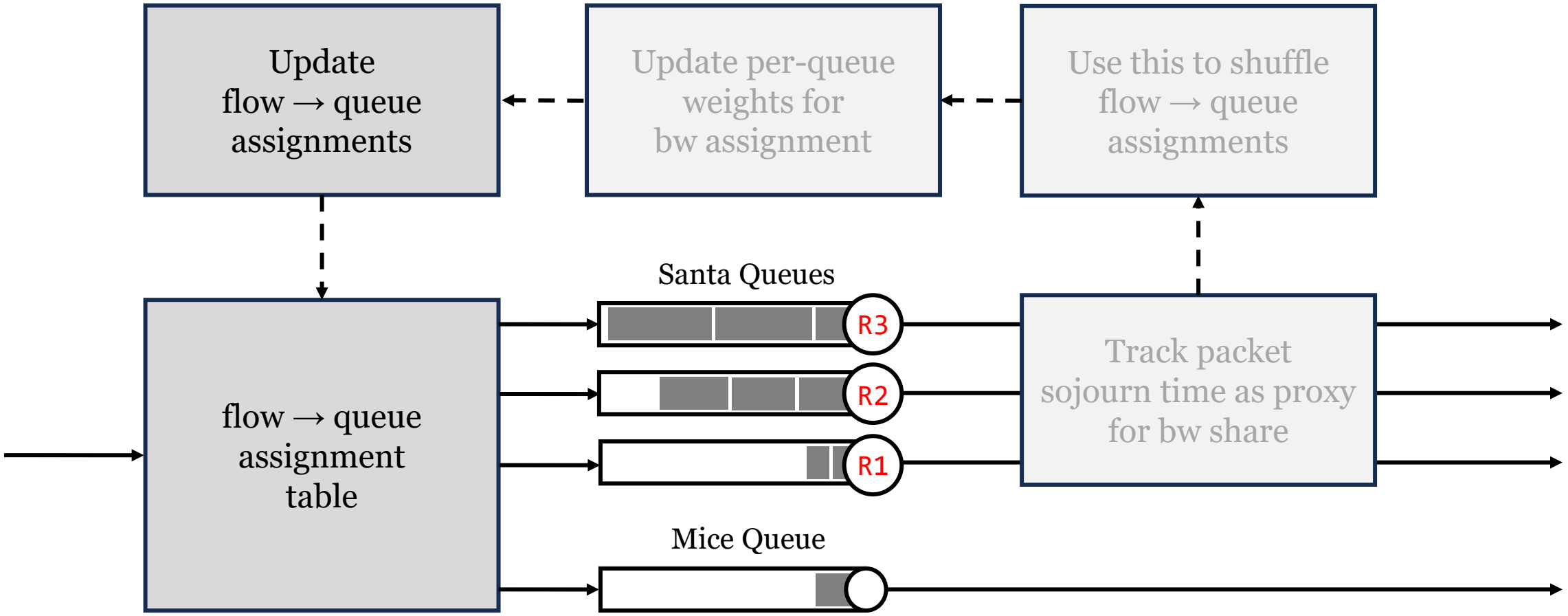
We implemented Santa on a p4 switch



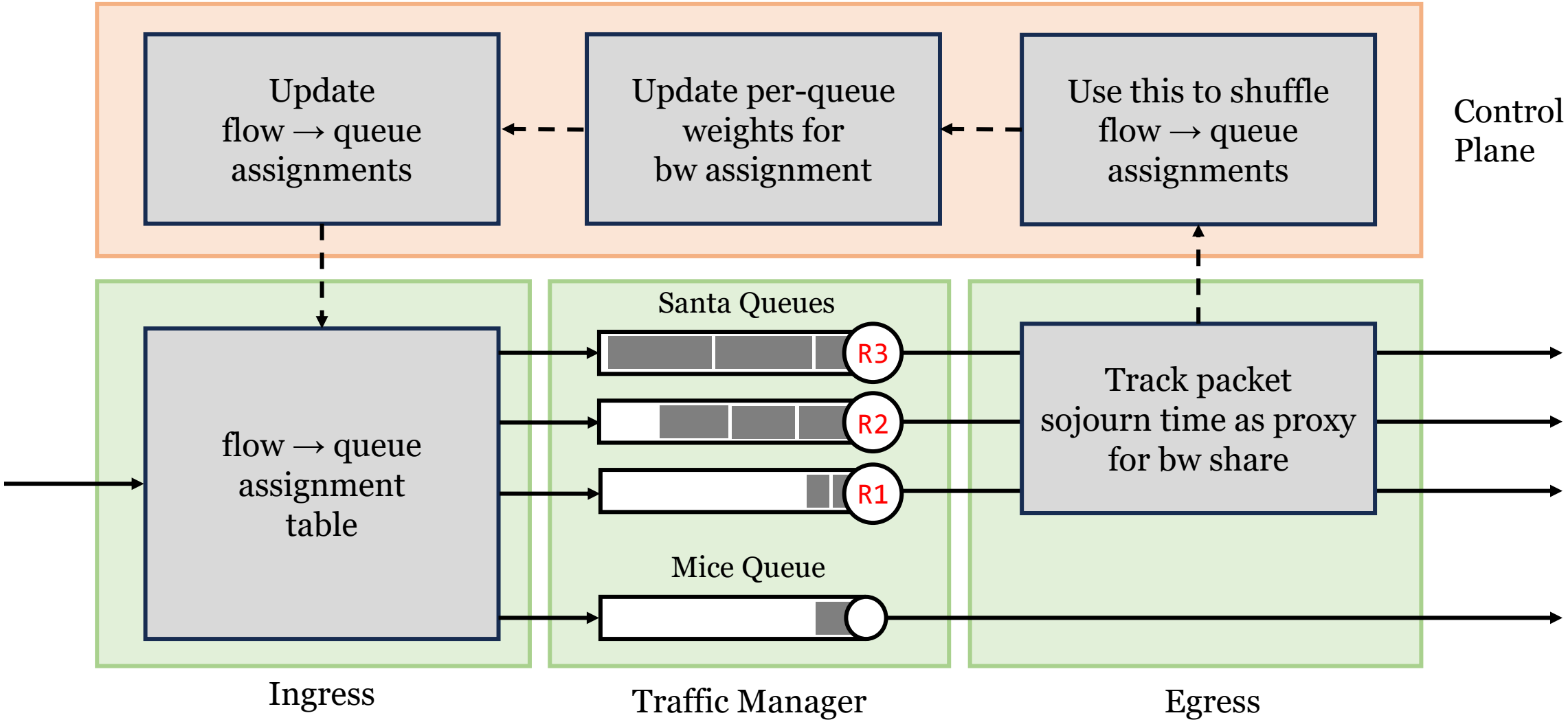
We implemented Santa on a p4 switch



We implemented Santa on a p4 switch



We implemented Santa on a p4 switch

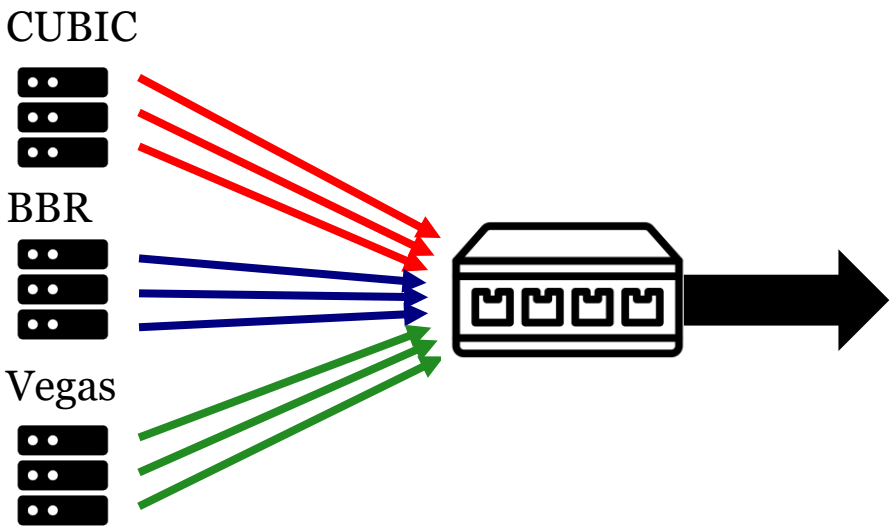


Does Santa provide Approximate Performance Isolation?

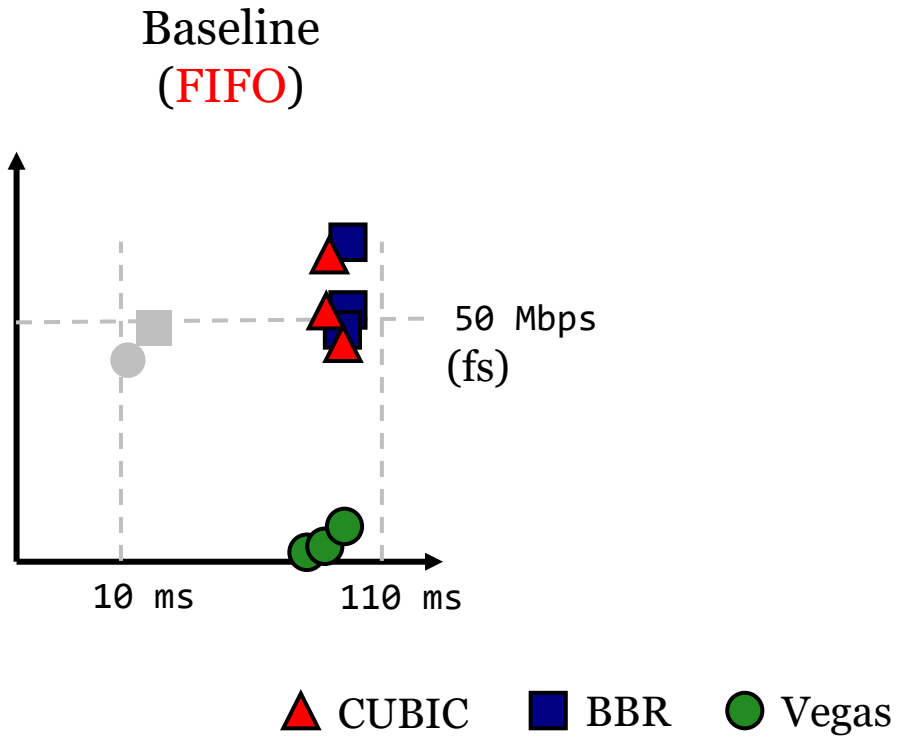
Long running flows (2 minutes)

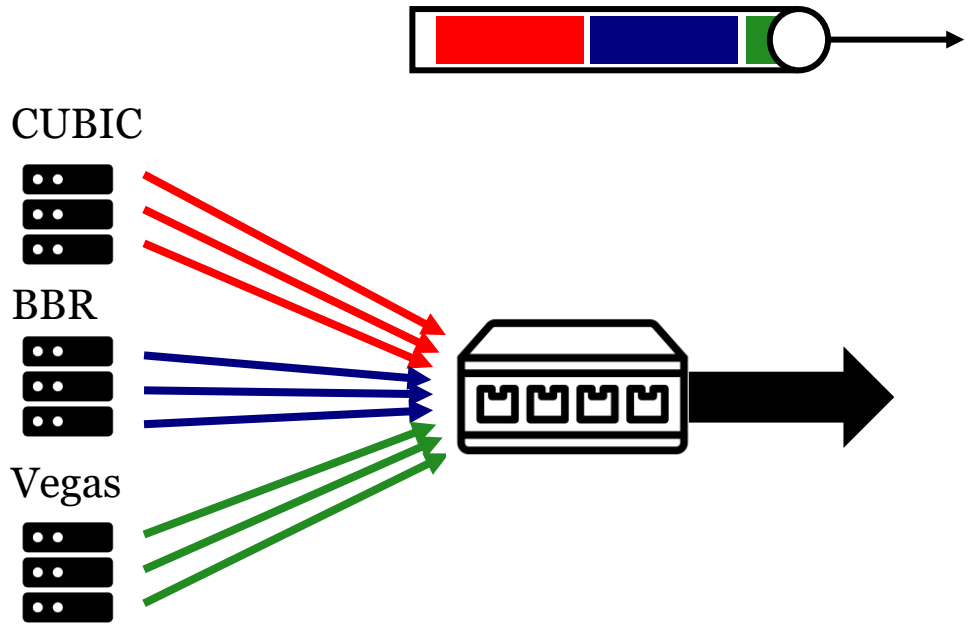
Santa shuffles every 5 seconds

Equal buffer and proportional bandwidth to all 3 queues

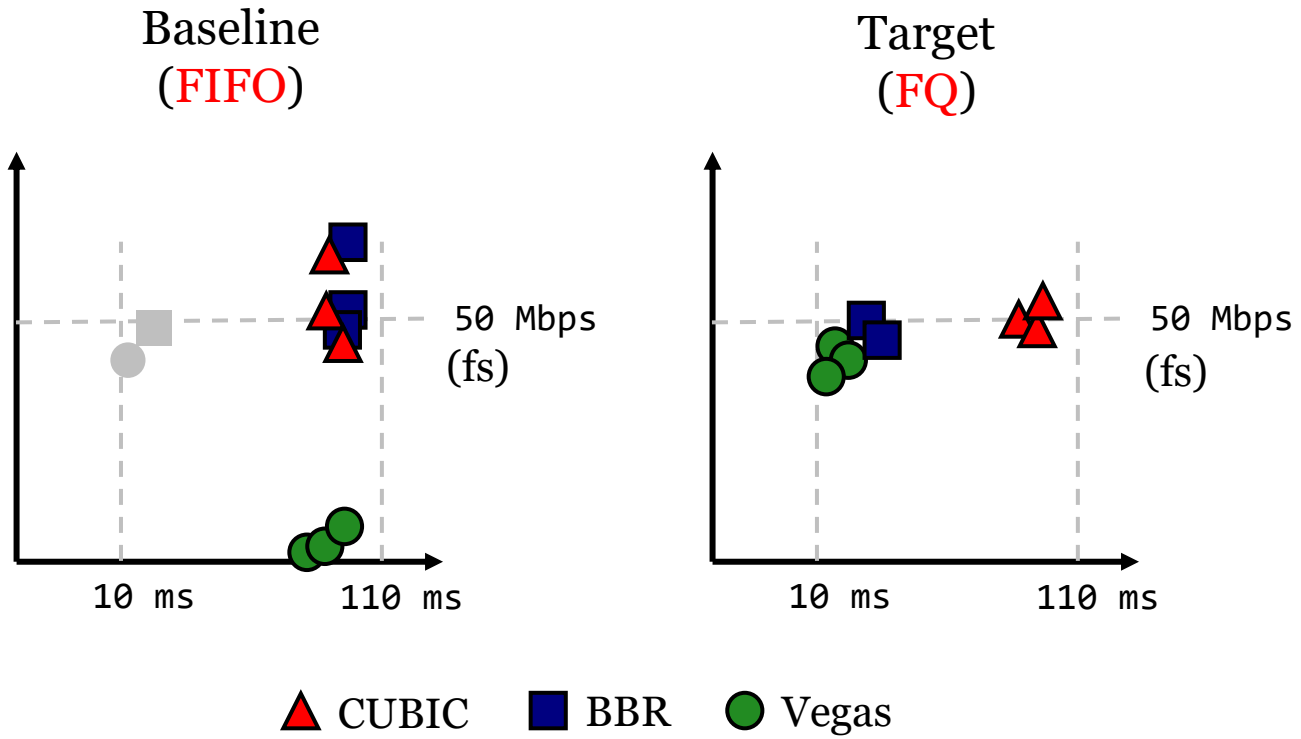


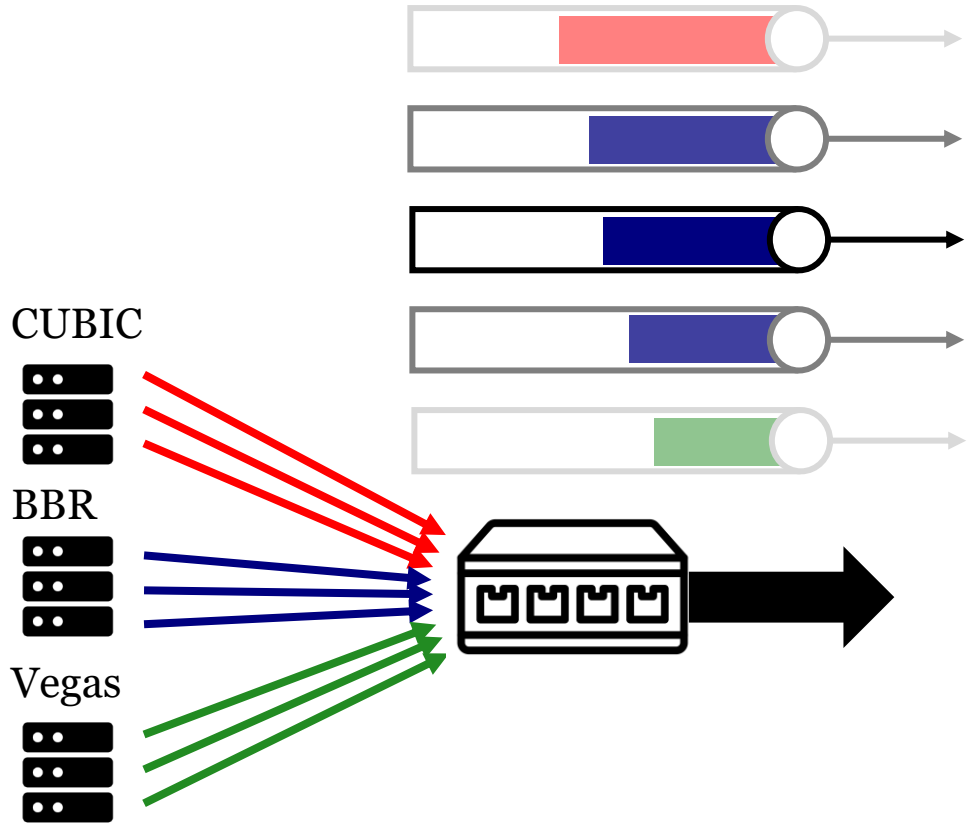
Does Santa provide Approximate Performance Isolation?



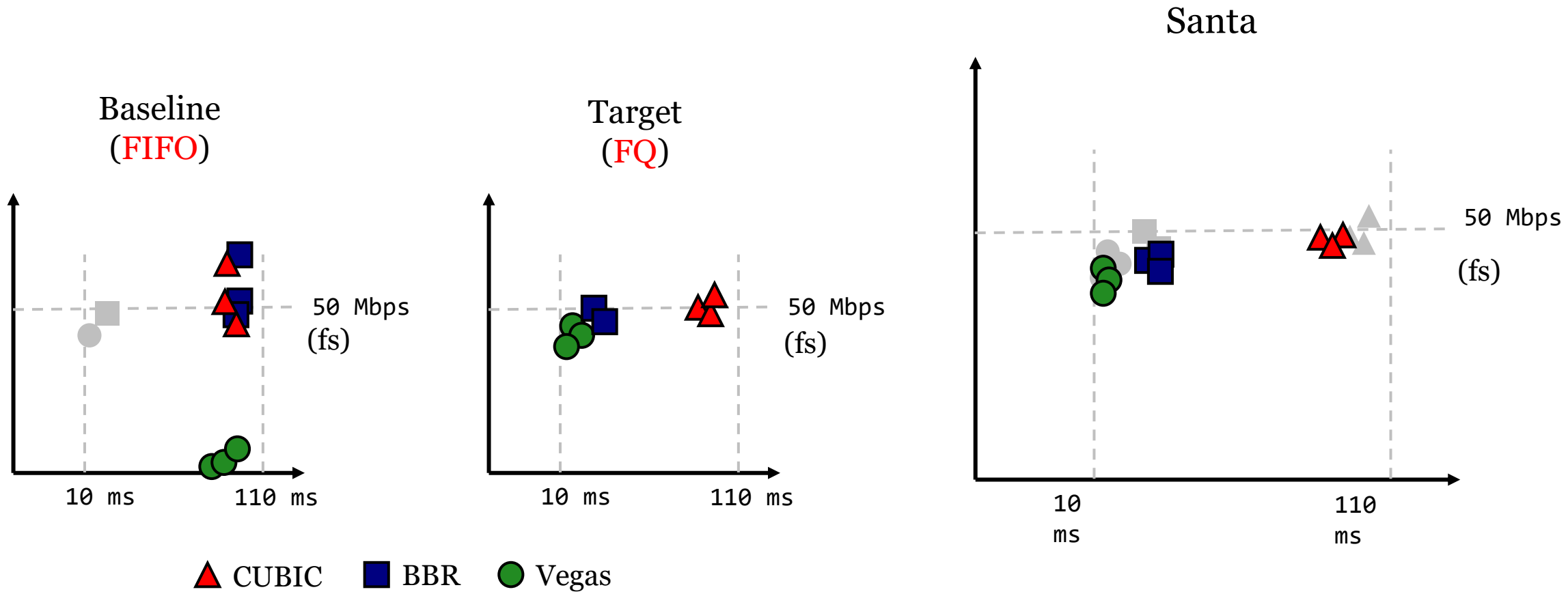


Does Santa provide Approximate Performance Isolation?

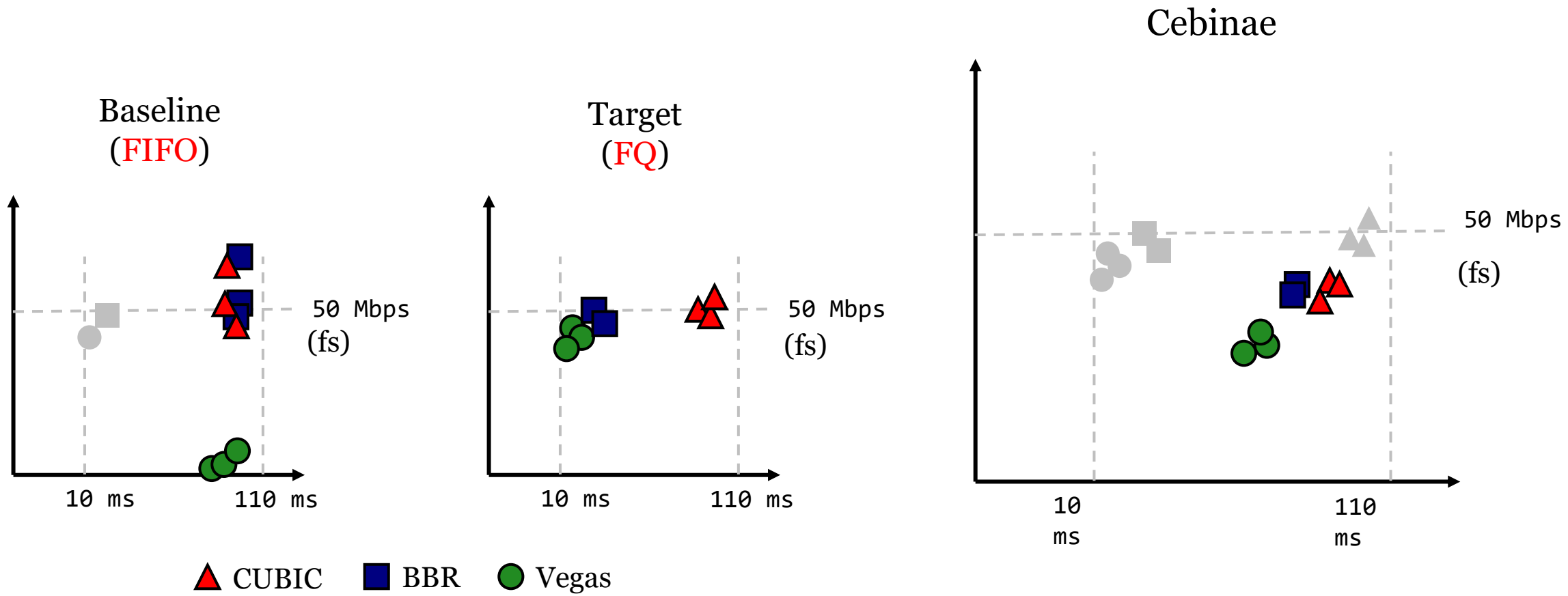




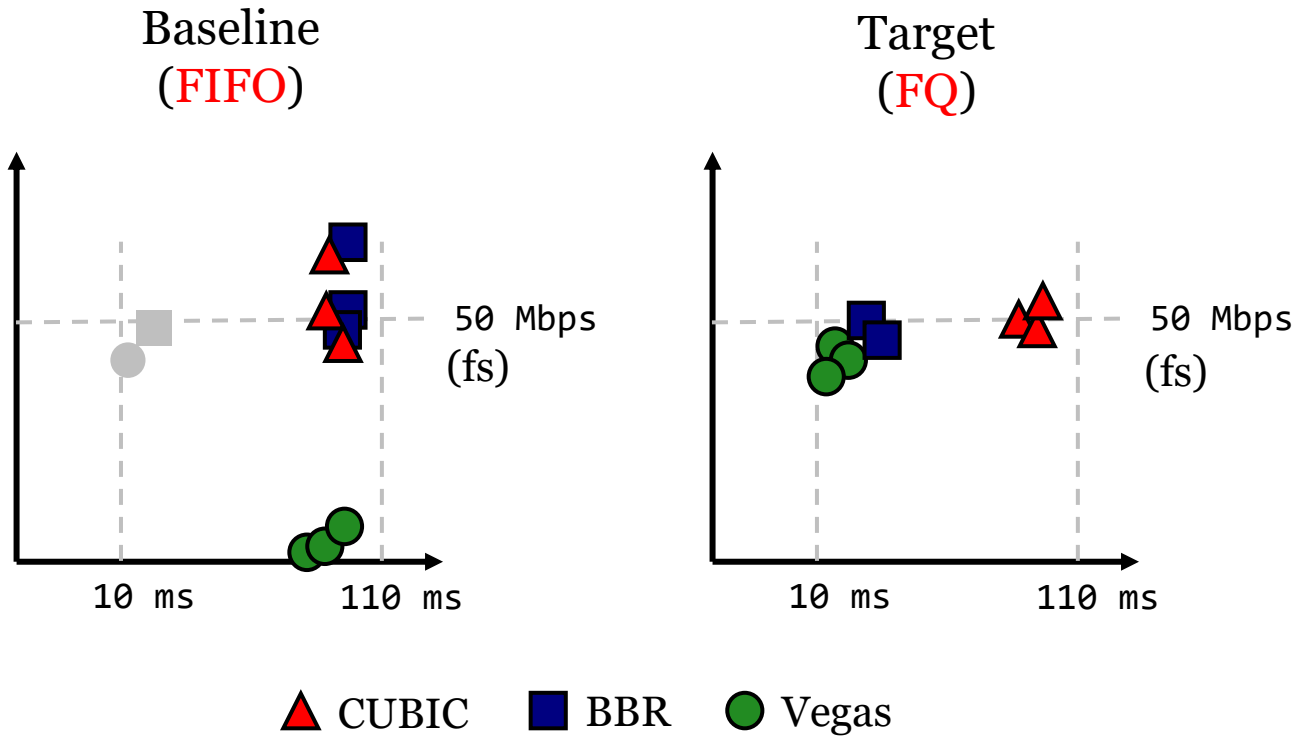
Santa does provide Approximate Performance Isolation!



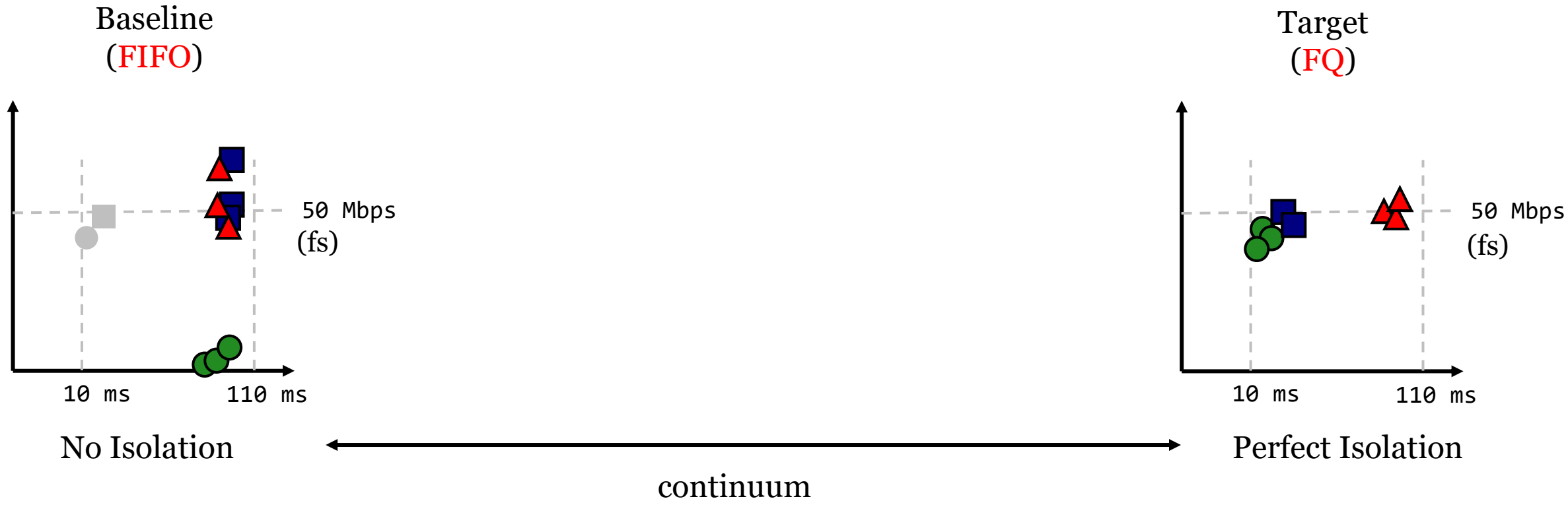
...while other approximate FQ mechanisms fail



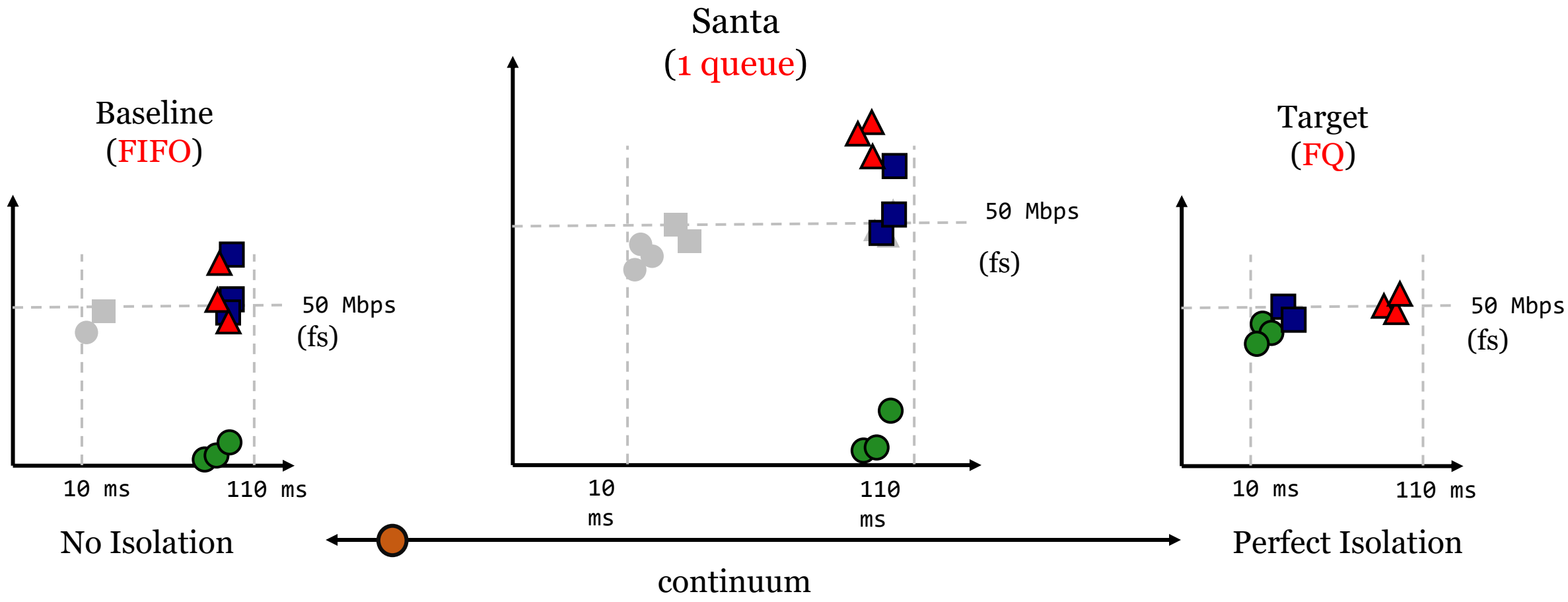
Not enough queues?
No problem.



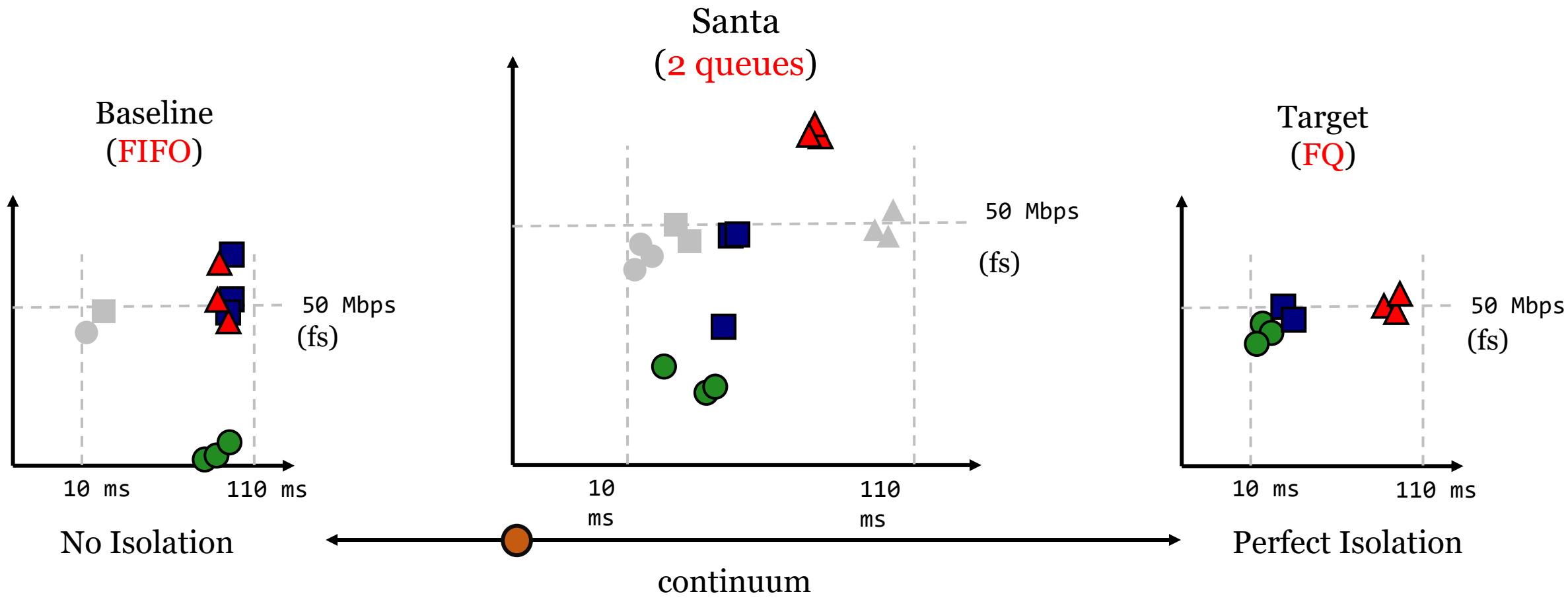
Not enough queues?
No problem.



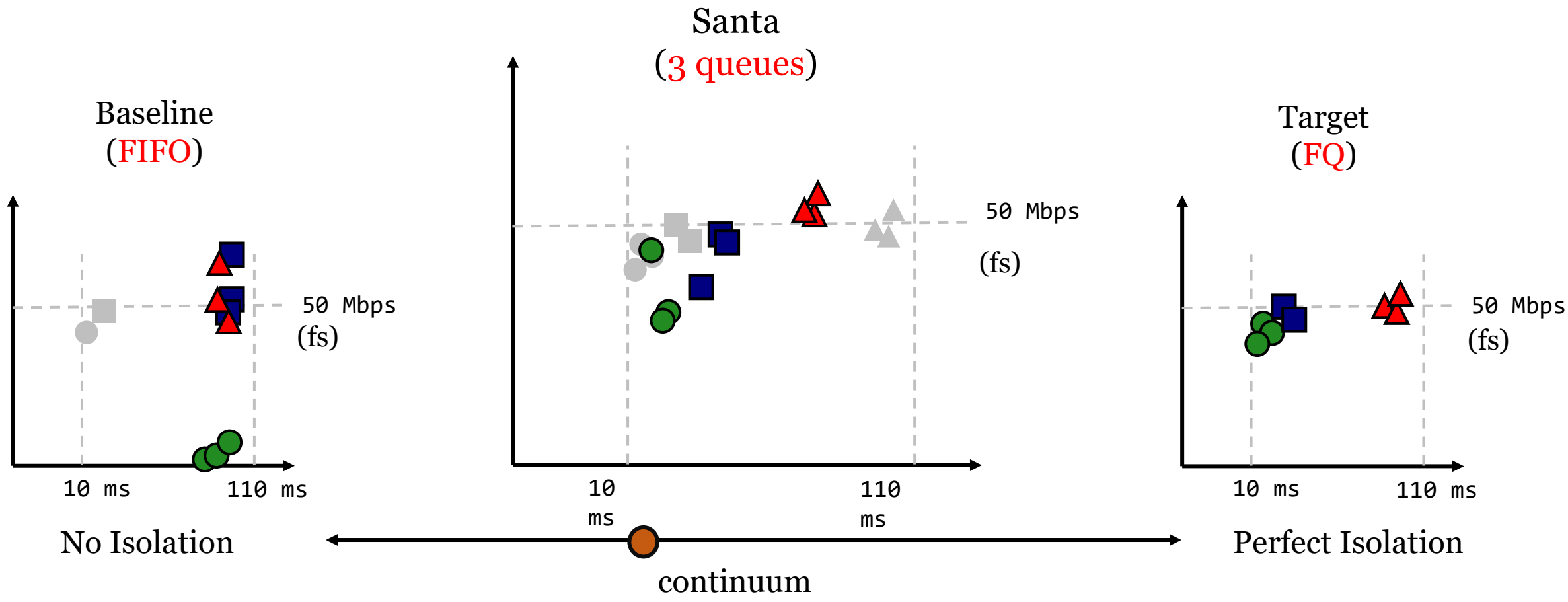
Not enough queues?
No problem.



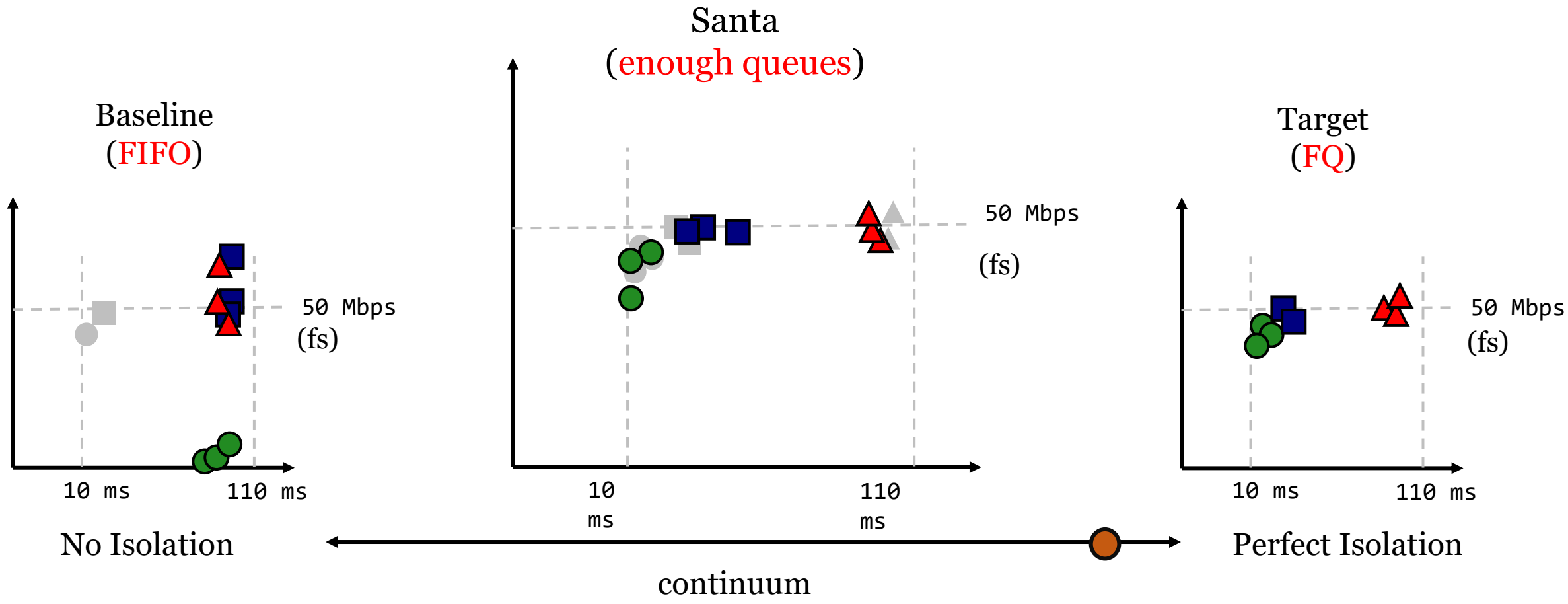
Not enough queues?
No problem.



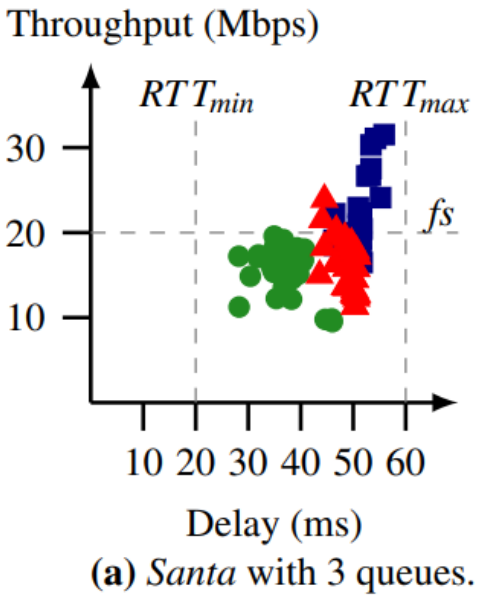
Not enough queues?
No problem.



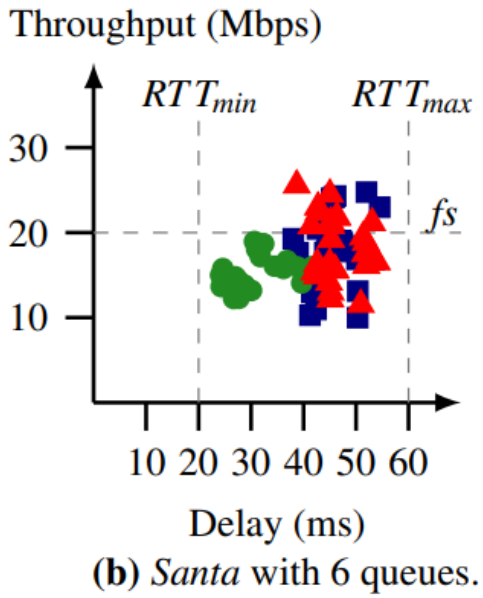
Not enough queues?
No problem.



Santa scales to larger number of flows



(a) *Santa* with 3 queues.



(b) *Santa* with 6 queues.

Tested mixes of
60 and 90 flows

More queues, the better

Santa has its limits

- Shuffle frequency is not an easy parameter to tune
  - Tradeoff between reactivity and churn
- We need to explore more dynamic buffer and bandwidth allocation strategies
  - More bandwidth to throughput hungry flows?
- New mechanism, new vulnerabilities?
  - How sound is this mechanism from a Game Theoretic point of view?

Learning to compare
Apples to Oranges on the Internet

We motivate Approximate Performance Isolation as a goal for the modern Internet

We demonstrate how Santa can achieve these goals practically

Santa is open source and available on GitHub!

NSDI 2026
Renton, WA, USA
5th May 2026

Managing Congestion Control Heterogeneity on the Internet with *Approximate Performance Isolation*

Ayush Mishra[1], Archit Bhatnagar[2], Yixuan Zhang[3], Ben Leong[4], Gao Ya[5], Raj Joshi[6,7]

[1]ETH Zurich, [2]University of Michigan, [3]Tsinghua University
[4]National University of Singapore, [5]Wuxi Institute of Technology
[6]Harvard University, [7]Red Hat AI