Planck: Millisecond-scale Monitoring and Control for Commodity Networks

Jeff Rasley, Brent Stephens, Colin Dixon, Eric Rozner, Wes Felter, Kanak Agarwal, John Carter, Rodrigo Fonseca

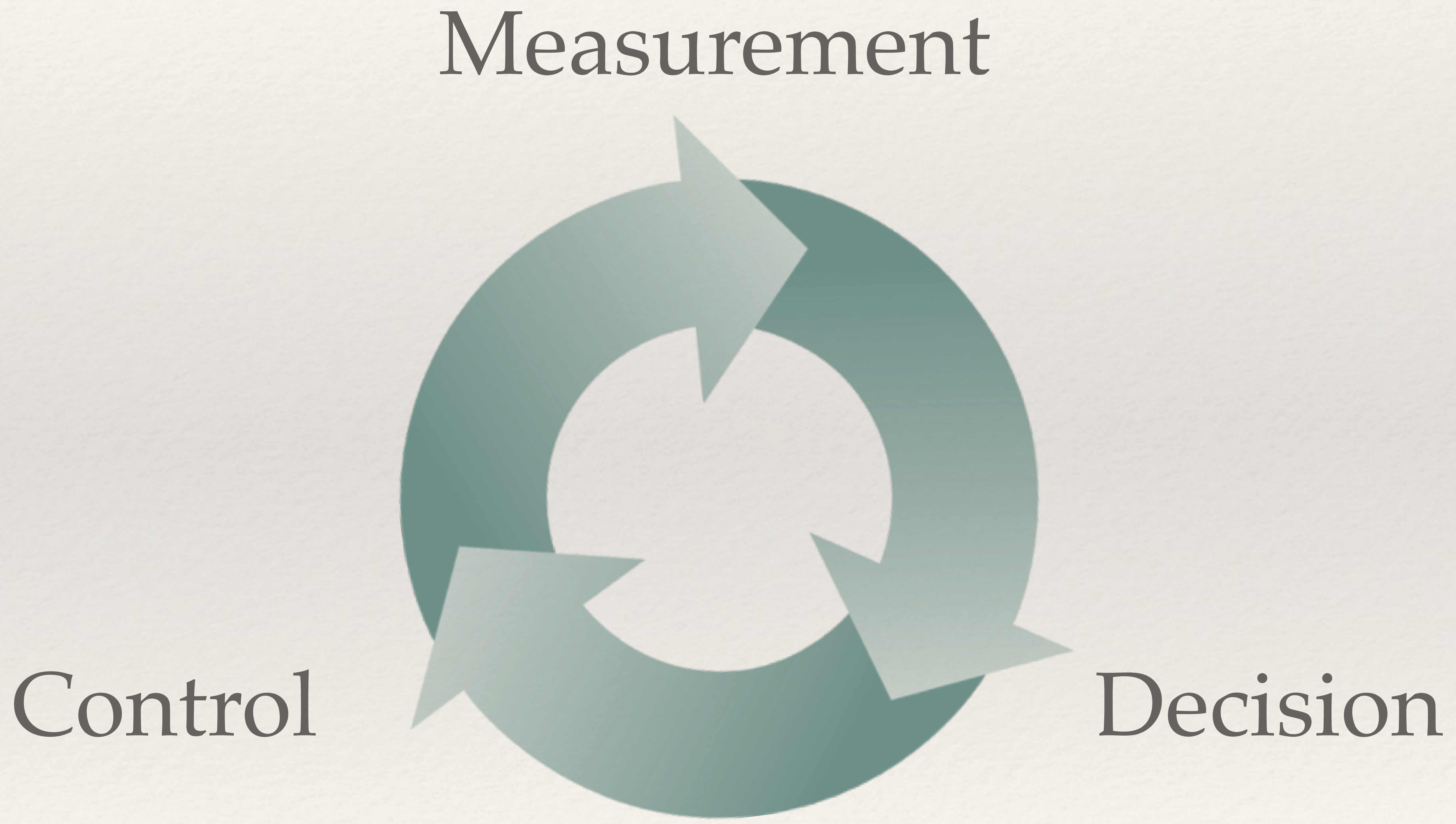
BROWN RICE IBM Research BROCADE

Self-Tuning Networks

Control Loop Examples

- Traffic Engineering
- Failure Avoidance
- Forwarding Behavior Verification

How fast can we do this?



Self-Tuning Networks

Control Loop Examples

- Traffic Engineering
- Failure Avoidance
- Forwarding Behavior Verification

How fast can we do this?

Self-Tuning Networks

Control Loop Examples

- Traffic Engineering
- Failure Avoidance
- Forwarding Behavior Verification

How fast can we do this?

Self-Tuning Networks

Control Loop Examples

- Traffic Engineering
- Failure Avoidance
- Forwarding Behavior Verification

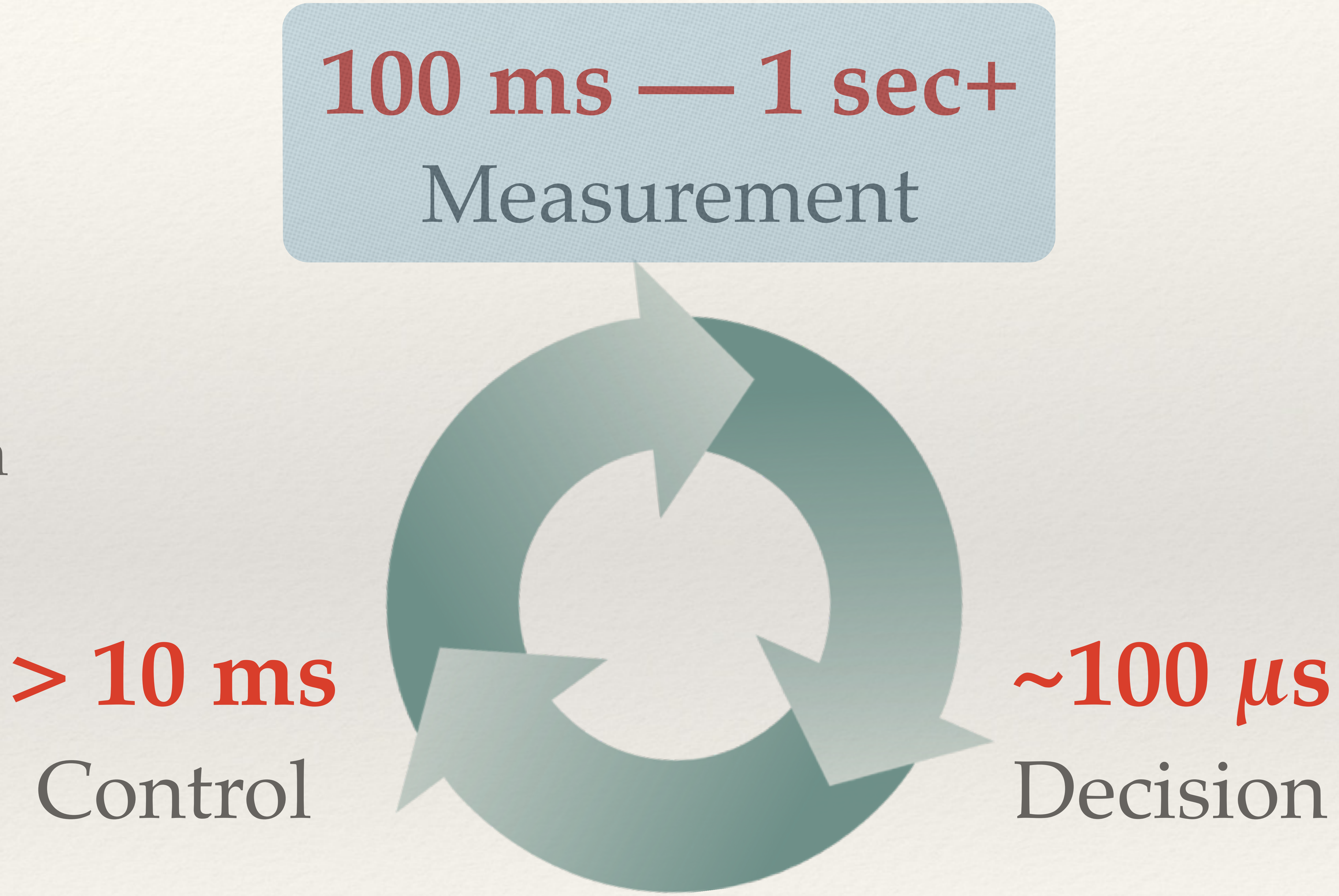
How fast can we do this?

Self-Tuning Networks

Control Loop Examples

- Traffic Engineering
- Failure Avoidance
- Forwarding Behavior Verification

How fast can we do this?



State-of-the-Art Measurement

System	Measurement Speed (ms)
Hedera (NSDI '10)	5,000
DevoFlow Polling (Sigcomm '11)	500–15,000
Mahout Polling (Infocom '11)	190
sFlow/OpenSample (ICDCS '14)	100
Helios (Sigcomm '10)	77.4

State-of-the-Art Measurement

System	Measurement Speed (ms)
Hedera (NSDI '10)	5,000
DevoFlow Polling (Sigcomm '11)	500–15,000
Mahout Polling (Infocom '11)	190
sFlow/OpenSample (ICDCS '14)	100
Helios (Sigcomm '10)	77.4
Planck	< 4.2

Outline

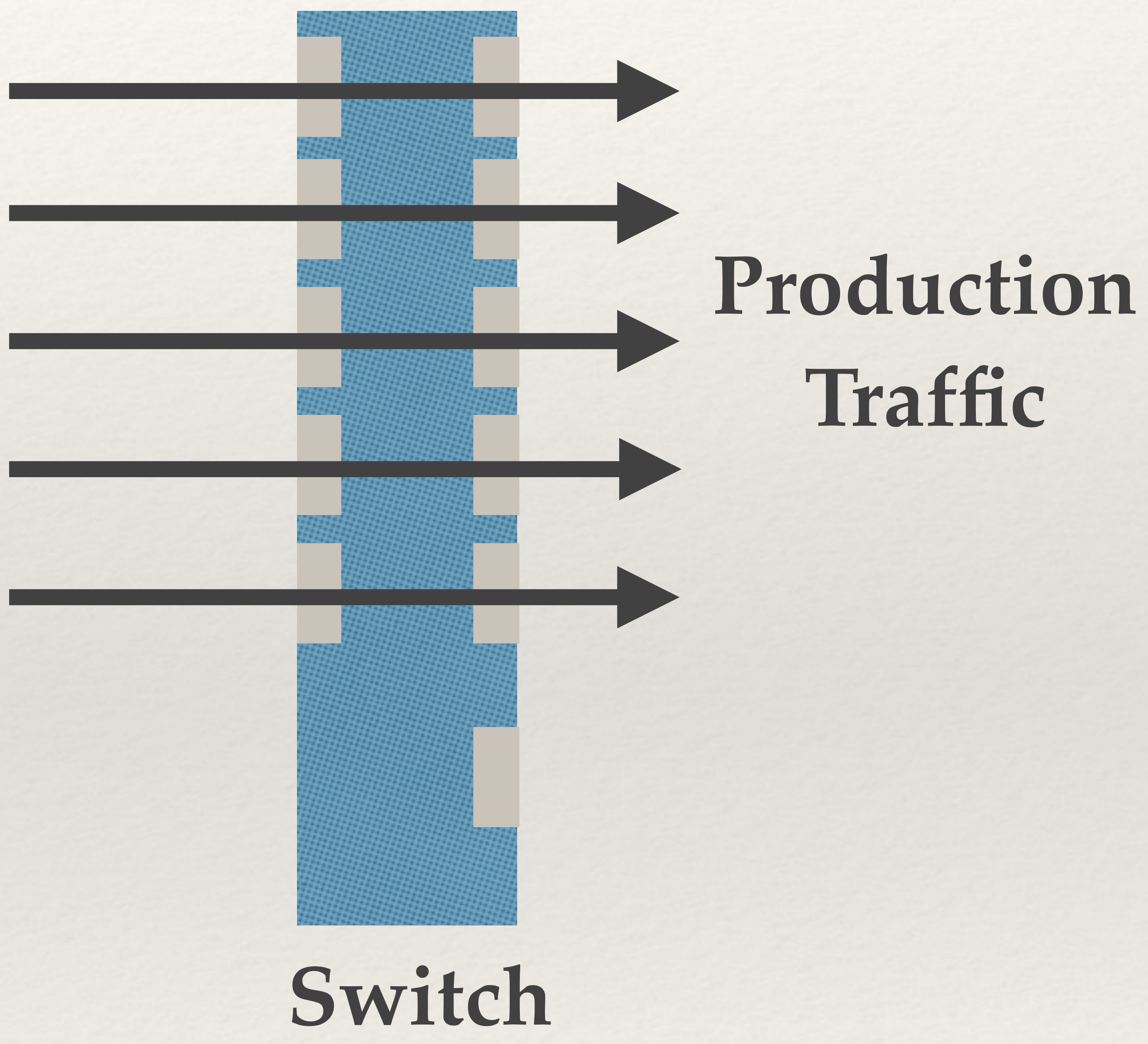
- Motivation
- Planck Architecture
- Is Planck Feasible?
- Is Planck Useful?
  - Microbenchmarks
  - Traffic Engineering

Architecture Goals

- Obtain very fast samples across all switches in the network
- Use those samples to infer global state of the network
  - Flow throughput
  - Flow paths
  - Port congestion state

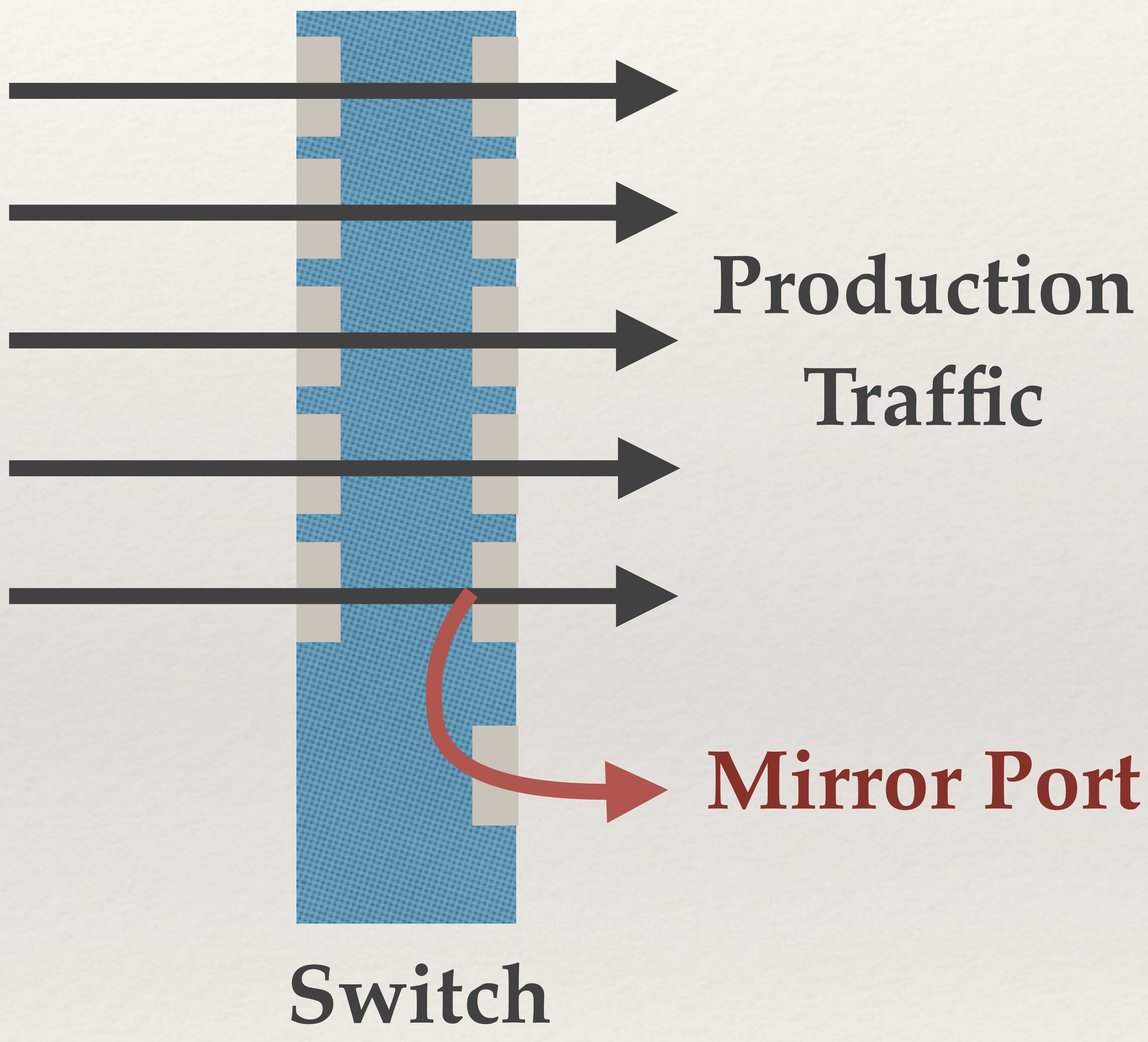
Our Solution: Repurpose Port Mirroring

- Modern switches support port-mirroring
  - Copies all packets e.g. going out a port to a designated mirror port
- Mirror all ports to a single mirror port
  - Intentional oversubscription
  - Drop behavior approximates sampling
  - Data-plane sampling much faster than control-plane based approaches



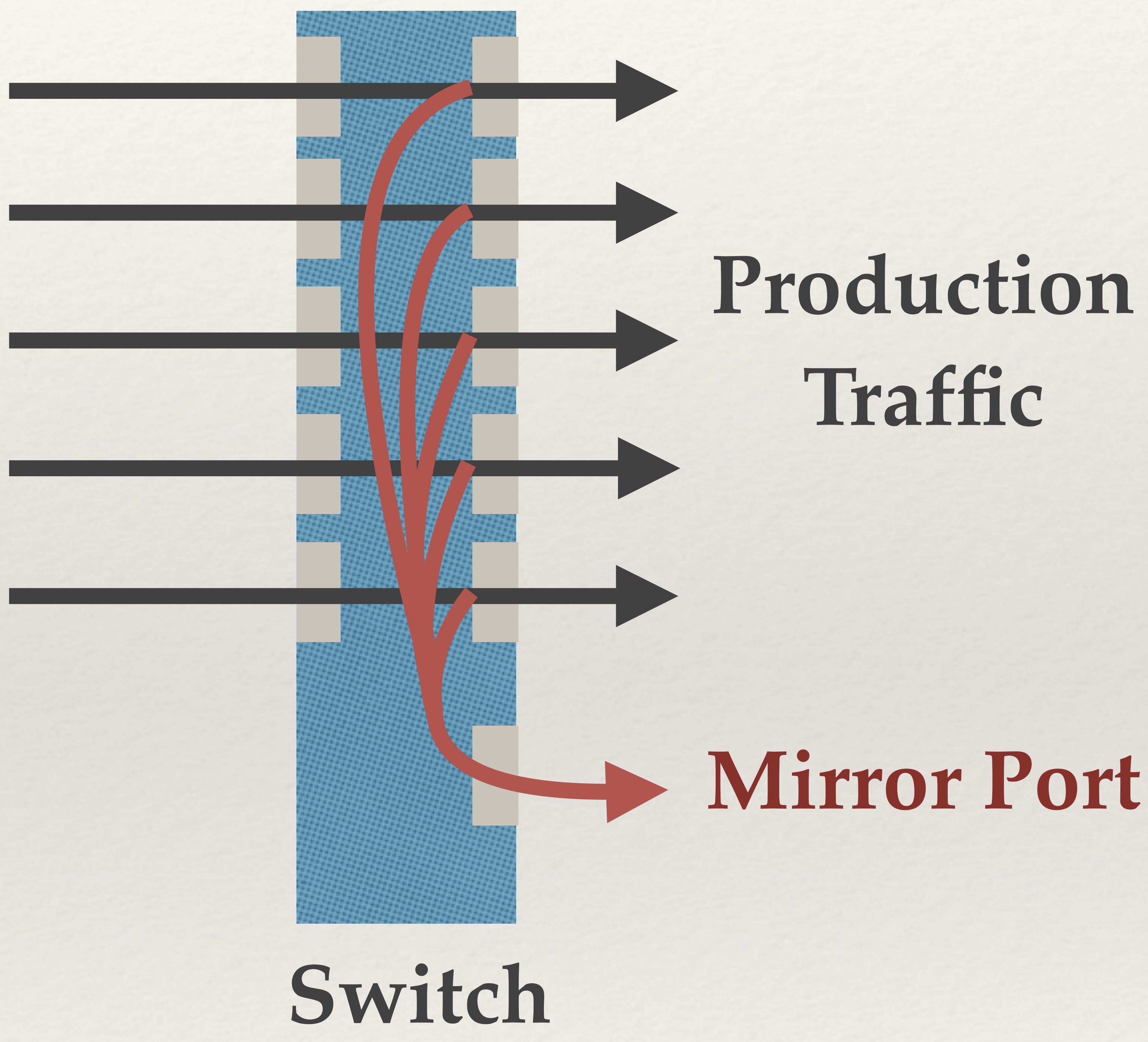
Our Solution: Repurpose Port Mirroring

- Modern switches support port-mirroring
  - Copies all packets e.g. going out a port to a designated mirror port
- Mirror all ports to a single mirror port
  - Intentional oversubscription
  - Drop behavior approximates sampling
  - Data-plane sampling much faster than control-plane based approaches



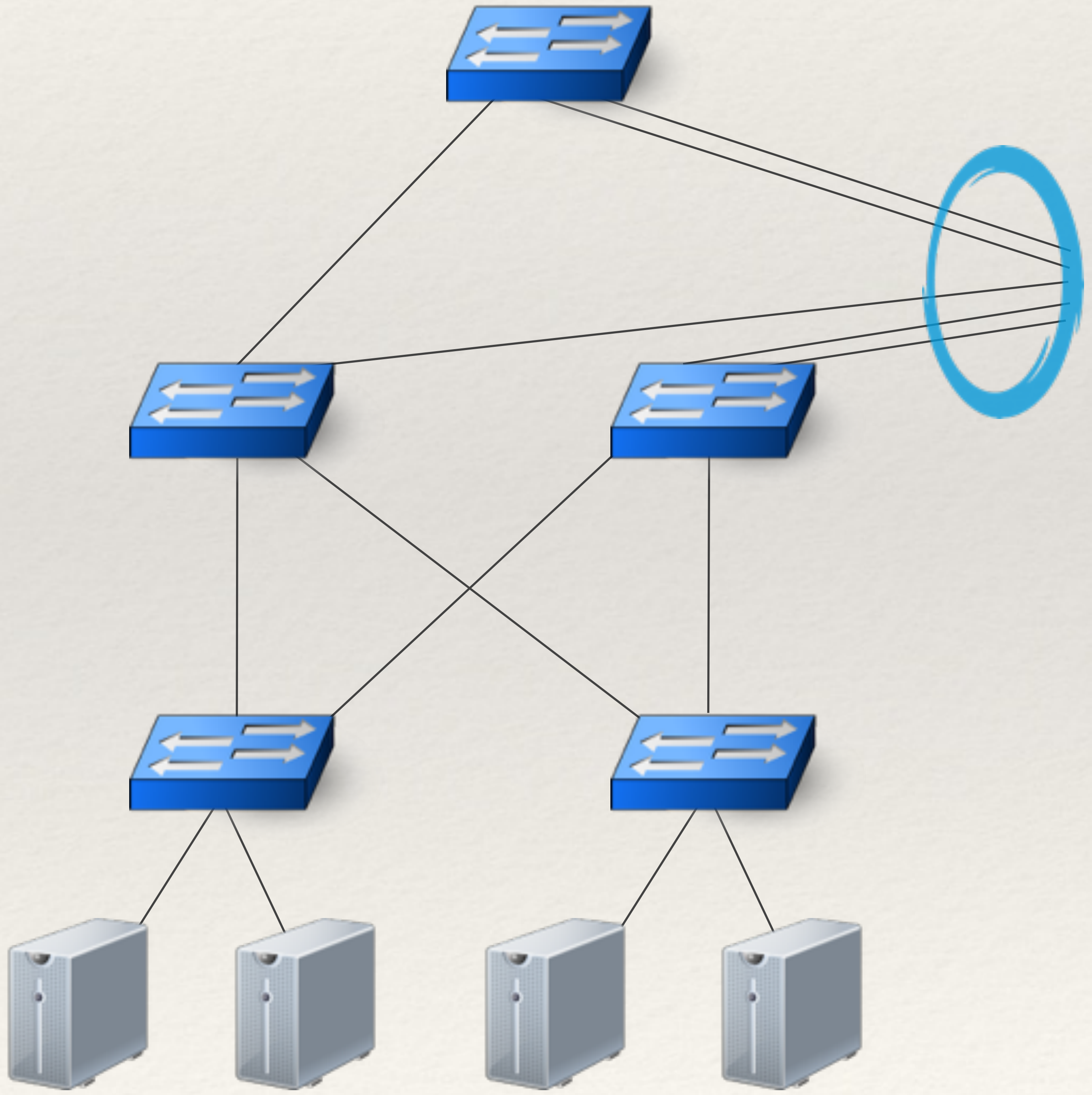
Our Solution: Repurpose Port Mirroring

- Modern switches support port-mirroring
  - Copies all packets e.g. going out a port to a designated mirror port
- Mirror all ports to a single mirror port
  - Intentional oversubscription
  - Drop behavior approximates sampling
  - Data-plane sampling much faster than control-plane based approaches



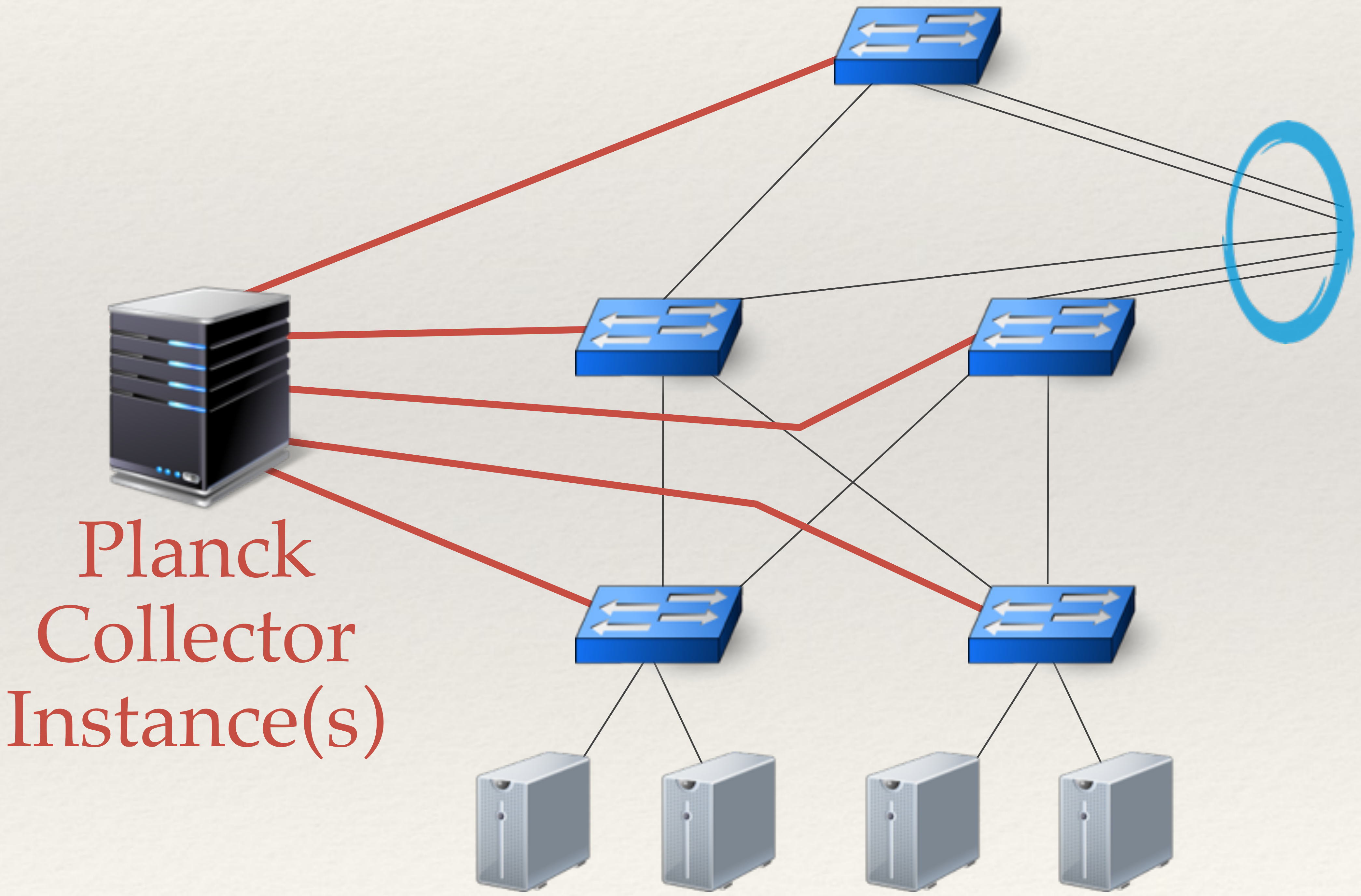
Planck Architecture

- Oversubscribed port-mirroring as a primitive
- Collectors receive samples from mirror ports
  - Netmap for fast processing
- Reconstruct flow information across all flows in the network
- Collectors can interact with an SDN controller to implement various applications



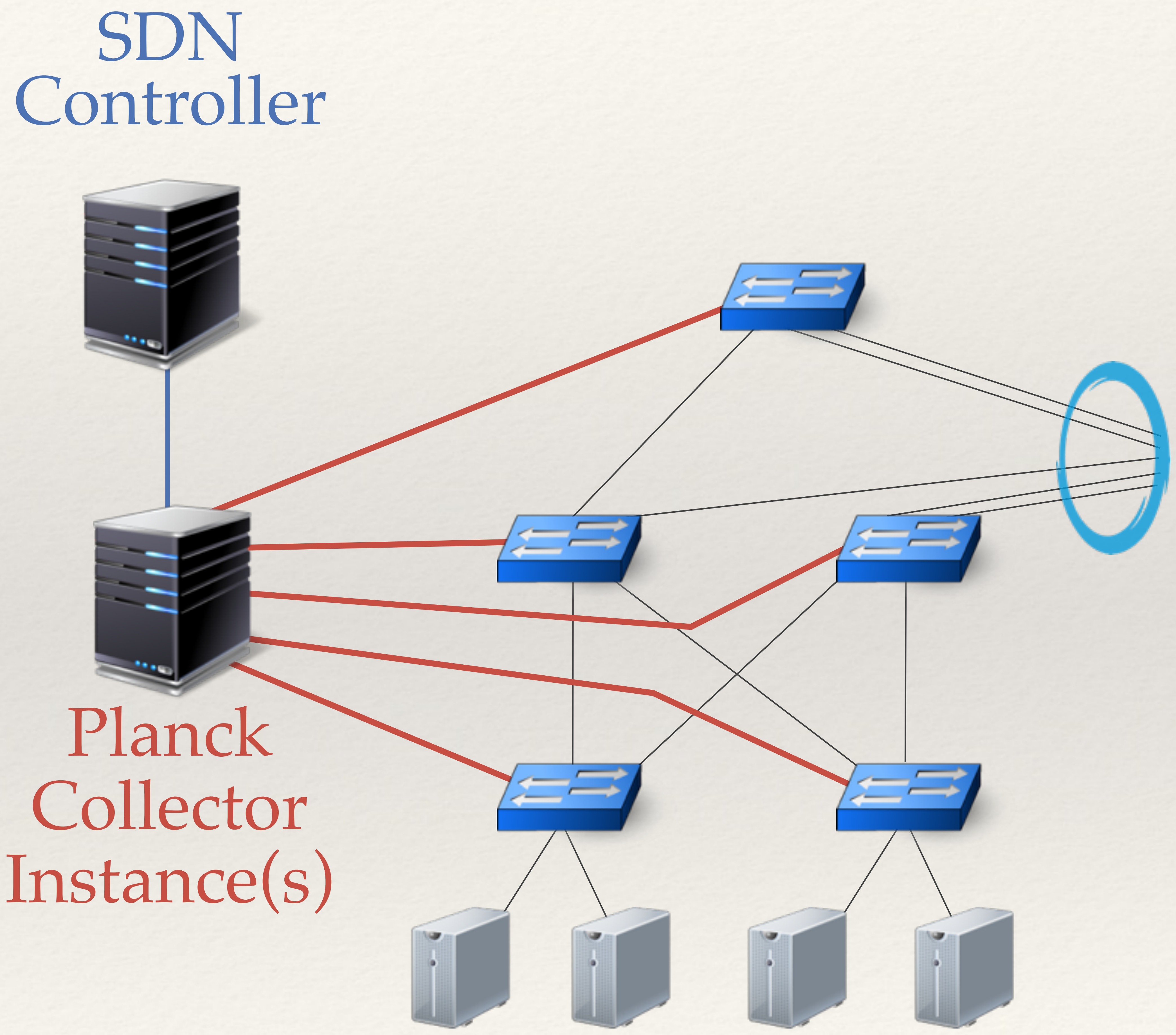
Planck Architecture

- Oversubscribed port-mirroring as a primitive
- Collectors receive samples from mirror ports
  - Netmap for fast processing
- Reconstruct flow information across all flows in the network
- Collectors can interact with an SDN controller to implement various applications



Planck Architecture

- Oversubscribed port-mirroring as a primitive
- Collectors receive samples from mirror ports
  - Netmap for fast processing
- Reconstruct flow information across all flows in the network
- Collectors can interact with an SDN controller to implement various applications

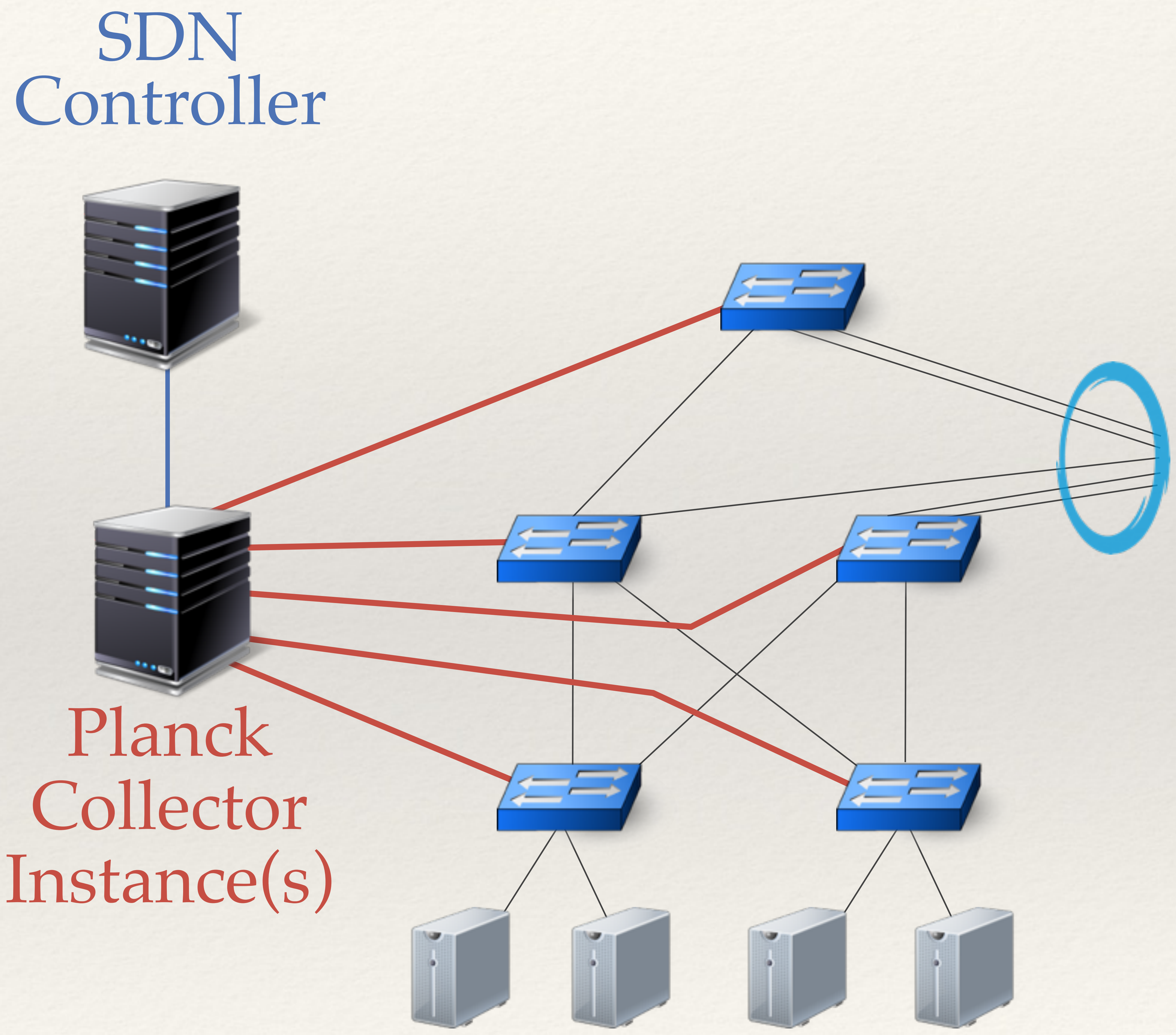


Outline

- Motivation
- Planck Architecture
- Is Planck Feasible?
- Is Planck Useful?
  - Microbenchmarks
  - Traffic Engineering

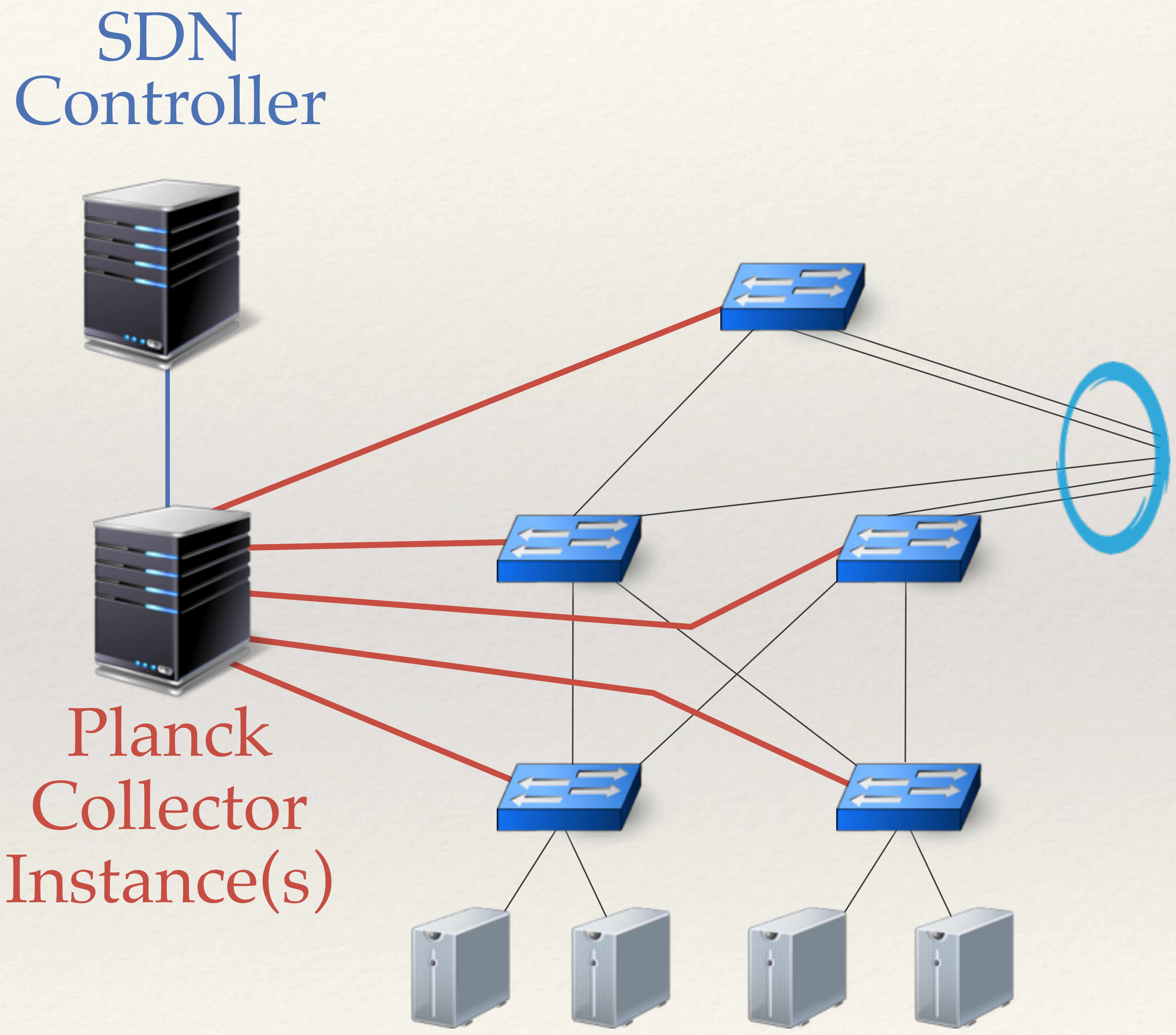
Is Planck Feasible?

- Does Planck hurt production traffic?
- Can Planck infer throughput?
- Can Planck infer congested ports?



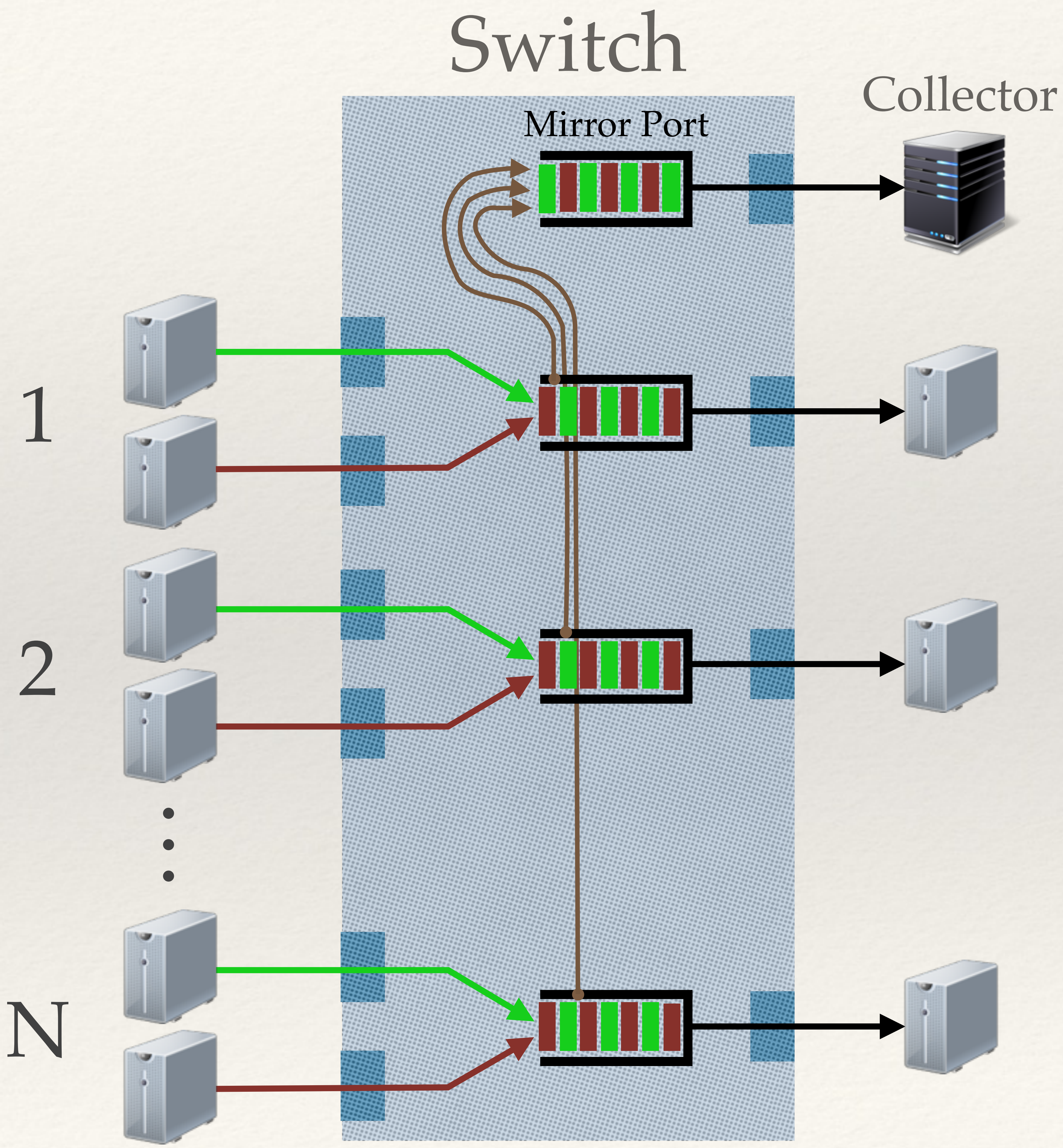
Is Planck Feasible?

- Does Planck hurt production traffic?
- Can Planck infer throughput?
- Can Planck infer congested ports?



Experiment Setup

- Vary the number of congested ports
- N number of: 2 senders paired with 1 receiver
  - TCP will fill up the output buffer going to the receiver
- 15 trials for each config with/without Planck-mirroring
  - Monitor latency, packet loss and throughput

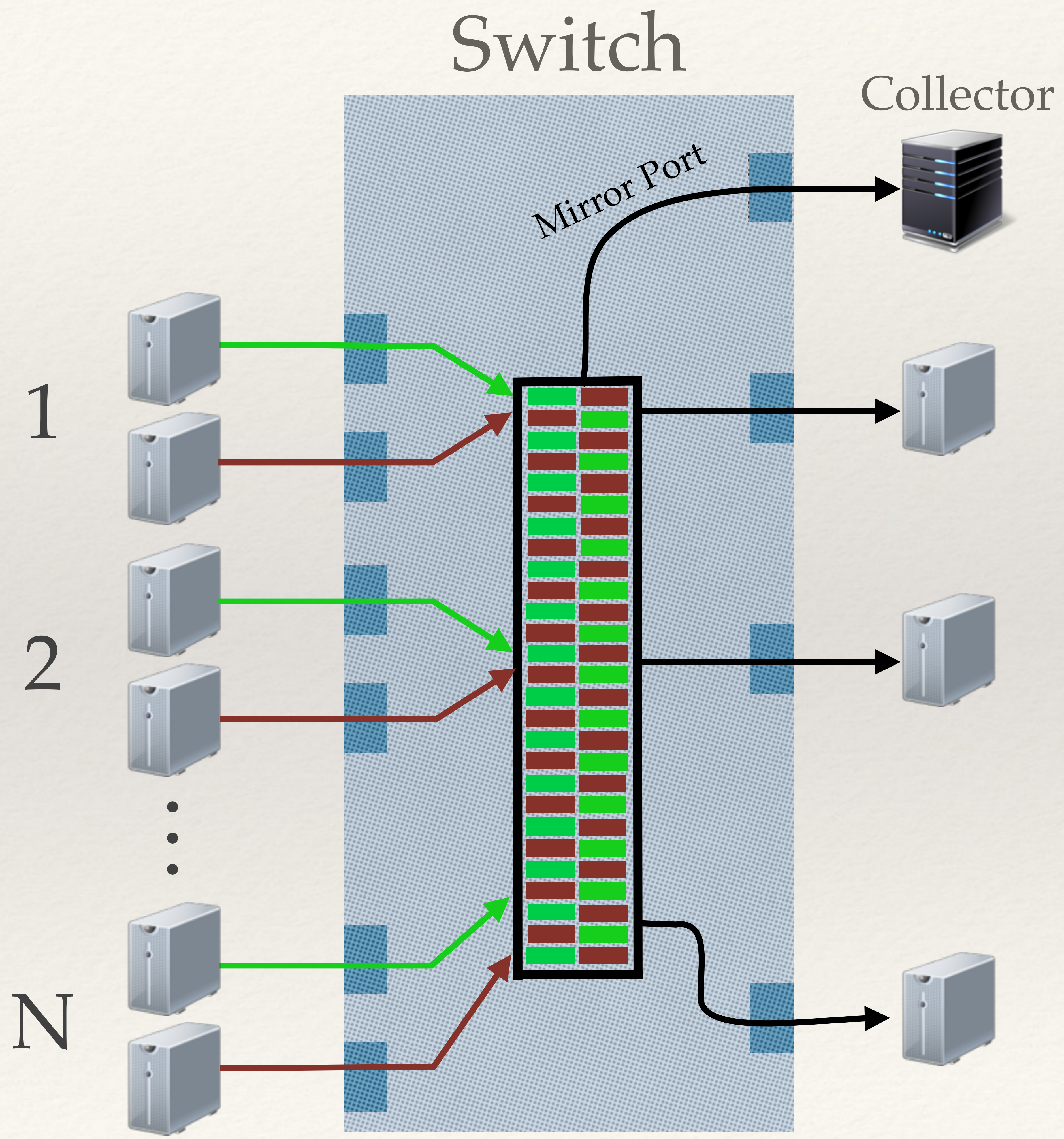


Switches Share Buffers

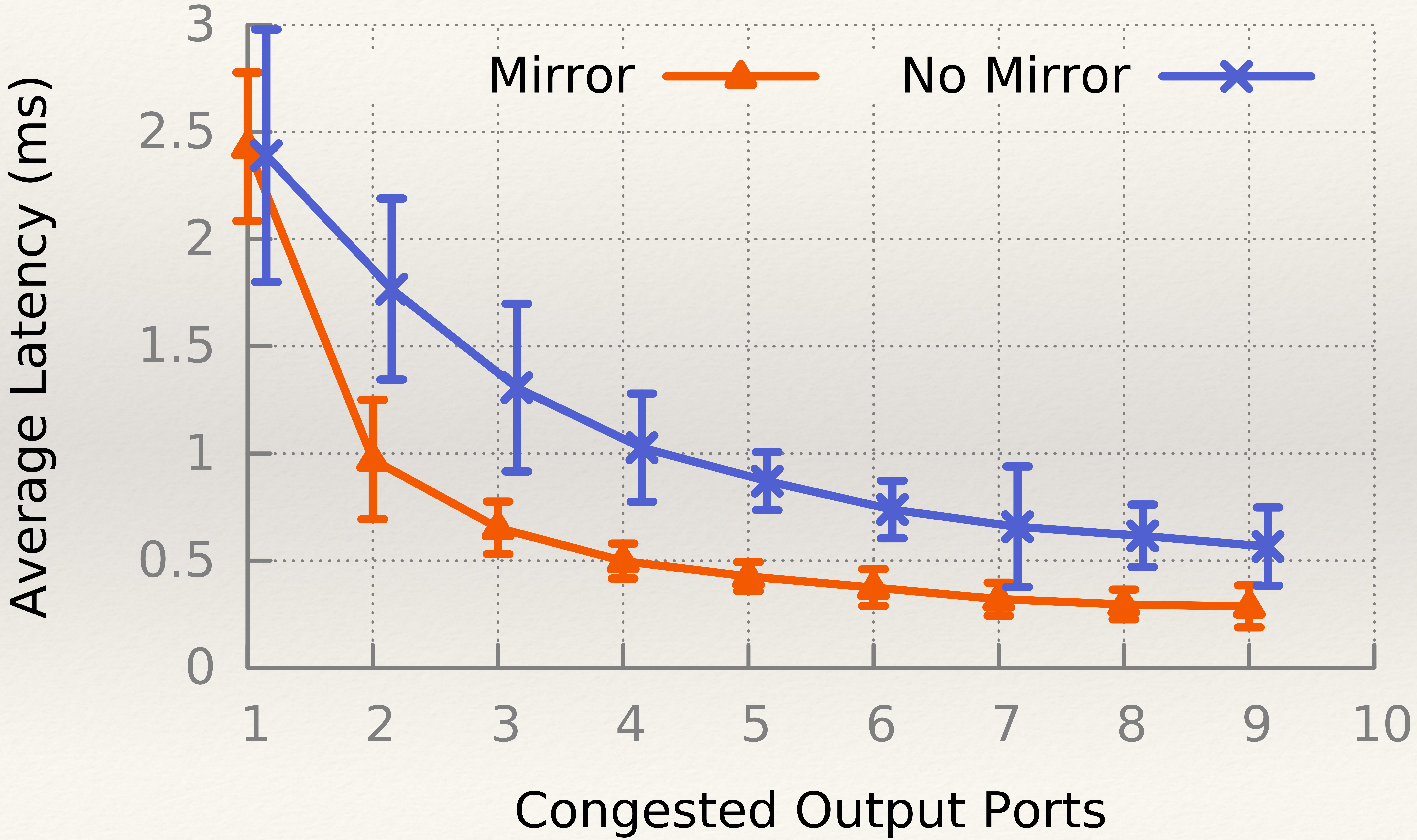
- Modern switches use shared buffers
- Independent queues per ports is not completely accurate
- Memory consumed by an oversubscribed mirror port may use space that production traffic could use

Switches Share Buffers

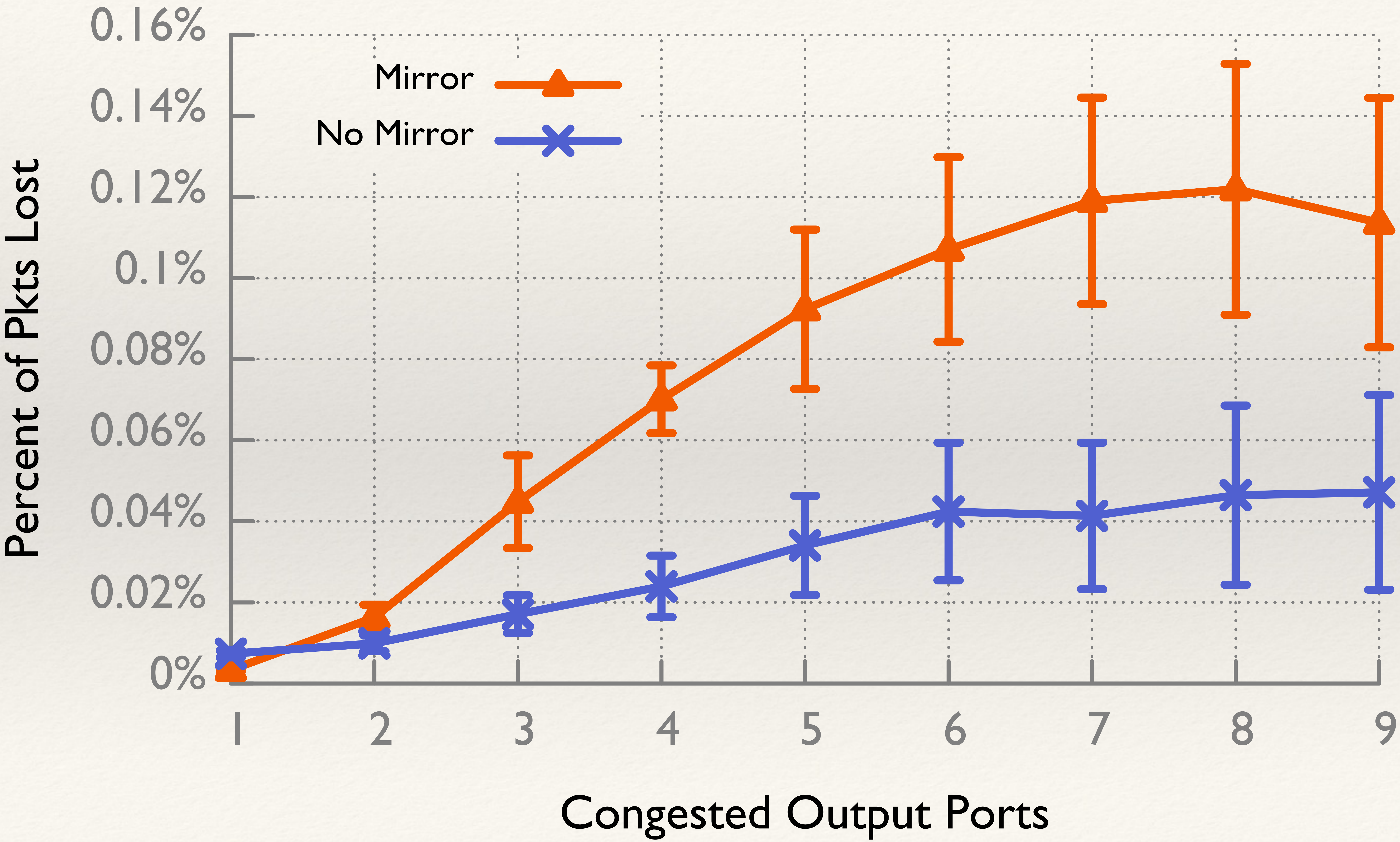
- Modern switches use shared buffers
- Independent queues per ports is not completely accurate
- Memory consumed by an oversubscribed mirror port may use space that production traffic could use



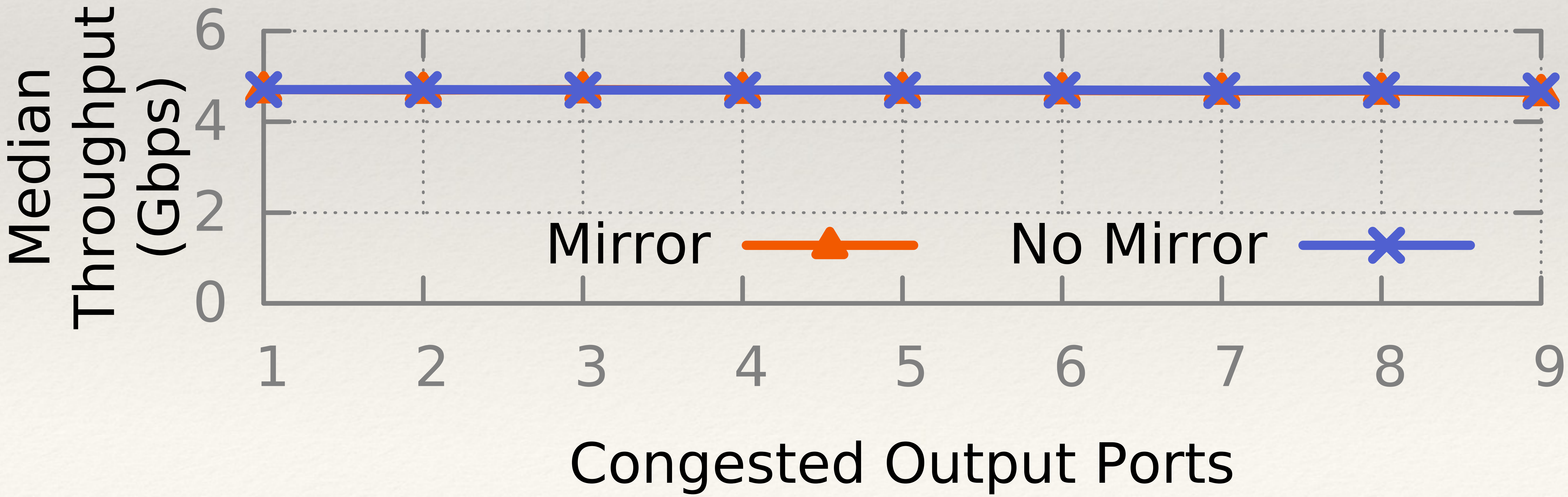
Production Traffic Latency



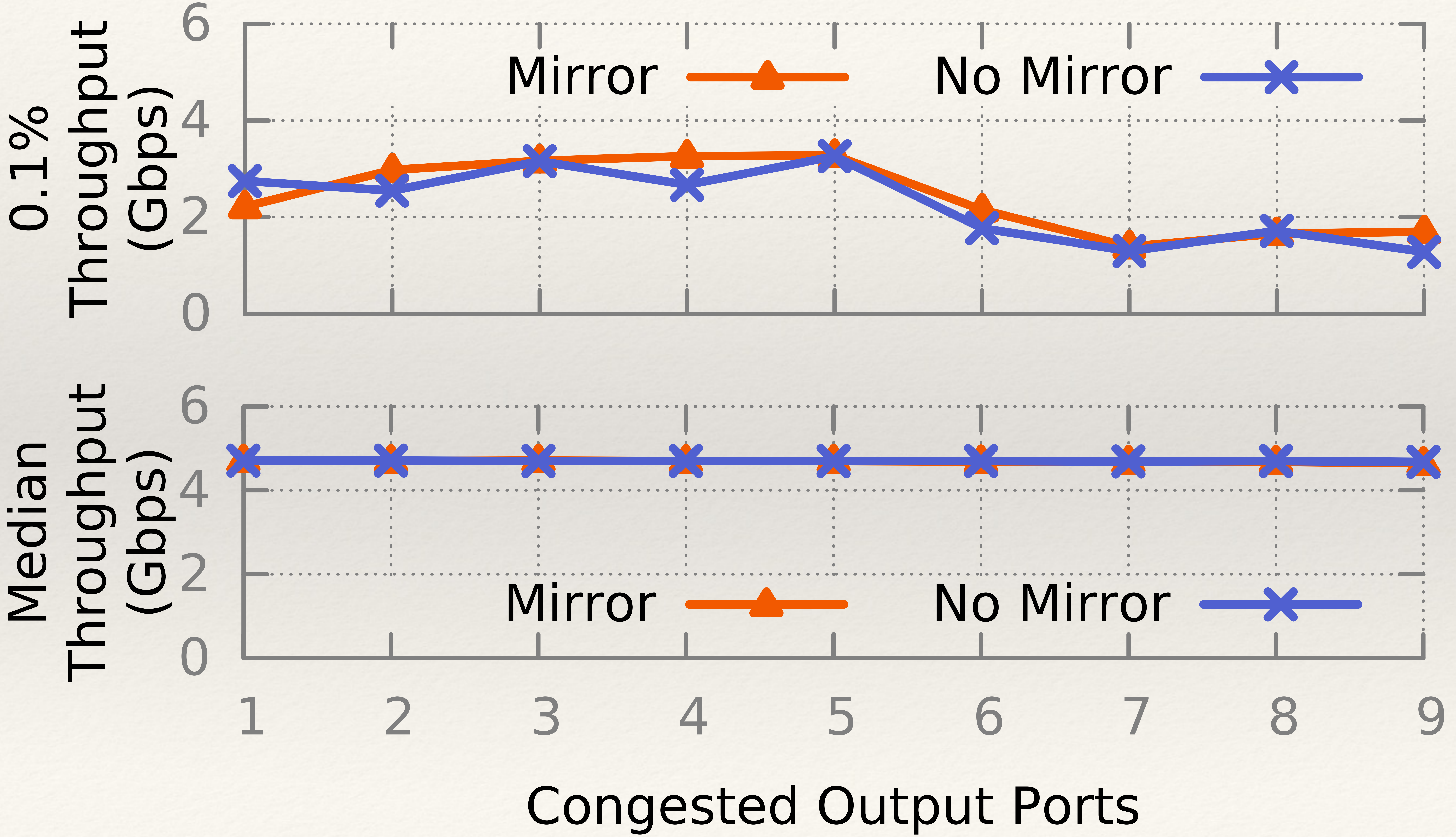
Production Traffic Packet Loss



Production Throughput

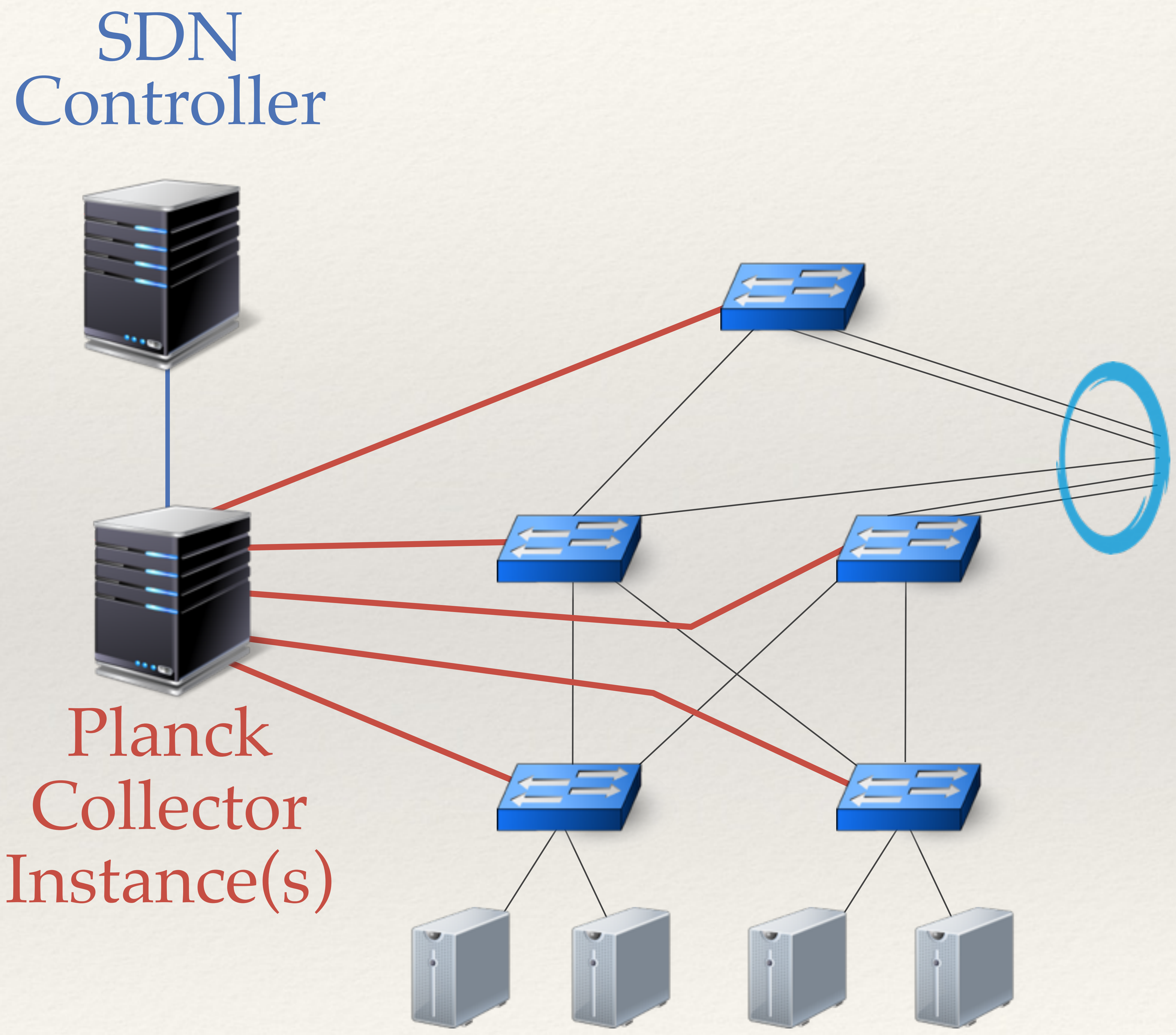


Production Throughput

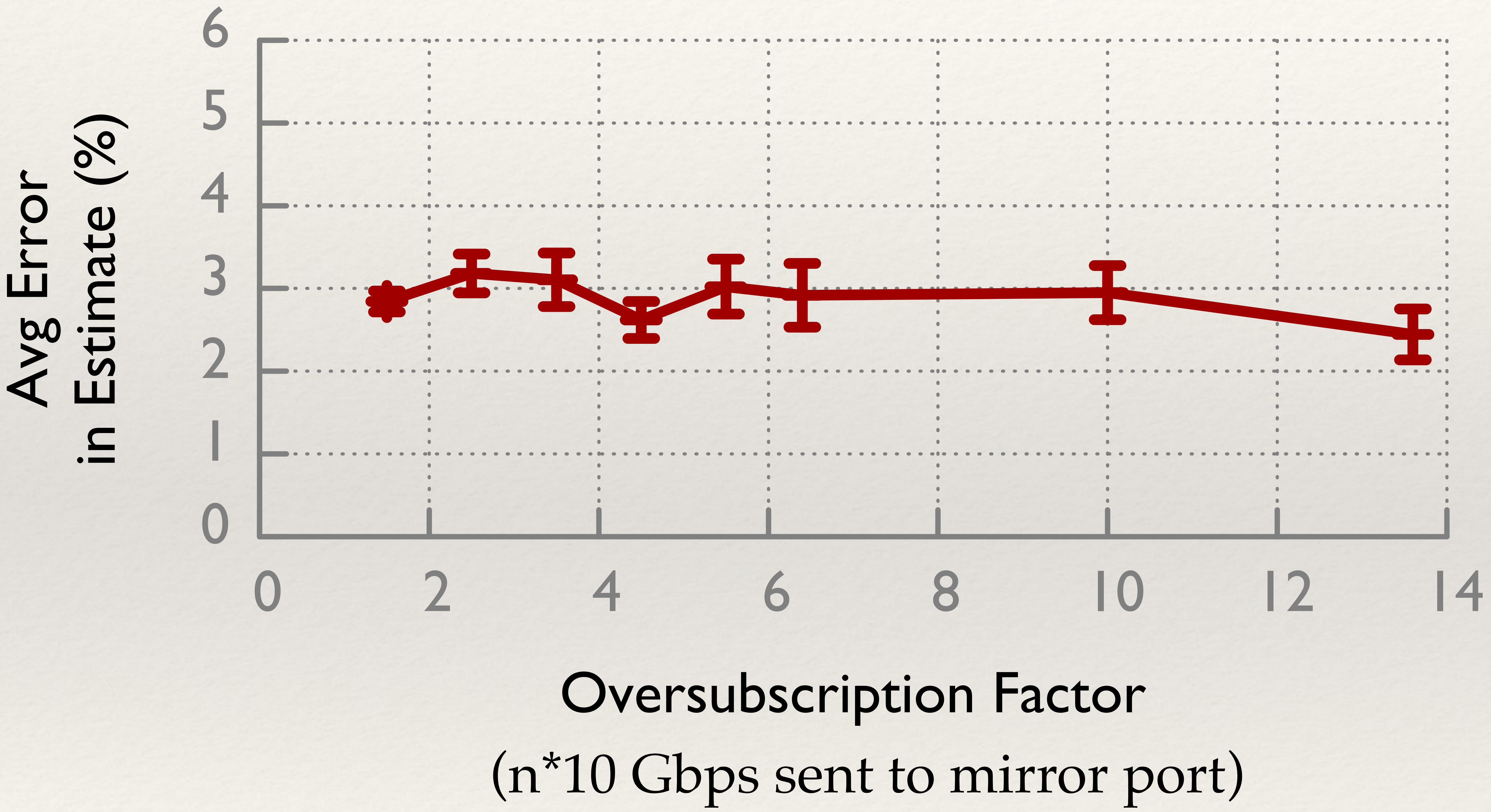


Is Planck Feasible?

- Does Planck hurt production traffic?
- Can Planck infer throughput?
- Can Planck infer congested ports?

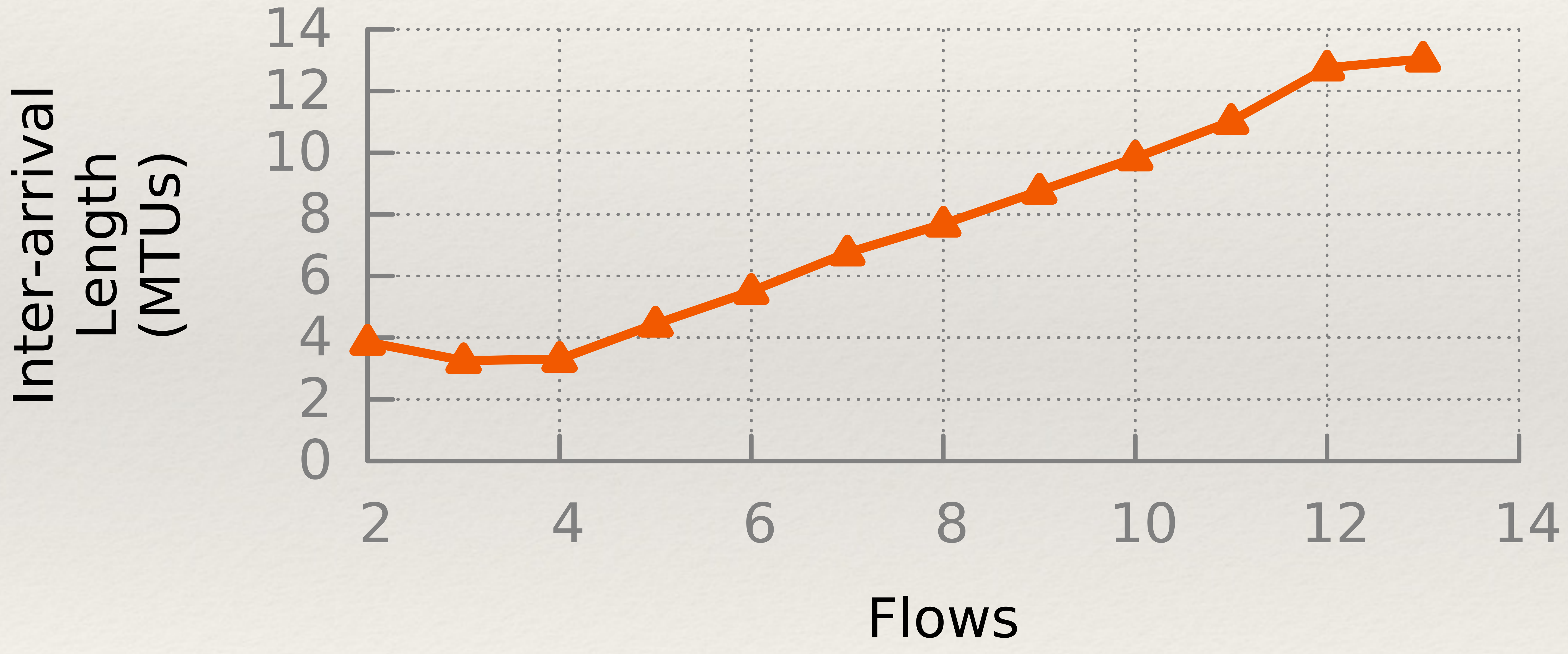


Throughput Estimation is Accurate



- Estimates are trivial if sampling rate is known
- Leverage TCP seq# in packets
- Smoothed estimates in 200–700 μs
- See paper for more details

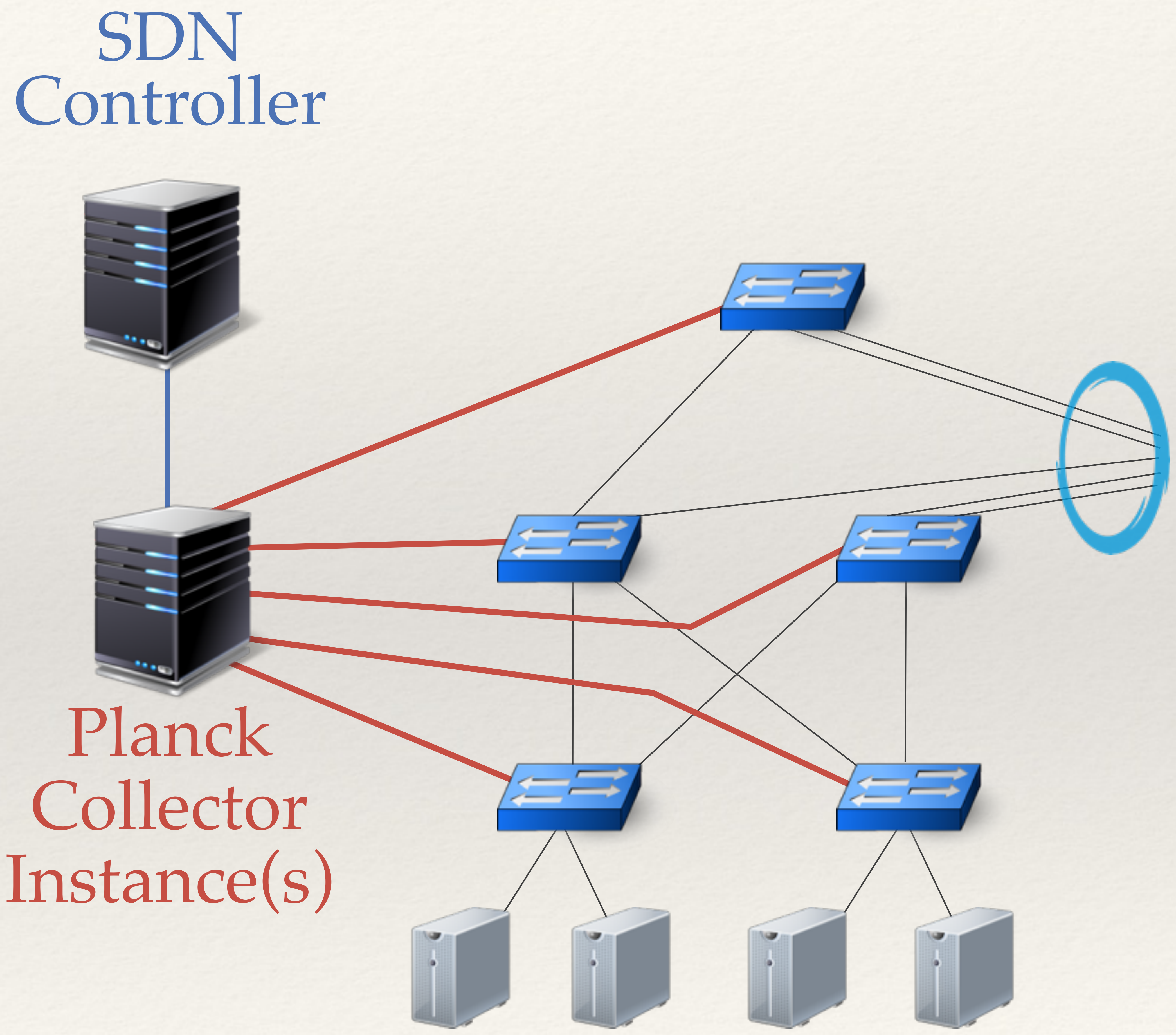
Sample Inter-Arrival Length



- x13 10 Gbps flows
- Grows roughly linearly
- See paper for further results

Is Planck Feasible?

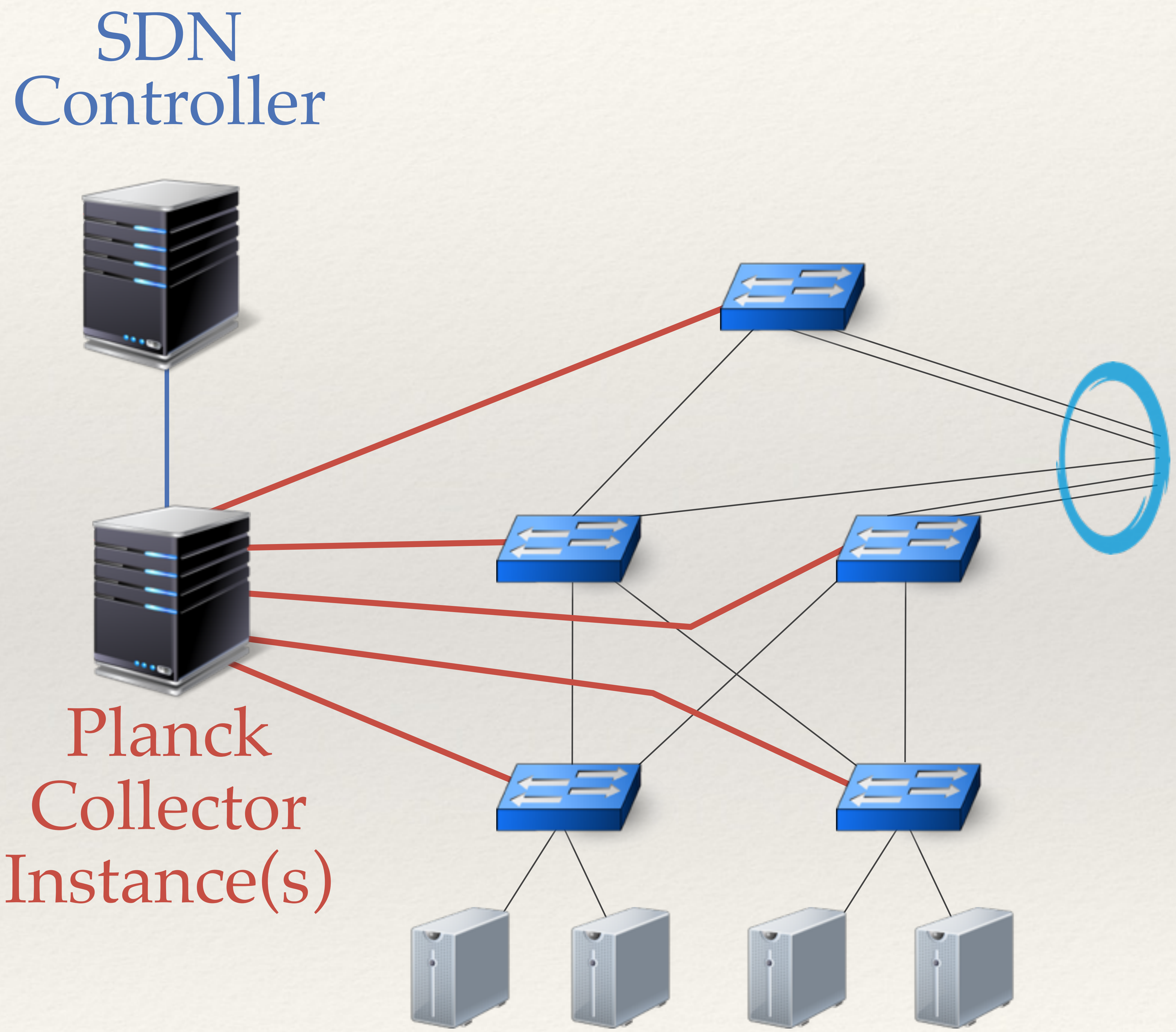
- Does Planck hurt production traffic?
- Can Planck infer throughput?
- Can Planck infer congested ports?



Is Planck Feasible?

- Does Planck hurt production traffic?
- Can Planck infer throughput?
- Can Planck infer congested ports?

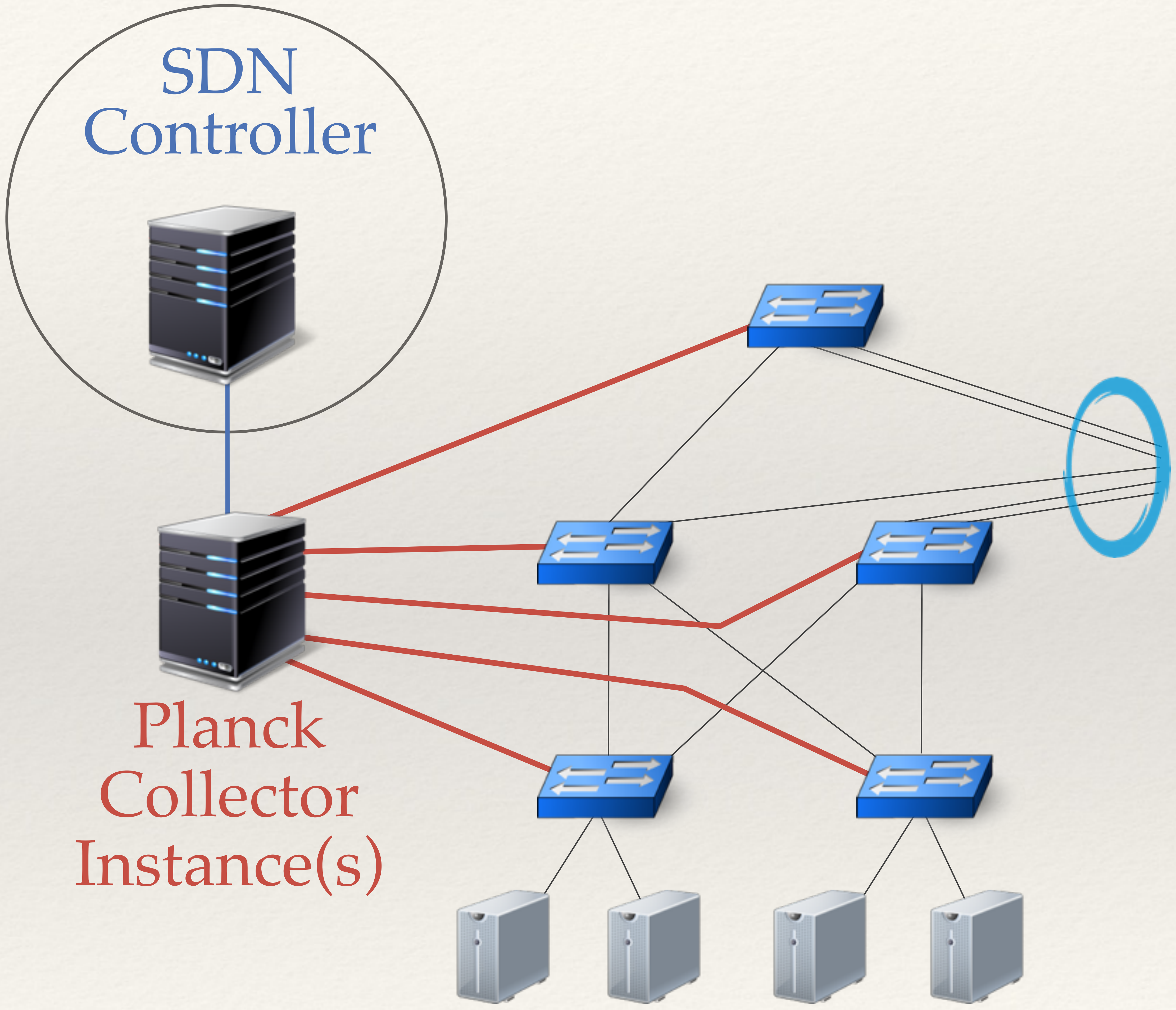
Planck provides flow throughput, a flow just needs to be mapped to a port



Is Planck Feasible?

- Does Planck hurt production traffic?
- Can Planck infer throughput?
- Can Planck infer congested ports?

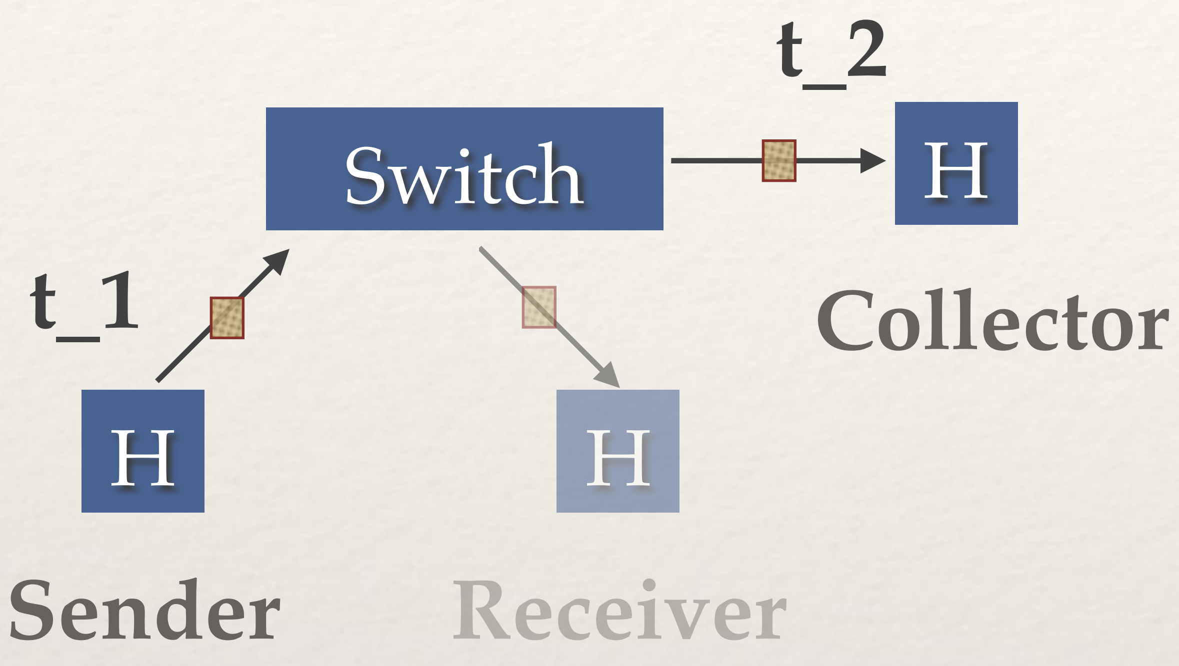
Planck provides flow throughput, a flow just needs to be mapped to a port



Outline

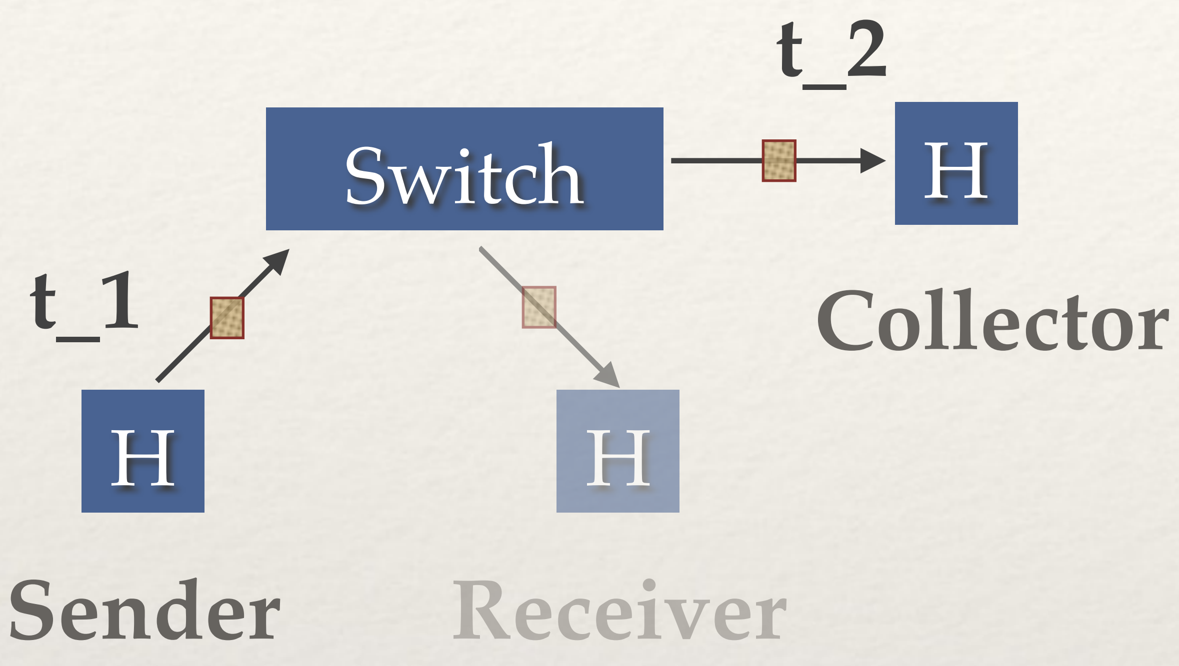
- Motivation
- Planck Architecture
- Is Planck Feasible?
- Is Planck Useful?
    - Microbenchmarks
    - Traffic Engineering

Sample Latency



Latency = t_2 - t_1

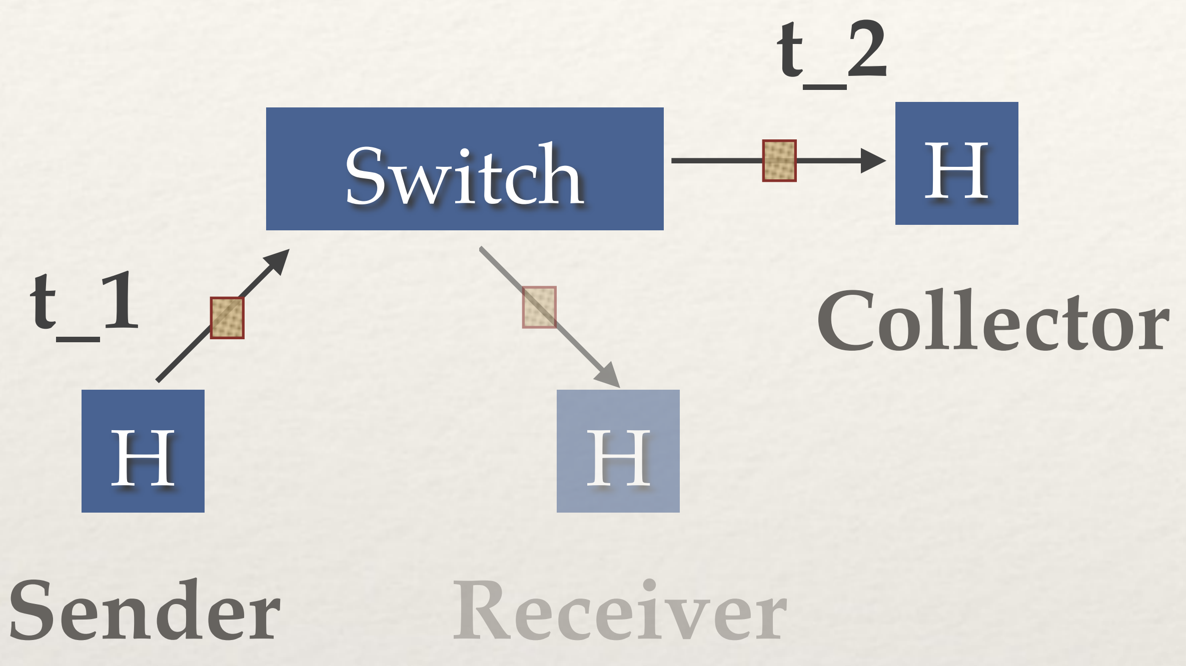
Sample Latency



Latency = t_2 - t_1

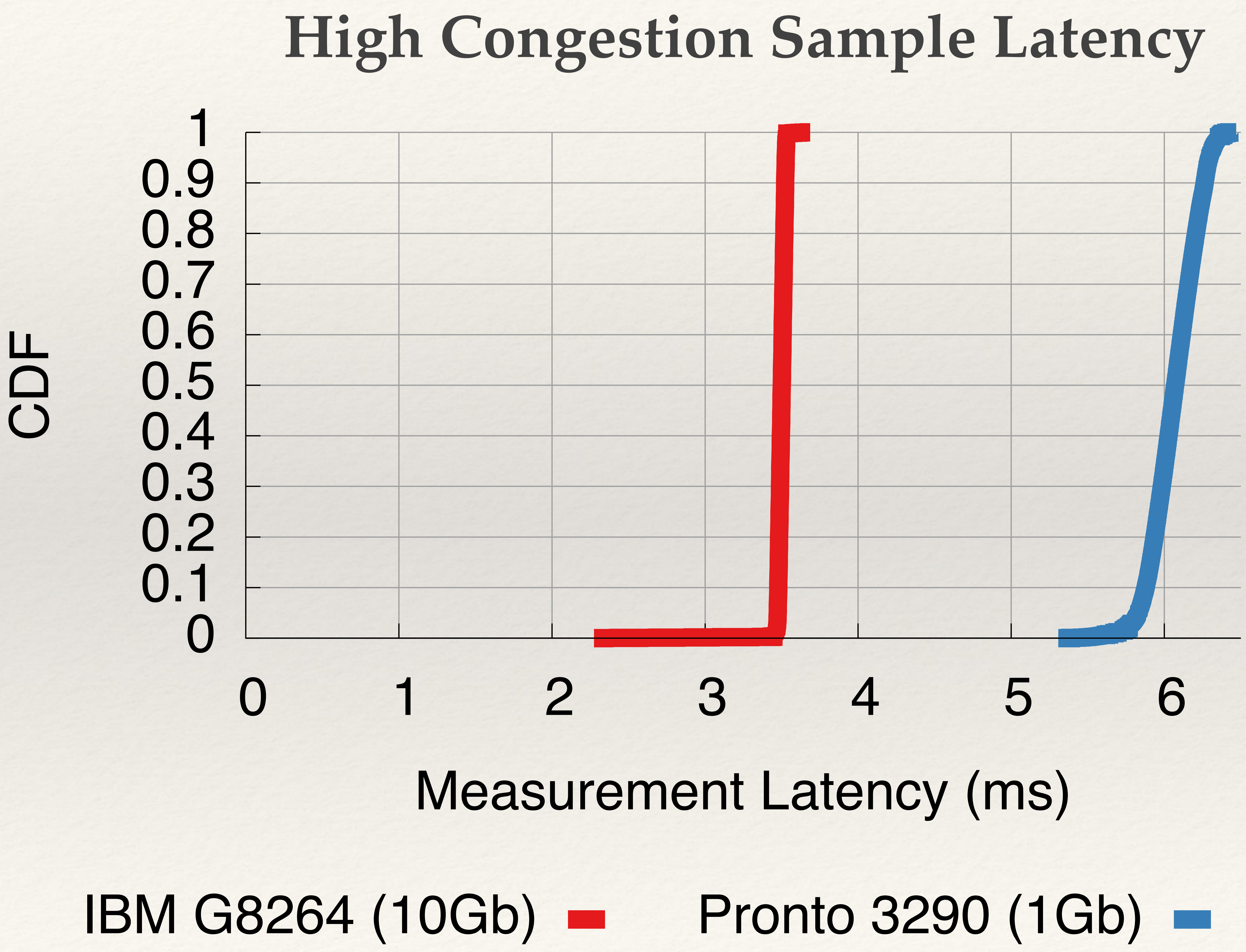
Low Congestion Sample Latency: 75–150 μs

Sample Latency



Latency = t_2 - t_1

Low Congestion Sample Latency: 75–150 μs



Control Loop Times

- < 3.5 ms to obtain sample + < 700μs to get tput estimate = **4.2 ms worst-case measurement time for 10 Gb**
- Planck achieves measurement speeds 18x – 291x faster than recent approaches
- Shadow MAC addresses [1] and some ARP tricks allow re-routing at < 3ms
- See paper for more details

[1] Shadow MACs: Scalable Label-switching for Commodity Ethernet (HotSDN '14)

Control Loop Times

- < 3.5 ms to obtain sample + < 700μs to get tput estimate = **4.2 ms worst-case measurement time for 10 Gb**
- Planck achieves measurement speeds 18x – 291x faster than recent approaches
- Shadow MAC addresses [1] and some ARP tricks allow re-routing at < 3ms
- See paper for more details

Planck: < 4.2 ms

~~**100 ms – 1 sec+**~~

Measurement

> 10 ms

Control

~100 μs

Decision

[1] Shadow MACs: Scalable Label-switching for Commodity Ethernet (HotSDN '14)

Control Loop Times

- < 3.5 ms to obtain sample + < 700μs to get tput estimate = **4.2 ms worst-case measurement time for 10 Gb**
- Planck achieves measurement speeds 18x – 291x faster than recent approaches
- Shadow MAC addresses [1] and some ARP tricks allow re-routing at < 3ms
- See paper for more details

Planck: < 4.2 ms

~~100 ms – 1 sec+~~

Measurement

< 3 ms

~~> 10 ms~~

Control

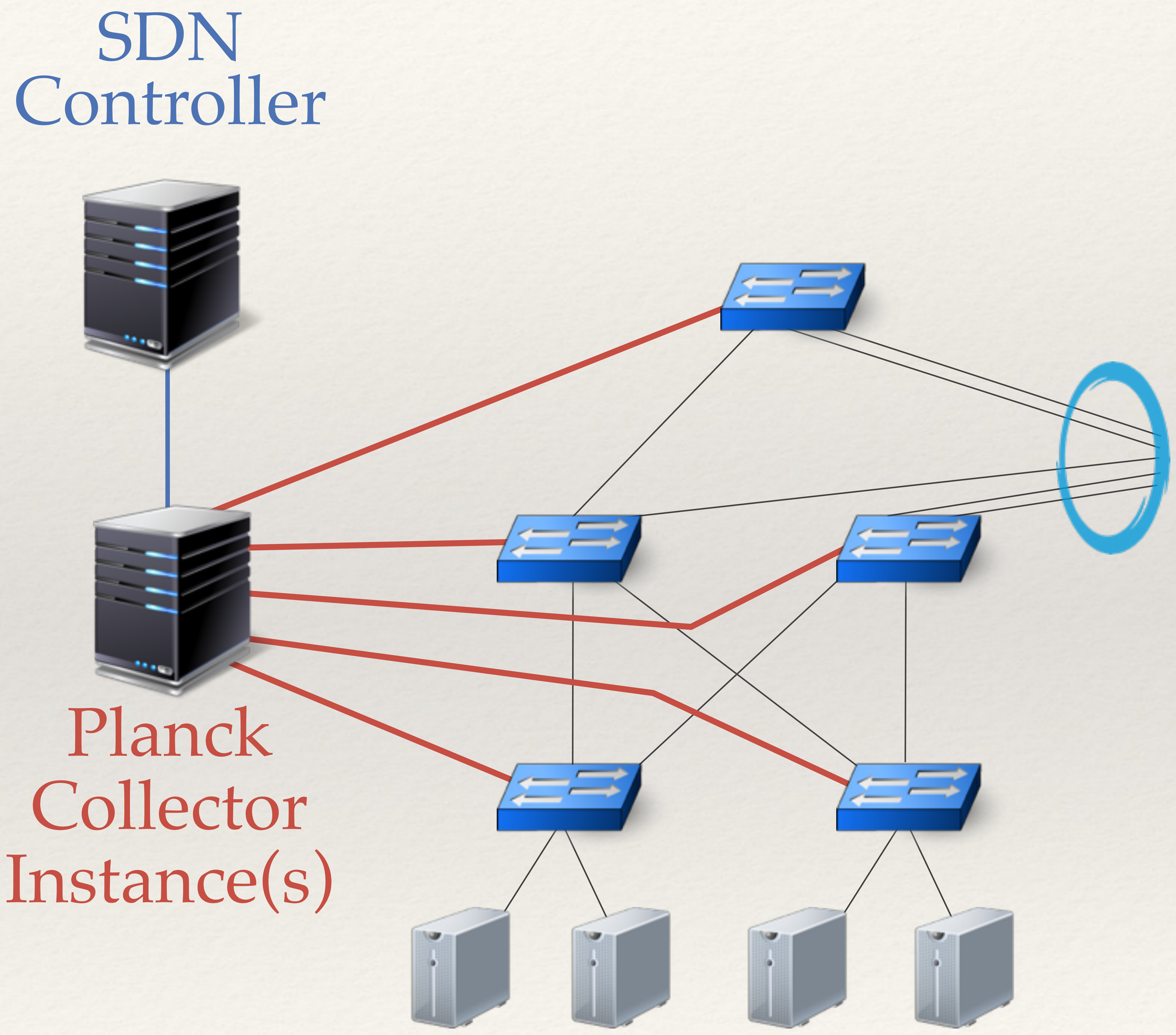
~100 μs

Decision

[1] Shadow MACs: Scalable Label-switching for Commodity Ethernet (HotSDN '14)

Planck as a Platform

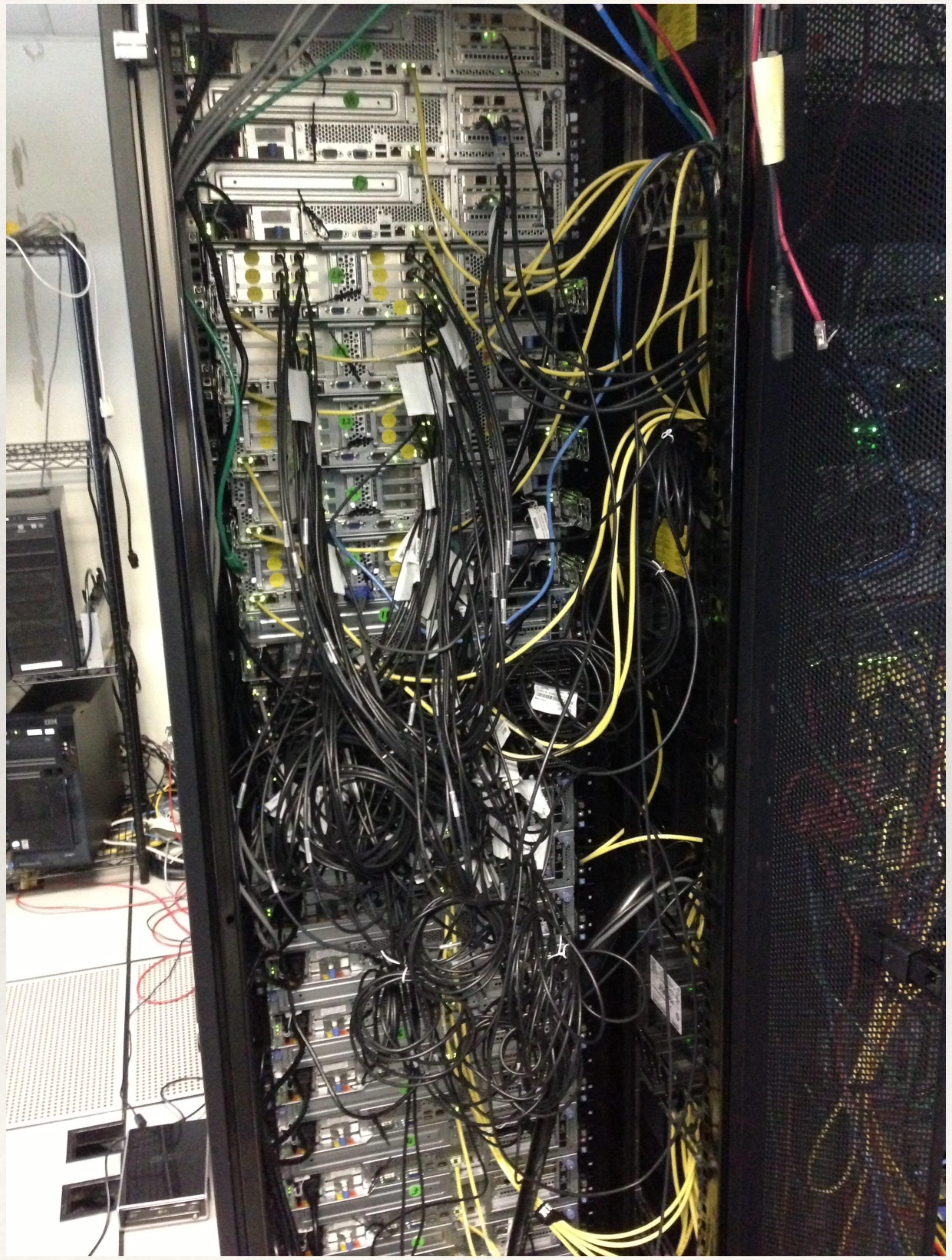
- Vantage point mirroring
  - tcpdump for switches
- Global view of the network
  - flow data across all links
- Traffic engineering
  - congested port notifications



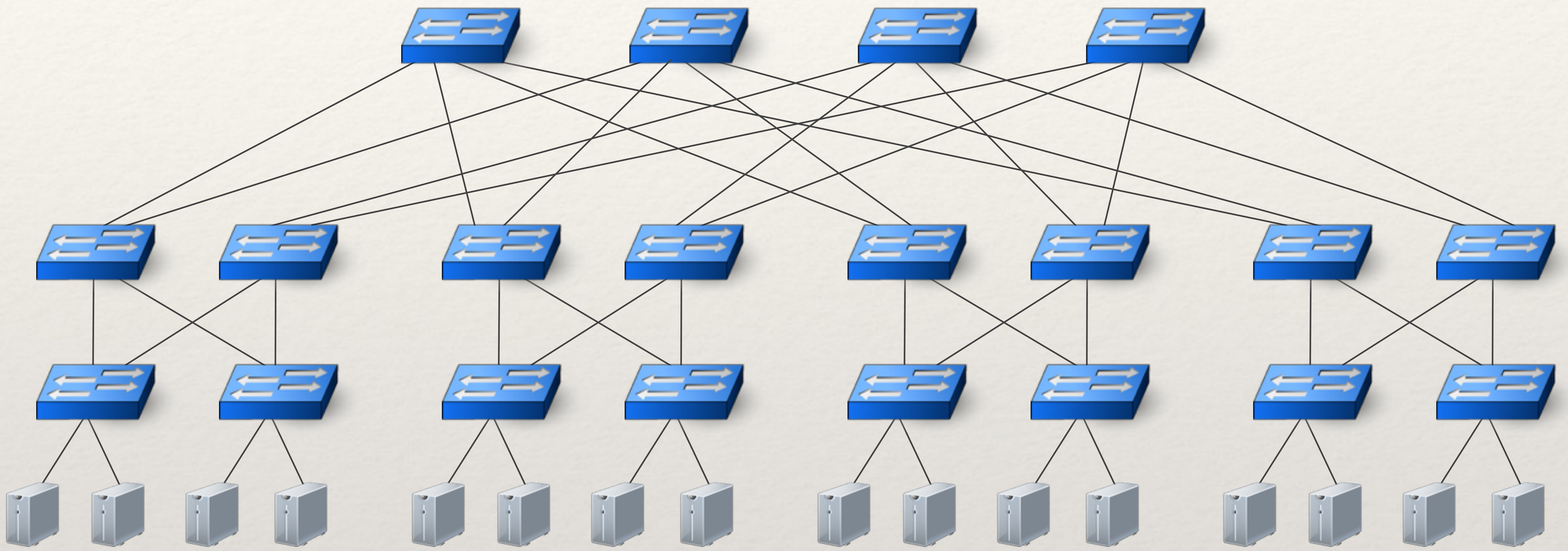
Outline

- Motivation
- Planck Architecture
- Is Planck Feasible?
- Is Planck Useful?
  - Microbenchmarks
  - Traffic Engineering

Testbed



16 Host Fat Tree



- Split x4 IBM G8264 (48-port) switches into 20 sub-switches
    - Routing via Floodlight plugin inspired by FlowVisor
- x3 server machines with x8 10 GbE NICs each
- x16 machines with x2 10 GbE NICs

Methodology

Traffic Engineering

- Floodlight-based module using Planck
- Collectors notify a controller when ports become congested

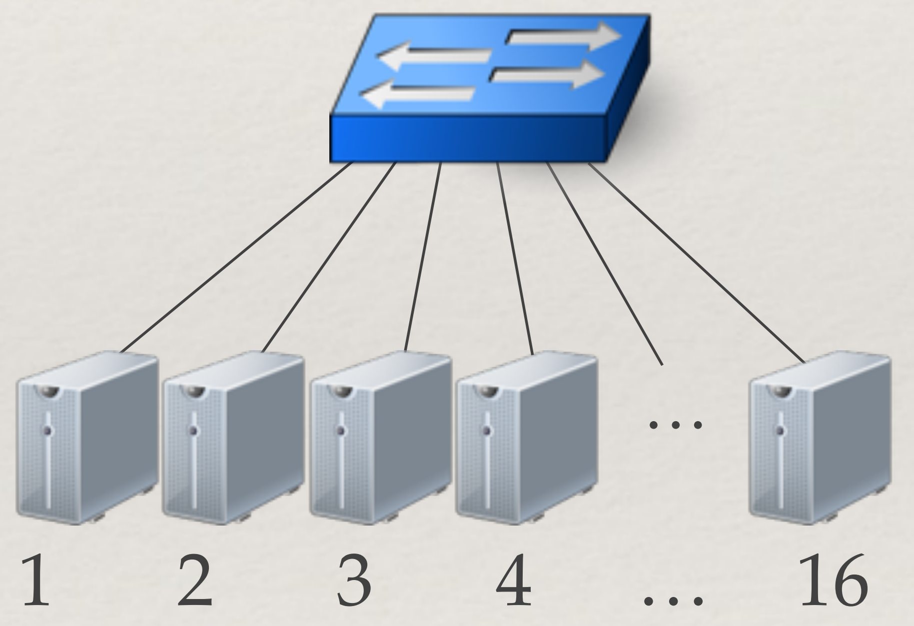
Workloads

- Shuffle
- Stride
- Random
- Random Bijection

Routing

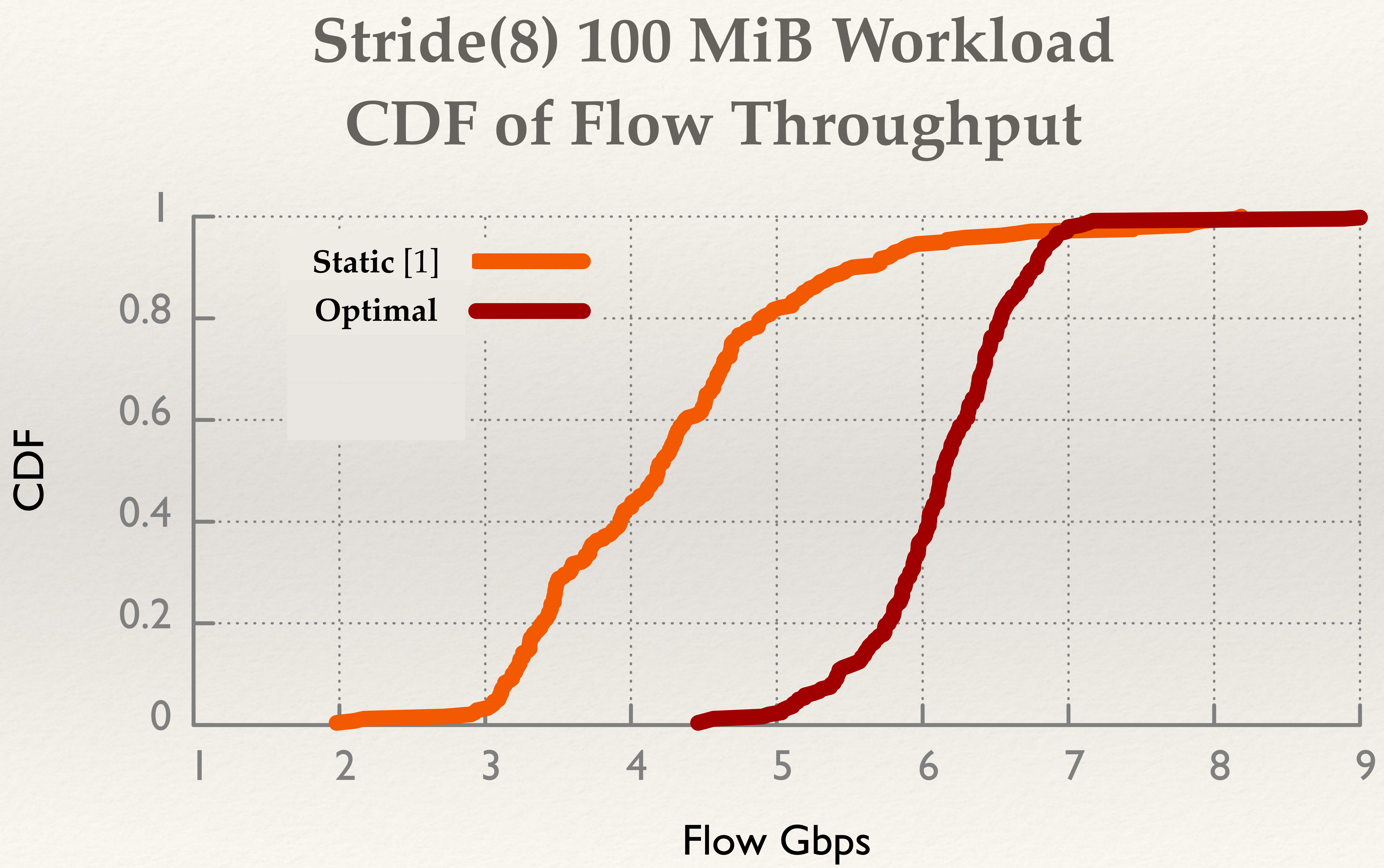
- Static [1]
- Poll-100 ms
- PlanckTE
- Optimal

Optimal Topology

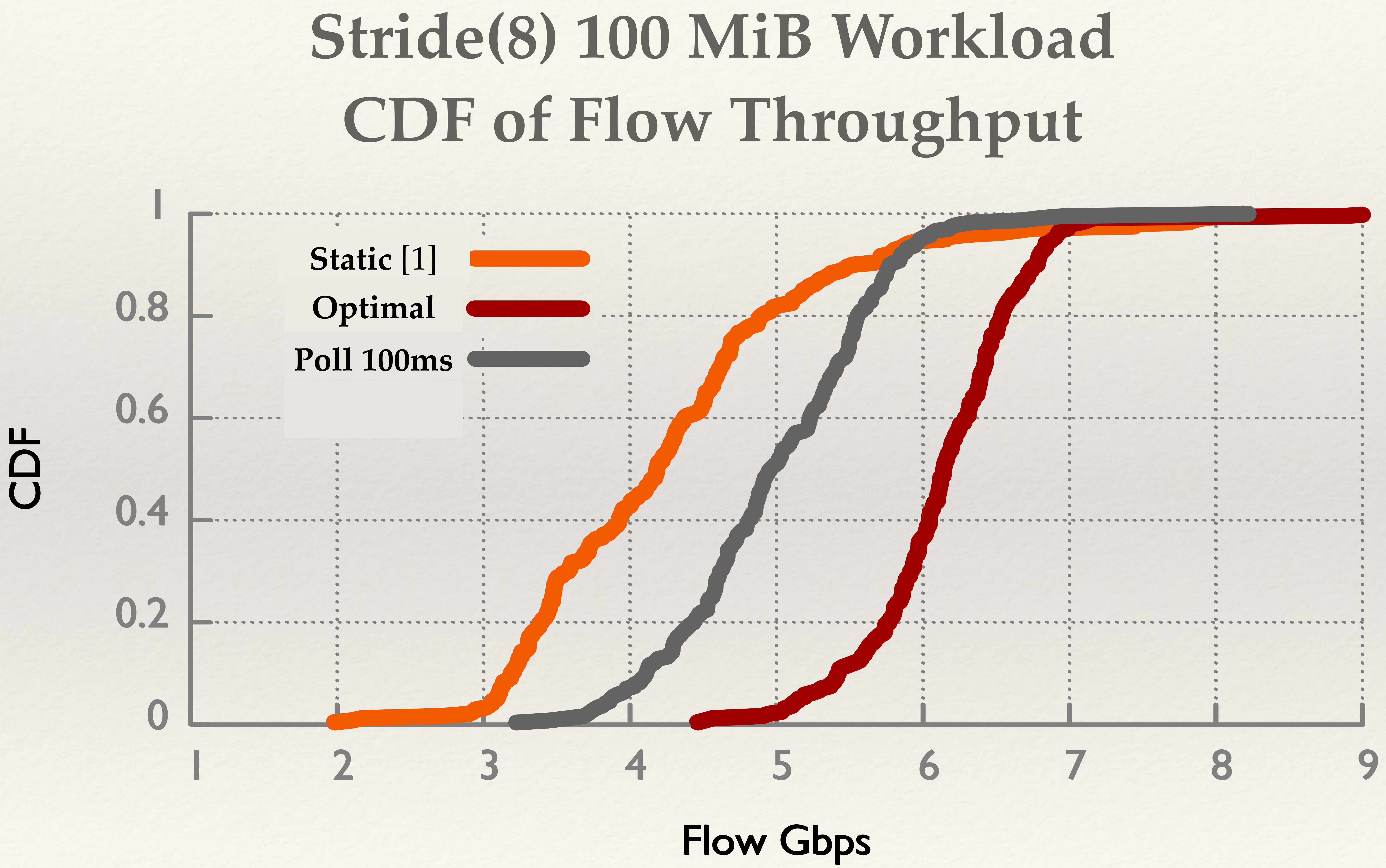


[1] PAST: Scalable Ethernet for Data Centers (CoNEXT '12)

Traffic Engineering

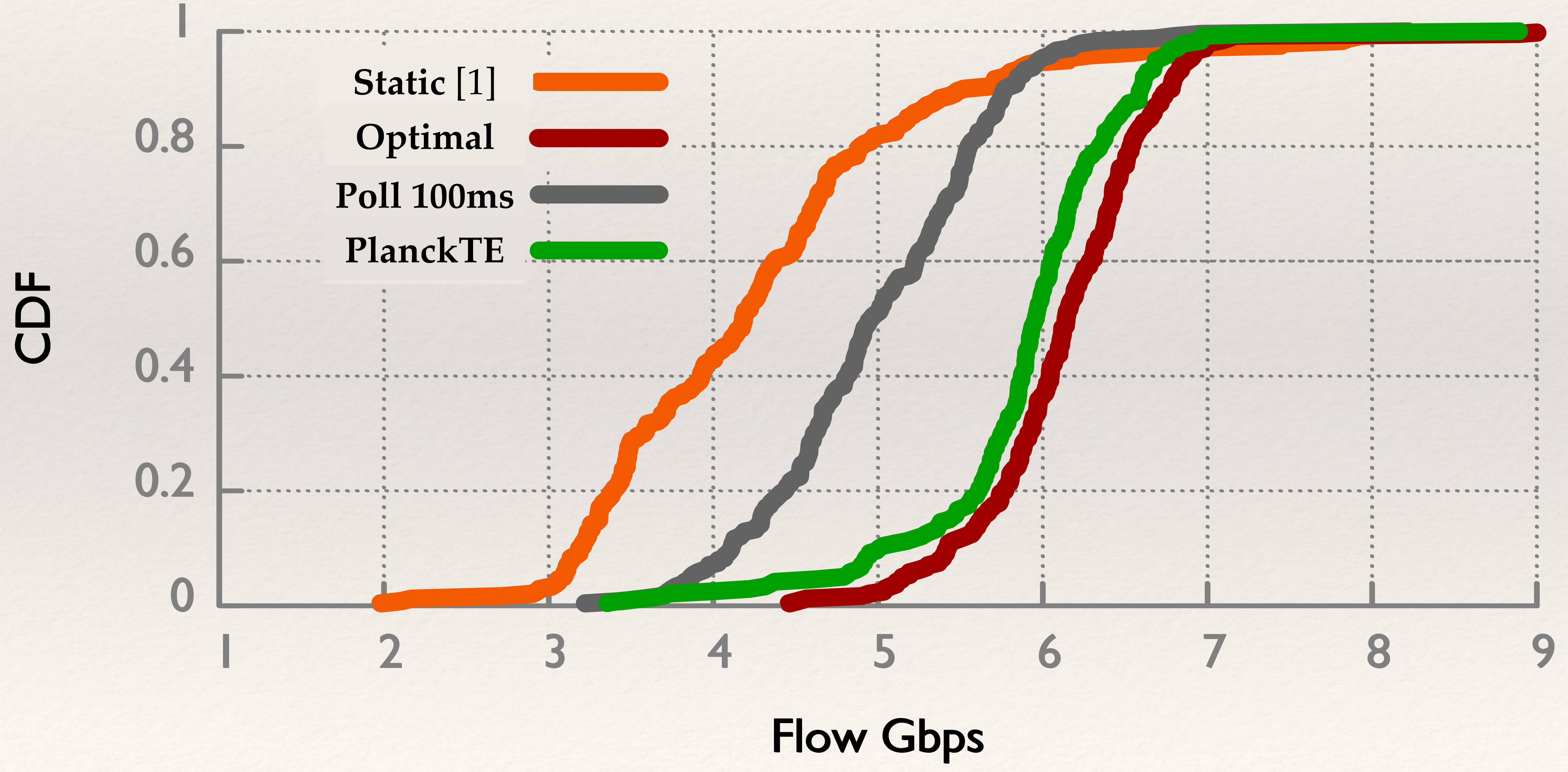


Traffic Engineering

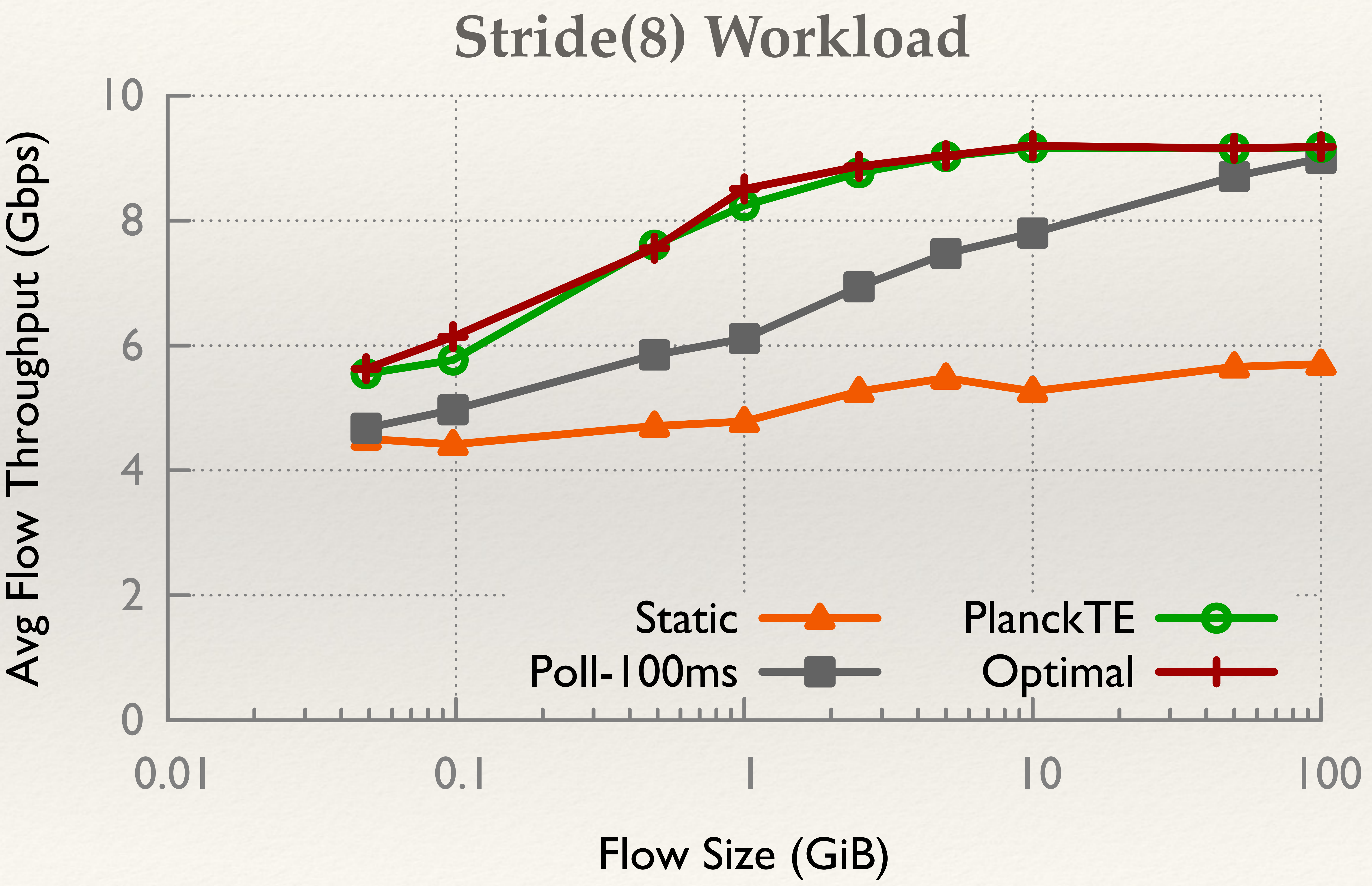


Traffic Engineering

Stride(8) 100 MiB Workload
CDF of Flow Throughput



Traffic Engineering



Future Work

- Planck should be able to go *much* faster!
  - Limit mirror port buffer
  - Truncation of samples
  - Improve re-routing time (via ARP improvements)
- **Control loop of 100s of μs is possible**

Conclusion

- Planck provides **1–2 orders of magnitude** faster throughput measurements over recent approaches (< 4.2 ms today and 100s of μs possible)
- Planck provides a platform for low-latency measurement
- Planck traffic engineering yields **near optimal** results even for small flows
- Measurements at these speeds prompt a re-thinking of how networks are managed