# Fast, Accurate Simulation for SDN Prototyping

Mukta Gupta
University of Wisconsin
mgupta@cs.wisc.edu

Joel Sommers
Colgate University
jsommers@colgate.edu

Paul Barford
University of Wisconsin
pb@cs.wisc.edu


# Publication

In the proceedings of hotSDN '13, ACM SIGCOMM, August 16, 2013, Hong Kong, China.

mgupta@cs.wisc.edu

# Overview

- Motivation
- Related Work
- Goals
- *fs-sdn* overview and background
- Design and Implementation
- System Evaluation
- Results: Speedup and Accuracy
- Summary

# Motivation

- Prototyping, evaluating and debugging SDN applications is hard
  - Increasing scale, diversity, and complexity of apps
  - Will my SDN app behave as expected when deployed in the wild?
  - Does it operate correctly and efficiently at scale?

# Overview

- ✓ Motivation
- ➢ Related Work
- ➢ Goals
- ➢ *fs-sdn* overview and background
- ➢ Design and Implementation
- ➢ System Evaluation
- ➢ Results: Speedup and Accuracy
- ➢ Summary

# SDN prototyping and debugging landscape

Custom fluid-flow simulators (e.g., Al-Fares et al., 2008)

Extensions to packet-level simulators (e.g., Klein and Jarschel, 2013)

fs-sdn

Mininet (Handigol et al., 2012)

Testbed experiments (e.g., Greenberg et al., 2009)

Virtual slice of a live network (e.g., with FlowVisor, Sherwood et al., 2010)

Simulation

Emulation

Live Deployment

mgupta@cs.wisc.edu

# Overview

- ✓ Motivation
- ✓ Related Work
- ➢ Goals
- ➢ *fs-sdn* overview and background
- ➢ Design and Implementation
- ➢ System Evaluation
- ➢ Results: Speedup and Accuracy
- ➢ Summary

# Goals

- A controller API environment to facilitate transition to live environments
- Ability to generate realistic application traffic flows
- Capability to scale up to large networks
- Facilities for detailed debugging and tracing

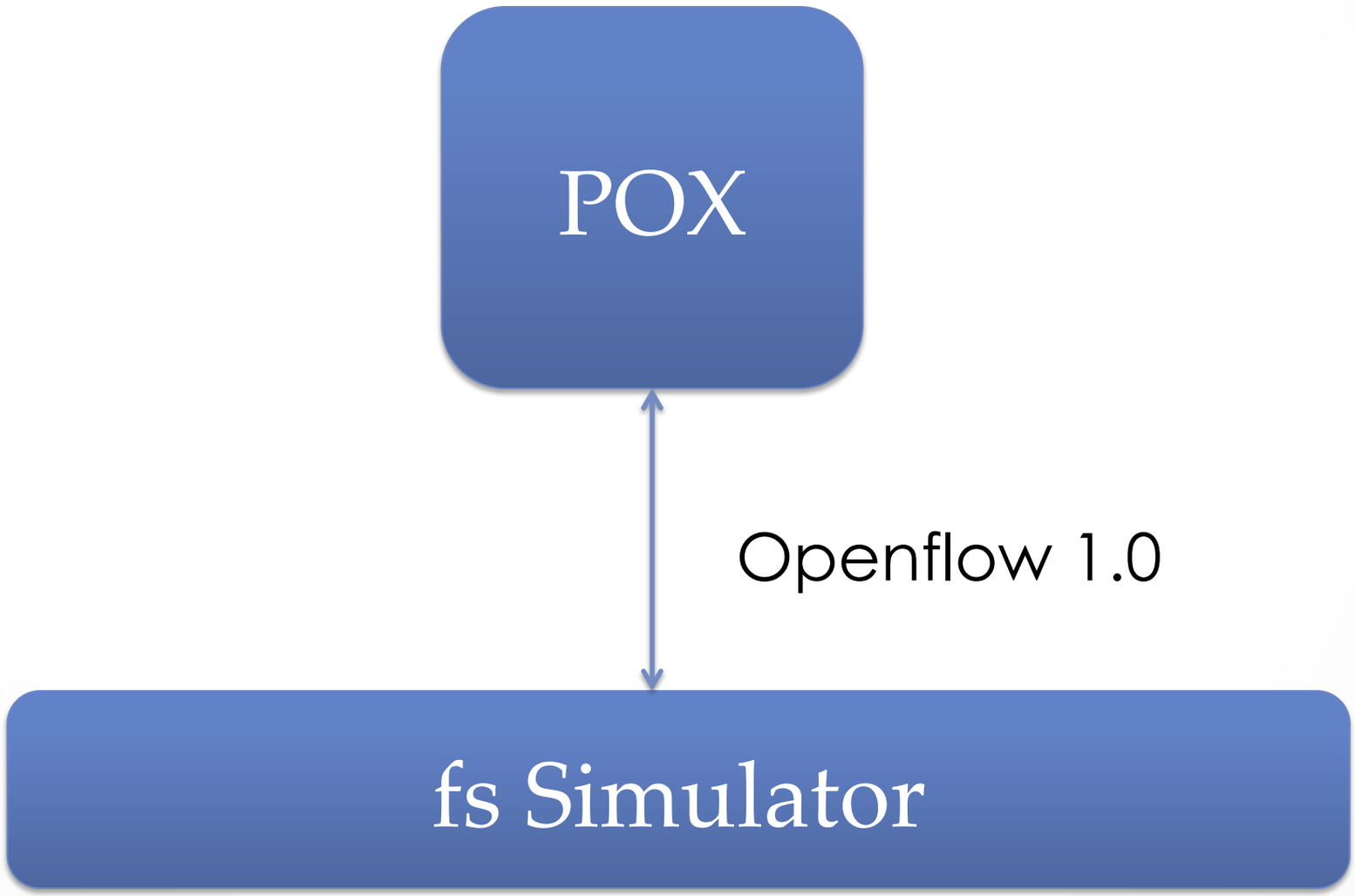
# Overview

- ✓ Motivation
- ✓ Related Work
- ✓ Goals
- ➢ *fs-sdn* overview and background
- ➢ Design and Implementation
- ➢ System Evaluation
- ➢ Results: Speedup and Accuracy
- ➢ Summary

# fs-sdn Background

- Designed as extensions to the *fs* network flow record generator (INFOCOM 2011)
  - Written in Python
  - Uses discrete event simulation to drive flow record generation
  - Flowlets instead of packets
  - Accurate to 1 second time scales, way faster than ns2

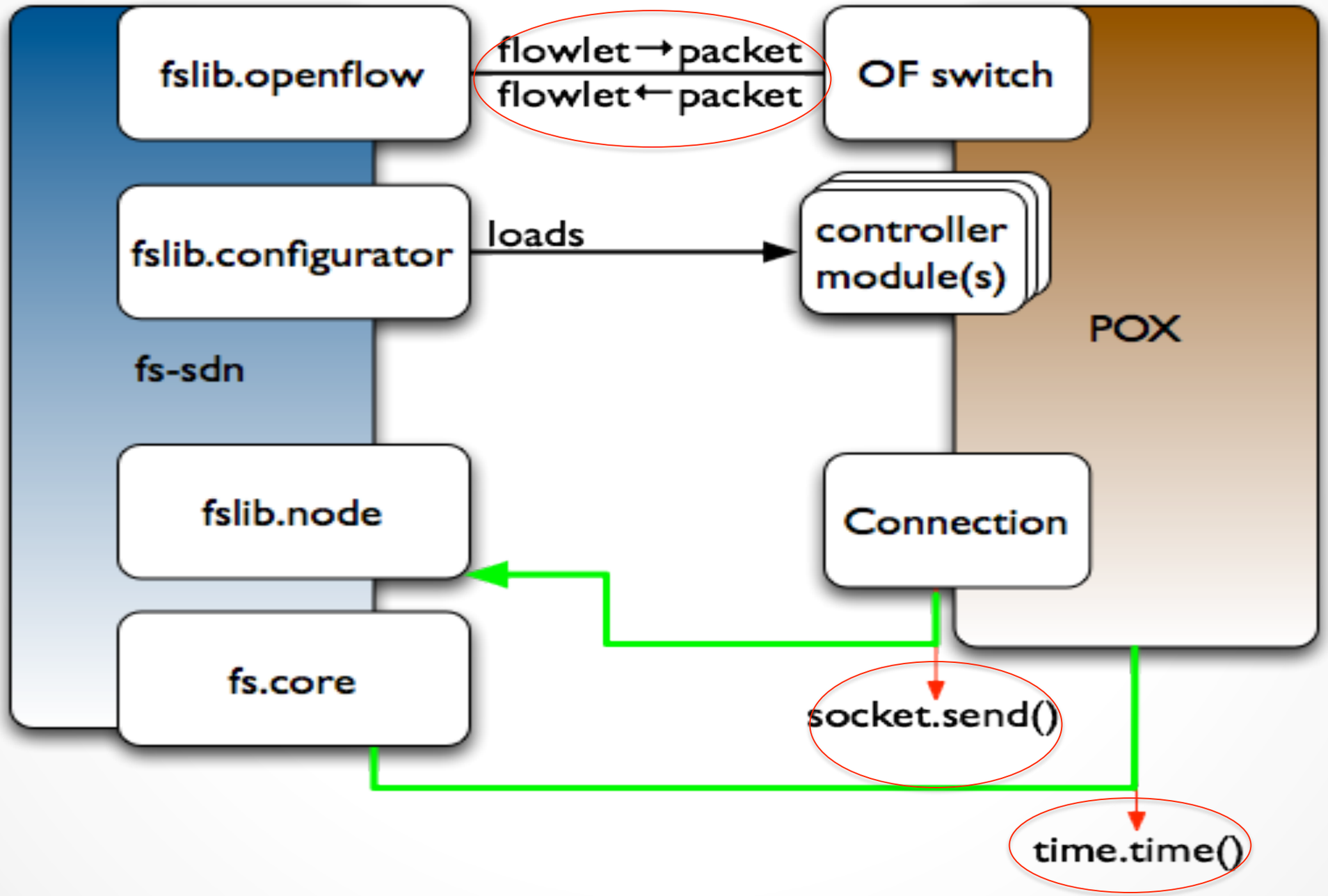
# fs-sdn Design



# Overview

- ✓ Motivation
- ✓ Related Work
- ✓ Goals
- ✓ *fs-sdn* overview and background
- ➢ Design and Implementation
- ➢ System Evaluation
- ➢ Results: Speedup and Accuracy
- ➢ Summary

# fs-SDN design and implementation




mgupta@cs.wisc.edu

# Overview

- ✓ Motivation
- ✓ Related Work
- ✓ Goals
- ✓ *fs-sdn* overview and background
- ✓ Design and Implementation
- ➢ System Evaluation
- ➢ Results: Speedup and Accuracy
- ➢ Summary

# System evaluation

- Set up congruent experiments in fs-sdn and Mininet

**Traffic**
- CBR (10 and 100 Mbps)
- Harpoon Traffic (5 and 25 Mbps)

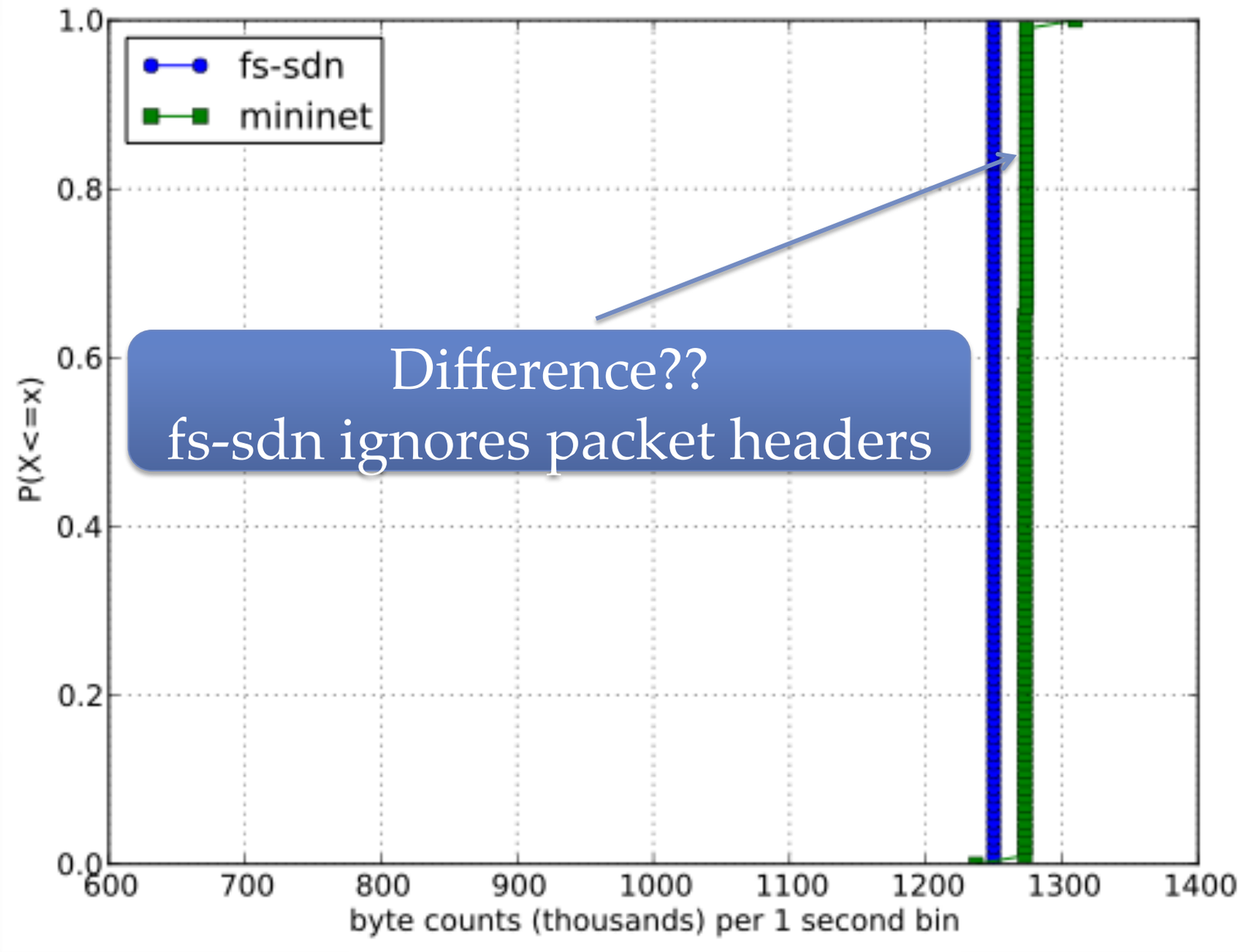
**Topology**
- 1
- 10
- 50
- 100

**Controller**
- L3 Shortest Path

# Overview

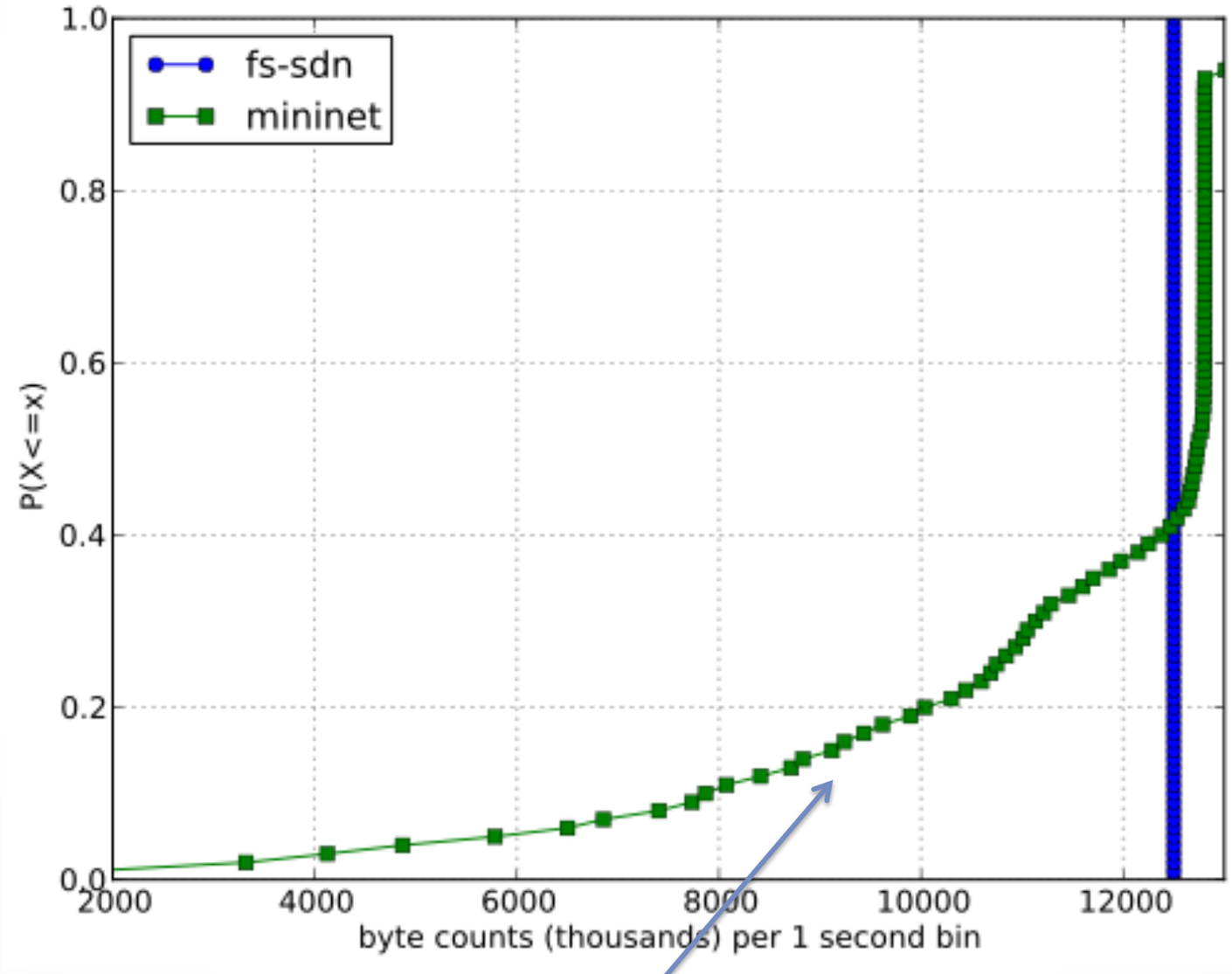
- ✓ Motivation
- ✓ Related Work
- ✓ Goals
- ✓ *fs-sdn* overview and background
- ✓ Design and Implementation
- ✓ System Evaluation
- ➢ Results: Speedup and Accuracy
- ➢ Summary

# Results: accuracy



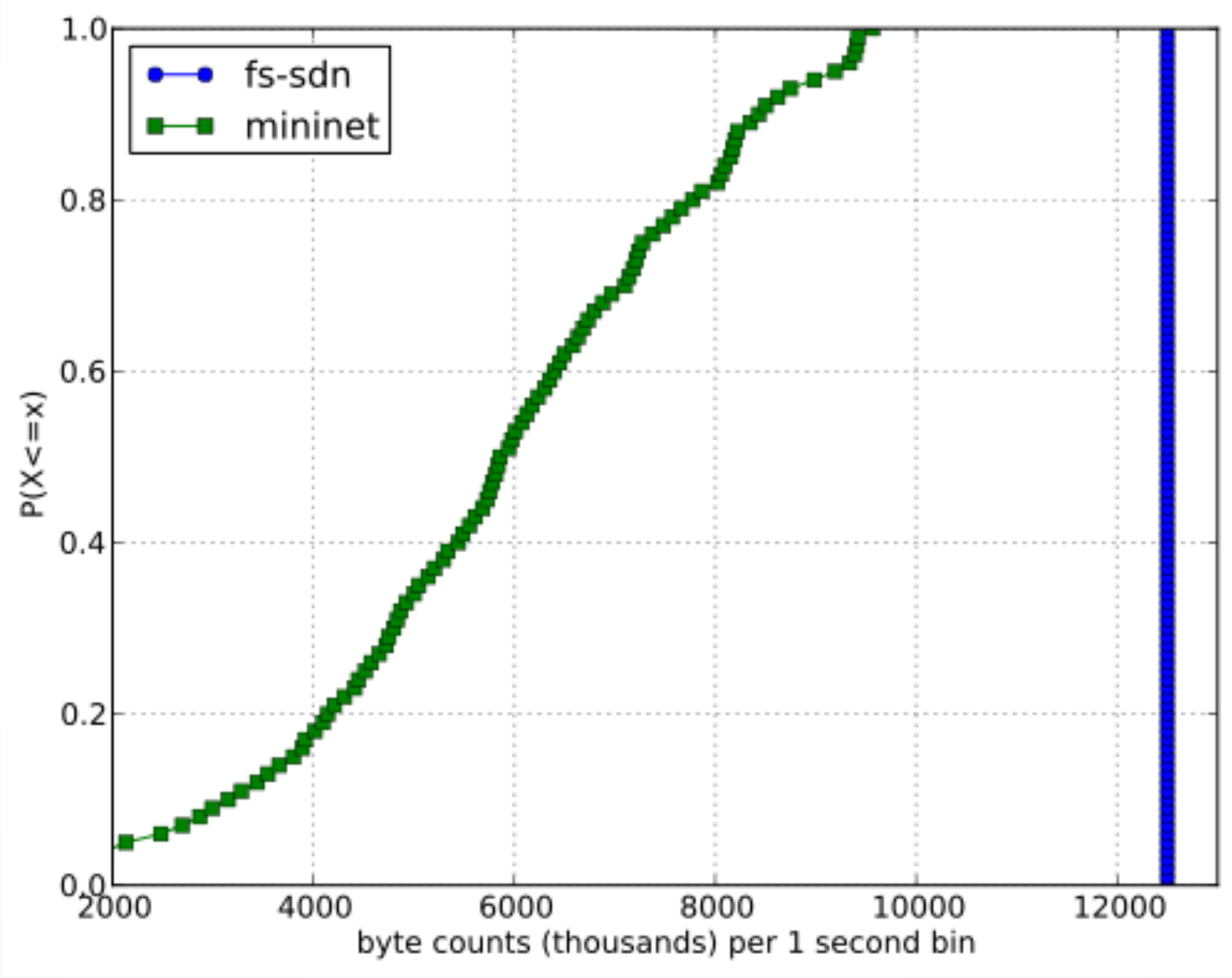

CBR low load (10 Mb/s),
small topology (10 switches)

# Results: accuracy




Performance impact of underlying host

# Results: accuracy




CBR high load (100 Mb/s),
medium topology (50 switches)

# Results: speedup

- Tables show fs-sdn execution times for scenarios with 900 simulated seconds

| UDP CBR traffic | | | | |
|---|---|---|---|---|
| Load | Tiny | Small | Medium | Large |
| Low | 6 | 8 | 33 | 72 |
| High | 4 | 8 | 31 | 76 |

| Harpoon traffic (Pareto distr. flow sizes) | | | | |
|---|---|---|---|---|
| Load | Tiny | Small | Medium | Large |
| Low | 16 | 33 | 104 | 193 |
| High | 30 | 62 | 194 | 337 |

# Overview

- ✓ Motivation
- ✓ Related Work
- ✓ Goals
- ✓ *fs-sdn* overview and background
- ✓ Design and Implementation
- ✓ System Evaluation
- ✓ Results: Speedup and Accuracy
- ➢ Summary

# Summary and future work

- Fast and Accurate Simulation
- Seamless transition of controllers to "real" deployments
- Code available: https://github.com/jsommers/fs

- Continued work
  - Debugging and tracing capability
  - Improve scalability through parallelizing fs
  - Is it possible to bridge other (including non-Python) controller platforms?

# Thank You

## Questions?



mgupta@cs.wisc.edu