Advanced Topics in Communication Networks

Programming Network Data Planes

Laurent Vanbever

nsg.ee.ethz.ch

ETH Zürich

Oct 25 2018

Two weeks ago on

Advanced Topics in Communication Networks

A bloom filter is a streaming algorithm

answering specific questions approximately.

Recap

A bloom filter is a streaming algorithm

answering specific questions approximately.

Is X in the stream?
What is in the stream? Invertible Bloom Filter

A bloom filter is a streaming algorithm

***answering specific questions approximately*.**

Is X in the stream?

What is in the stream? Invertible Bloom Filter

What about other questions?

Is a certain flow in the stream?

Bloom Filter

What flows are in the stream?

Invertible Bloom Filter, HyperLogLog Sketch, …

How frequently does an flow appear?

Count Sketch, CountMin Sketch, …

What are the most frequent elements?

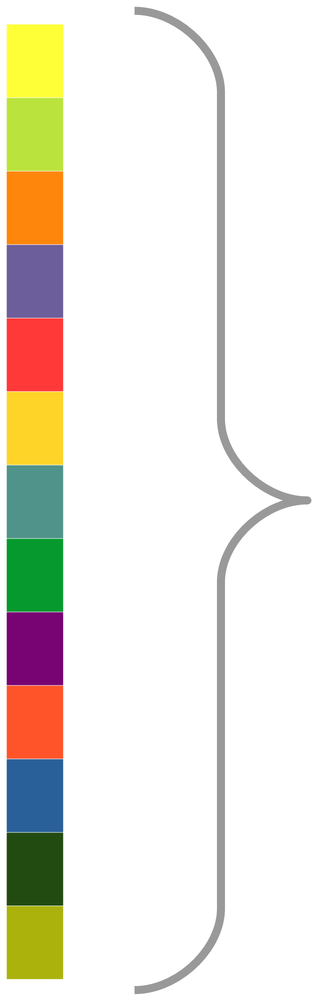
Count/CountMin + Heap, …

How many flows belong to a certain subnet?

SketchLearn SIGCOMM '18

In the worst case, an algorithm providing

exact frequencies requires **linear space**.



Data Stream

n elements in total

→ **n distinct elements**

(in the worst case)

→ **n counters** required? **:(**

Probabilistic datastructures can help again!

Bloom Filters

quickly "filter" only those elements that might be in the set

Save space by allowing false positives.

Sketches

provide a approximate frequencies of elemetns in a data stream.

Save space by allowing mis-counting.

*A **CountMin sketch** uses the same principles as a counting bloom filter, but is **designed** to have **provable L1 error bounds** for frequency queries.*

*A **CountMin sketch** uses the same principles as a counting bloom filter, but is **designed** to have **provable L1 error bounds** for frequency queries.*

relative to L1 norm

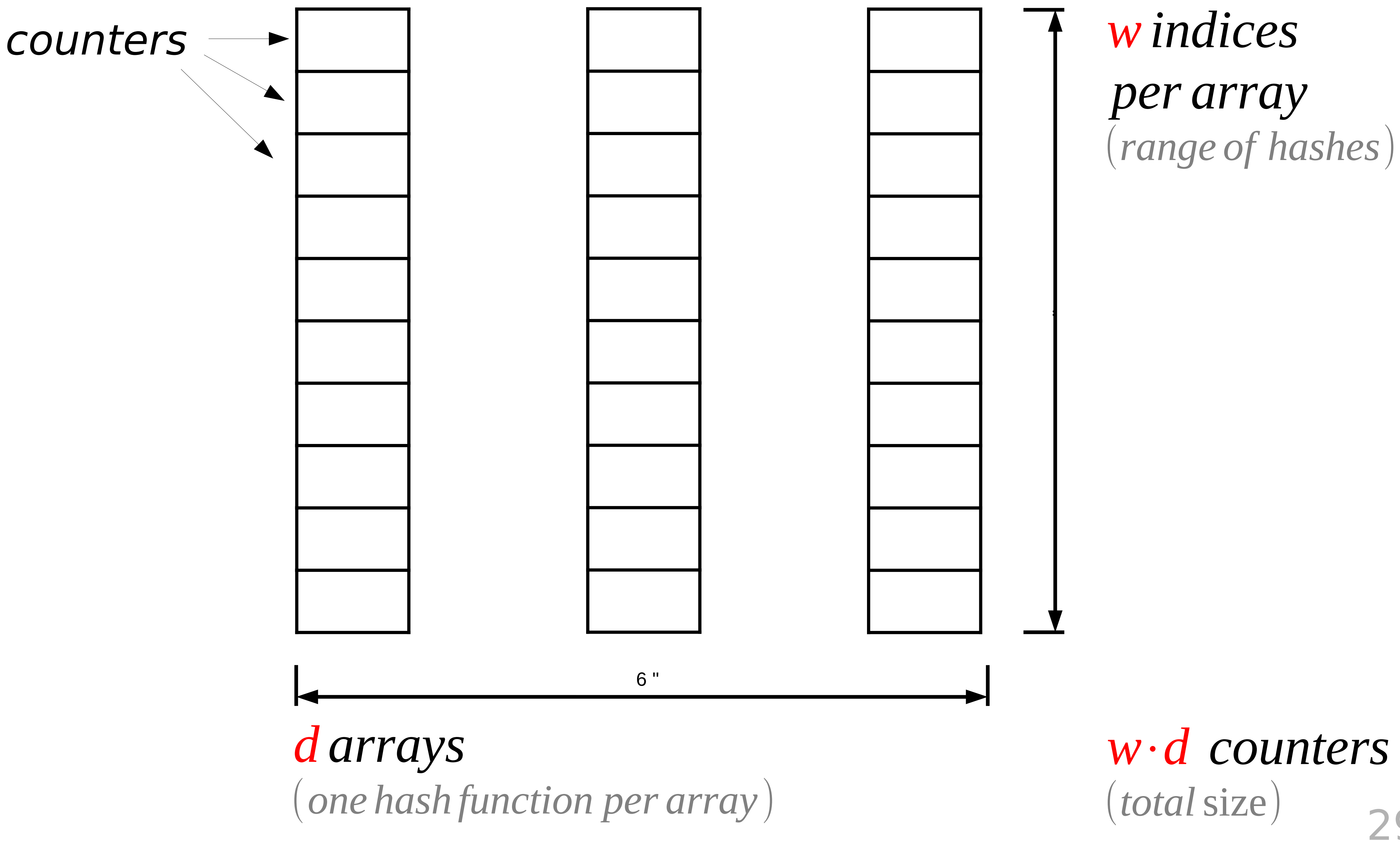
$$Pr\left[\ \underset{\text{estimated frequency}}{\hat{x}_i} \ - \ \underset{\text{true frequency}}{x_i} \ \geq \ \varepsilon \underset{\text{sum of frequencies}}{\|\mathbf{x}\|_1}\ \right] \leq \delta$$

The estimation error ***exceeds*** $\varepsilon \|\mathbf{x}\|_1$

with a ***probability smaller than*** δ

*A **CountMin sketch** uses the same principles as a counting bloom filter, but is **designed** to have **provable L1 error bounds** for frequency queries.*

A **CountMin** Sketch uses multiple arrays and hashes.



*A **CountMin sketch** uses the same principles as a counting bloom filter, but is **designed** to have **provable L1 error bounds** for frequency queries.*

***CountMin** sketch recipe*

Choose $d = \left\lceil \ln \frac{1}{\delta} \right\rceil, w = \left\lceil \frac{e}{\varepsilon} \right\rceil$

Then $\hat{x}_i - x_i \geq \varepsilon \|\mathbf{x}\|_1$ *with a probability less than* δ

*A **CountMin sketch** uses the same principles as a counting bloom filter, but is **designed** to have **provable L1 error bounds** for frequency queries.*

→ only one design out of many!

A ***Count sketch*** *uses the same principles as a counting bloom filter, but is* ***designed*** *to have* ***provable L2 error bounds*** *for frequency queries.*

The Count sketch uses **additional hashing** to give **L2 error bounds**, but requires more **resources**.

***CountMin** sketch recipe*

Choose $d = \left\lceil \ln \frac{1}{\delta} \right\rceil, w = \left\lceil \frac{e}{\varepsilon} \right\rceil$

Then $\hat{x}_i - x_i \geq \varepsilon \|\mathbf{x}\|_1$ *with a probability less than* δ

***Count** sketch recipe*

Choose $d = \left\lceil \ln \frac{1}{\delta} \right\rceil, w = \left\lceil \frac{e}{\boldsymbol{\varepsilon}^2} \right\rceil$

Then $\hat{x}_i - x_i \geq \varepsilon \|\mathbf{x}\|_2$ *with a probability less than* δ

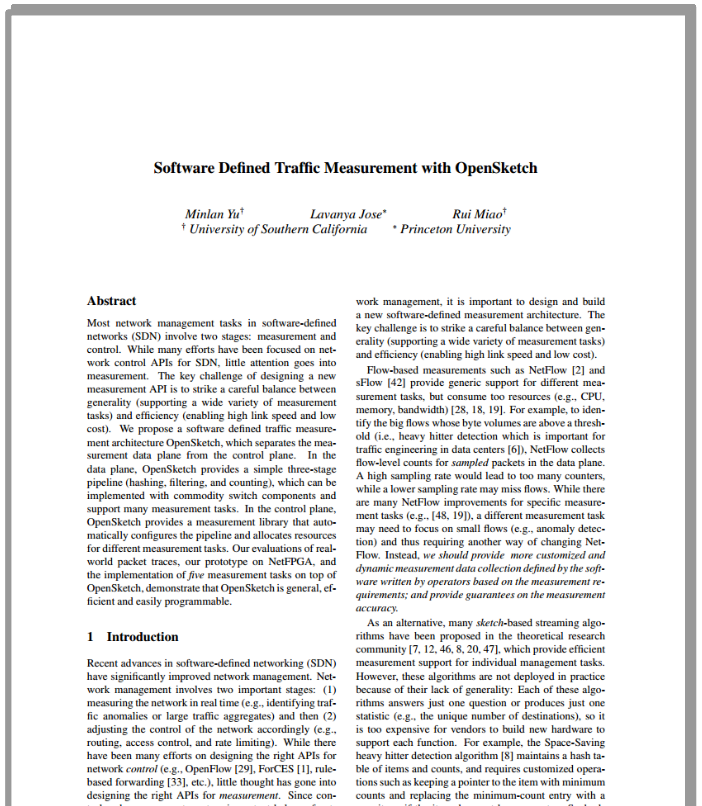
Sketches **are the new black**

...and many more!

OpenSketch

NSDI '13

[source]



Software Defined Traffic Measurement with OpenSketch

Minlan Yu† Lavanya Jose* Rui Miao†
† University of Southern California * Princeton University

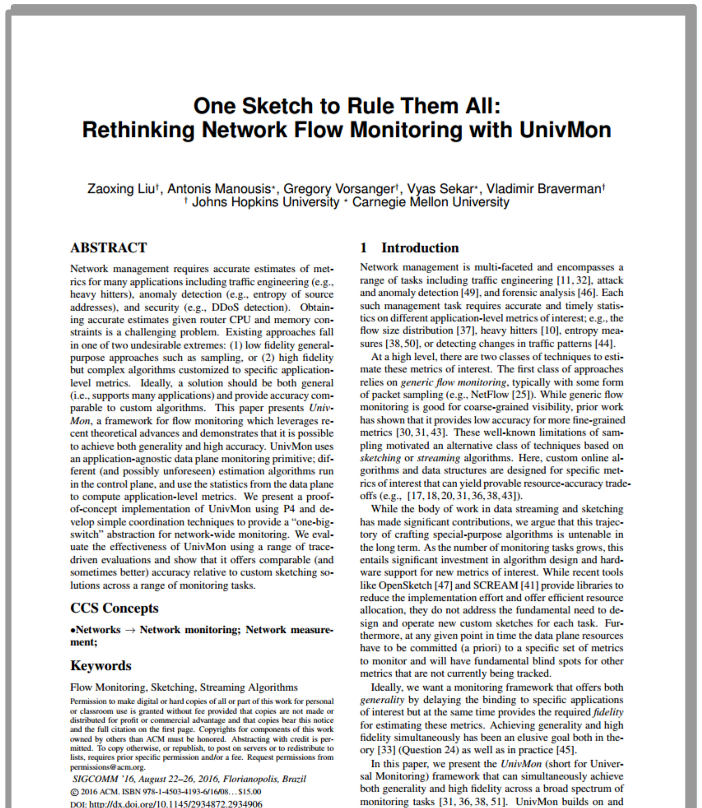
Abstract

1 Introduction

UnivMon

SIGCOMM '16

[source]



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

Zaoxing Liu†, Antonis Manousis*, Gregory Vorsanger†, Vyas Sekar*, Vladimir Braverman†
† Johns Hopkins University * Carnegie Mellon University

ABSTRACT

1 Introduction

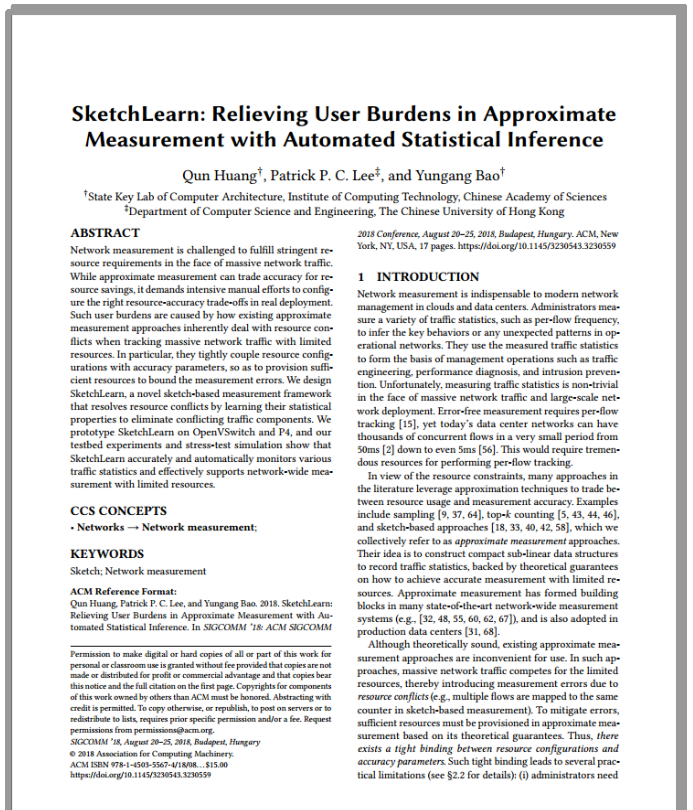
CCS Concepts

Keywords

SketchLearn

SIGCOMM '18

[source]



SketchLearn: Relieving User Burdens in Approximate Measurement with Automated Statistical Inference

Qun Huang†, Patrick P. C. Lee‡, and Yungang Bao†
†State Key Lab of Computer Architecture, Institute of Computing Technology, Chinese Academy of Sciences
‡Department of Computer Science and Engineering, The Chinese University of Hong Kong

ABSTRACT

1 INTRODUCTION

CCS CONCEPTS

KEYWORDS

Today we’ll talk about: important questions,

how ‘sketches’ answer them,

limitations of ‘sketches’,

and my master thesis :)

Sketches **compute statistical summaries**, favoring elements with **high frequency**.

$$Let \quad \varepsilon = 0.01, \quad \|\boldsymbol{x}\|_1 = 10000 \quad \left(\Rightarrow \varepsilon \cdot \|\boldsymbol{x}\|_1 = 100\right)$$

$$Assume\ two\ flows \quad x_a, \quad x_b,$$

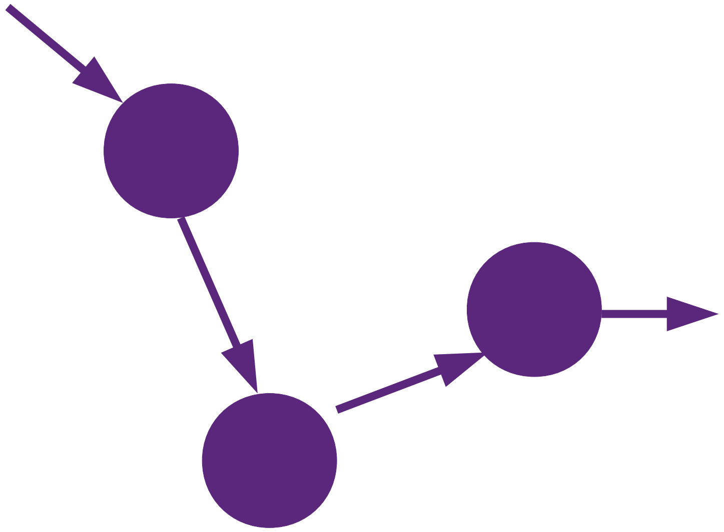
$$with \quad \|x_a\|_1 = 1000, \quad \|x_b\|_1 = 50$$

Error relative to **stream size**: 1%

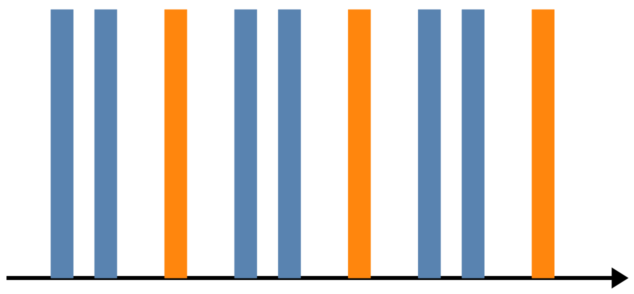
flow size: x_a: 10%, x_b: **200%**

Other Problems a Sketch **can't handle**

causality



patterns



rare things

This week on

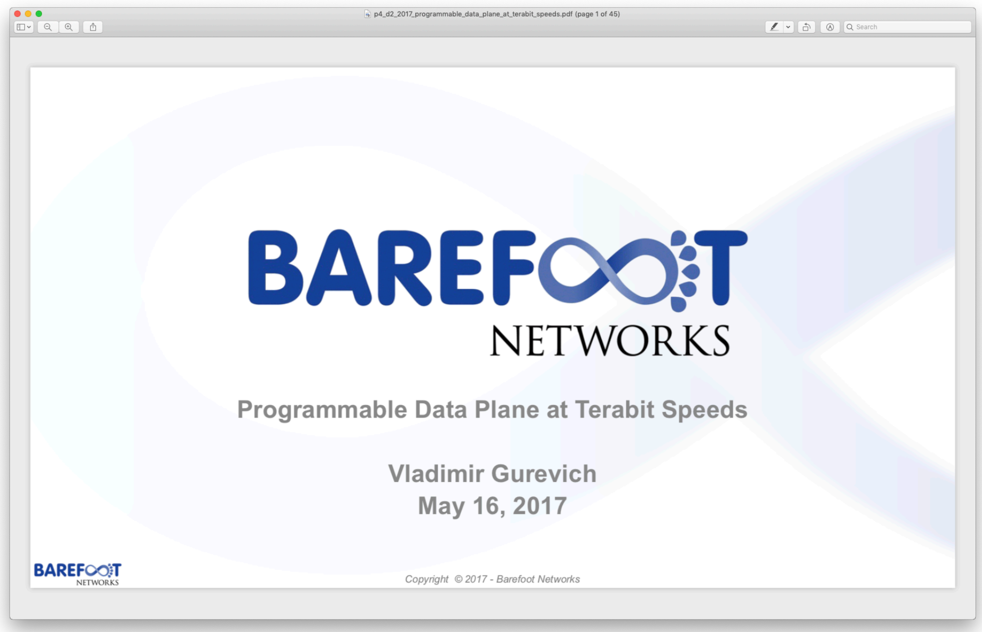
Advanced Topics in Communication Networks

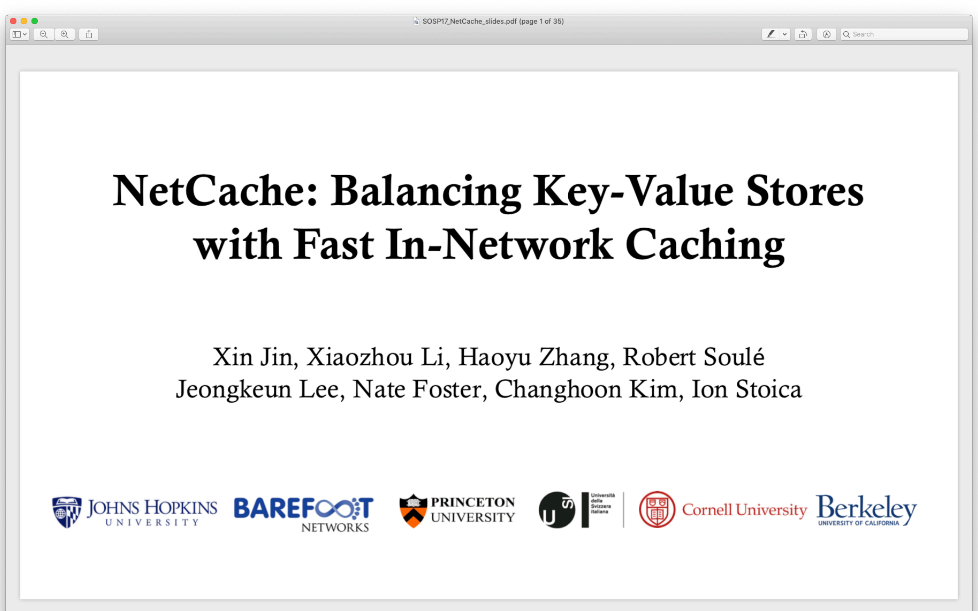
We will look at one example of a **P4 hardware** switch along with two examples of **P4-enabled applications**

P4 hardware	NetCache	NetChain
How to map P4 to Barefoot Tofino's platform	load-balancing cache for key-value store	consistent, fault-tolerant key-value store
	[SOSP'17]	[NSDI'18]

+ Albert Gran's Master Thesis presentation on "Making Scheduling Programmable"

See https://adv-net.ethz.ch for follow-up slides





Advanced Topics in Communication Networks

Programming Network Data Planes

Laurent Vanbever

nsg.ee.ethz.ch

ETH Zürich

Oct 25 2018