Chair of Communication Networks
Department of Electrical and Computer Engineering
Technical University of Munich

Adversarial Network Benchmarking: A Data-Driven Approach

Andreas Blenk (University of Vienna, TU Munich)°*

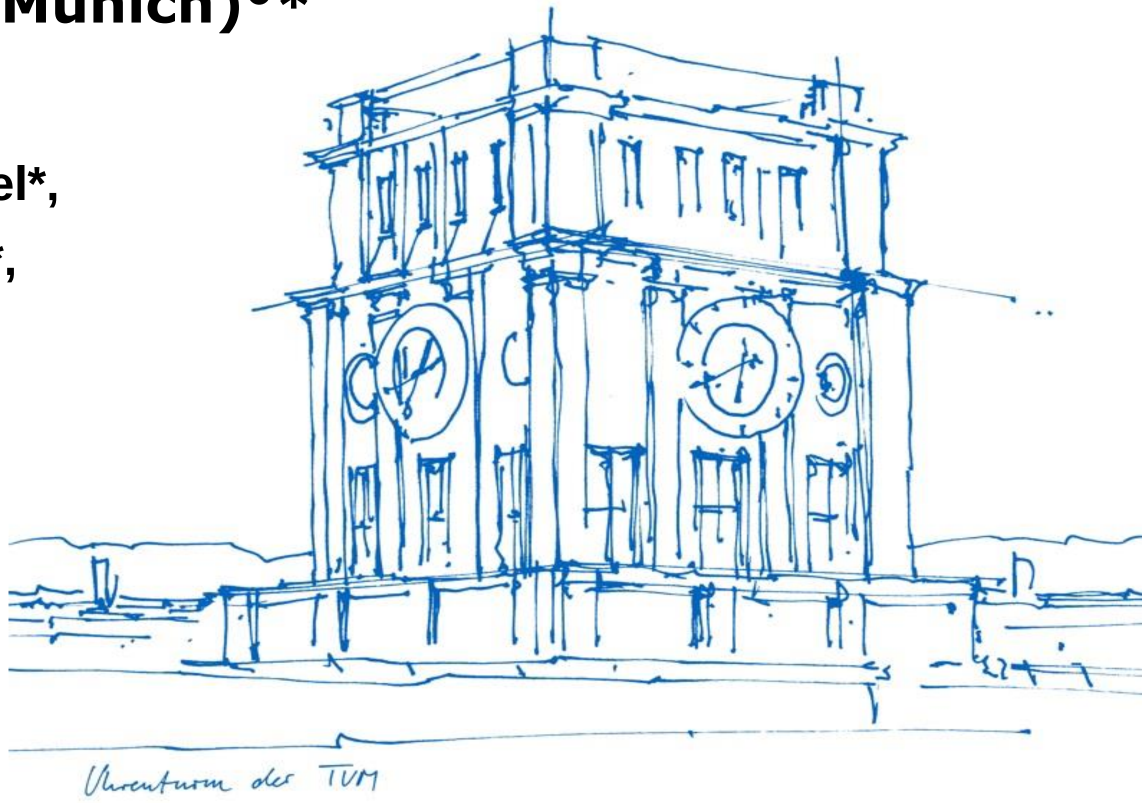
Joint work with:

Johannes Zerwas*, Patrick Kalmbach*, Laurenz Henkel*, Sebastian Lettner, Gábor Rétvári^, Wolfgang Kellerer*, Stefan Schmid°

**Technical University of Munich, Germany*

^Budapest University of Technology and Economics, Hungary

°Faculty of Computer Science, University of Vienna, Austria



SFB MAKI – Scientific Workshop 2020

Self-***Driving*** and Data ***Driven*** Networks?

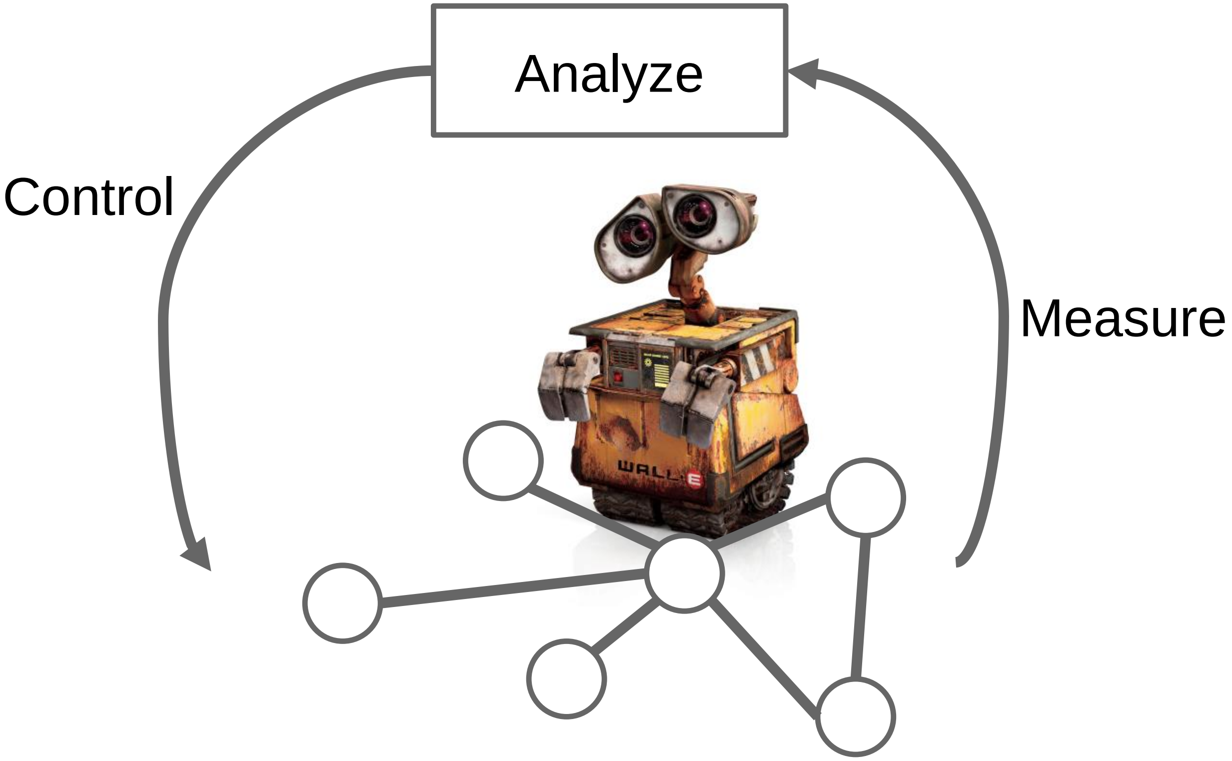
It is not about cars!

N. Feamster and J. Rexford, “Why (and How) Networks Should Run Themselves,” CoRR, vol. abs/1710.11583, 2017.

Self-***Driving*** and Data ***Driven*** Networks?

It is not about cars!

It is about networks that measure, analyze and control themselves!

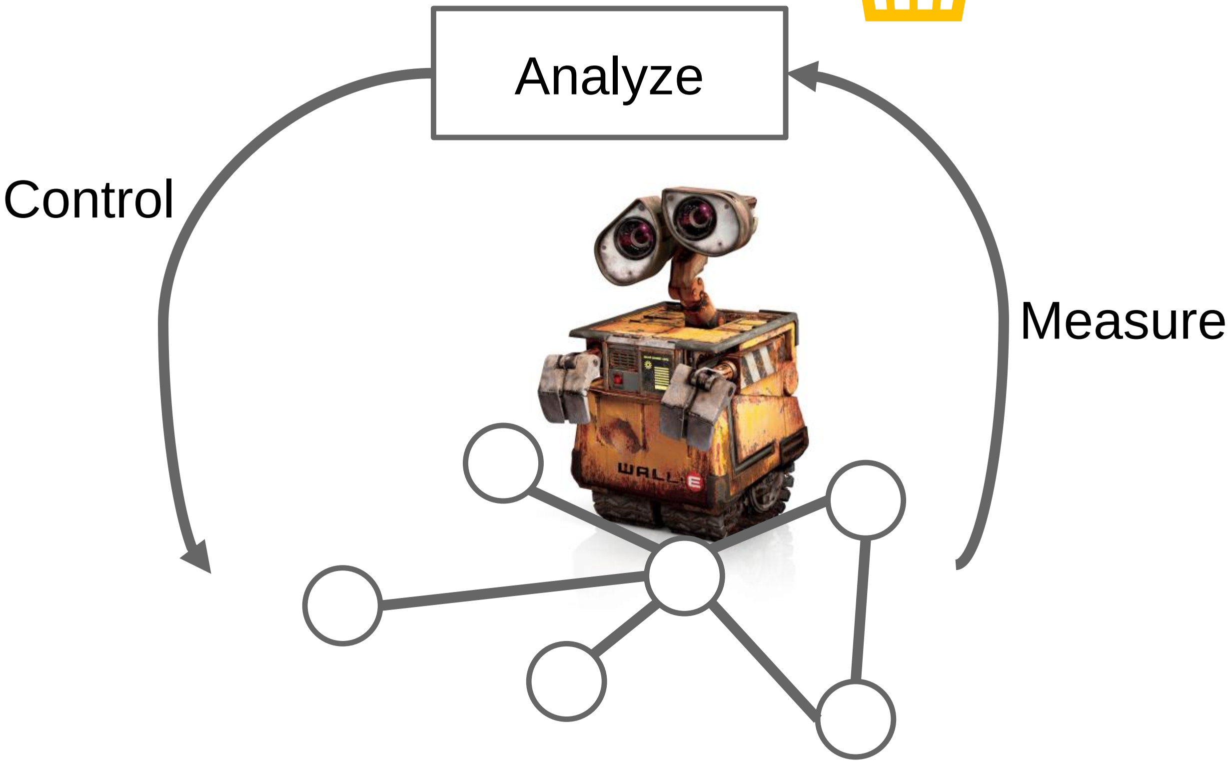


N. Feamster and J. Rexford, "Why (and How) Networks Should Run Themselves," CoRR, vol. abs/1710.11583, 2017.

Self-*Driving* and Data *Driven* Networks?

It is not about cars!

It is about networks that measure, analyze and control themselves!

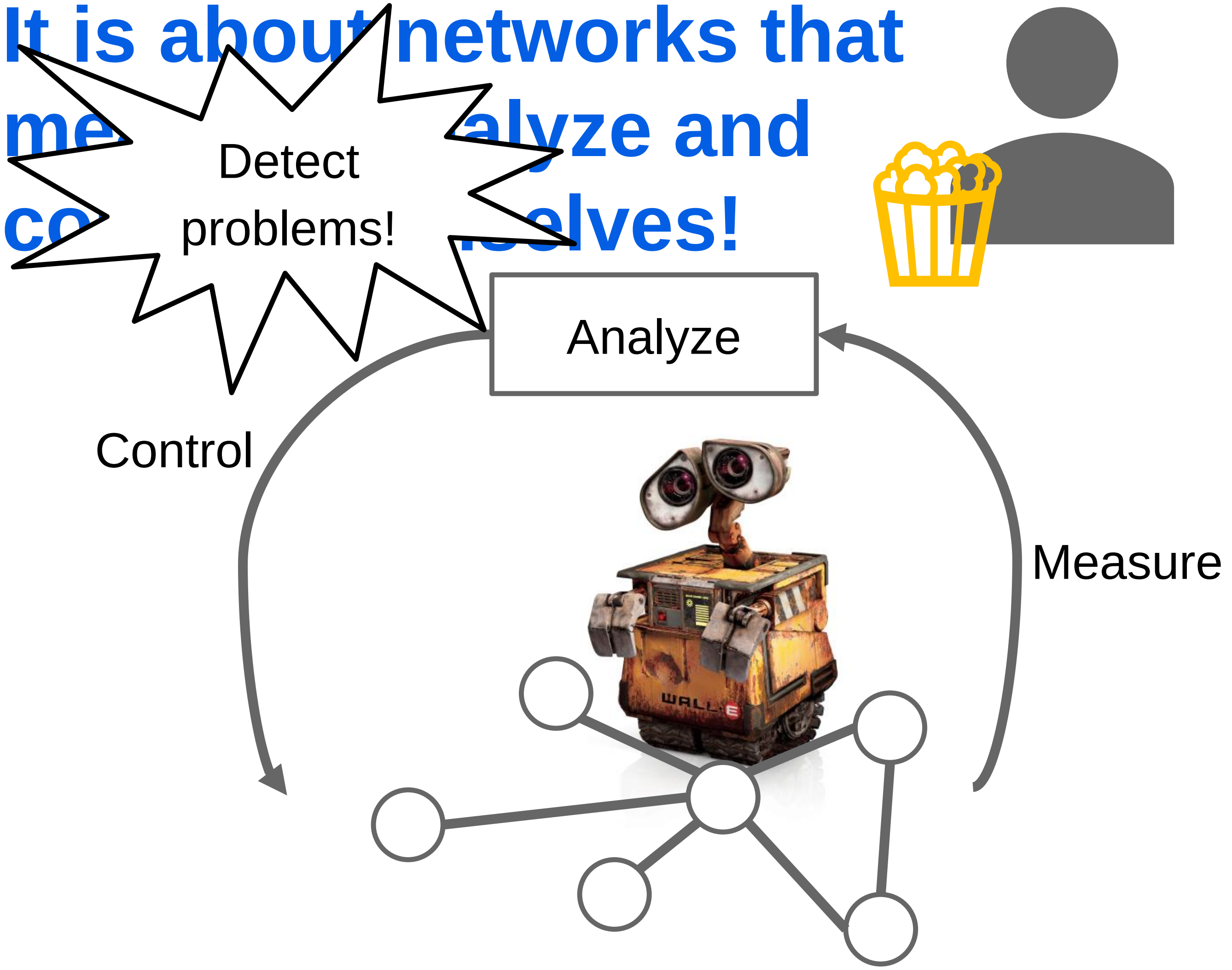


N. Feamster and J. Rexford, "Why (and How) Networks Should Run Themselves," CoRR, vol. abs/1710.11583, 2017.

Self-***Driving*** and Data ***Driven*** Networks?

It is not about cars!

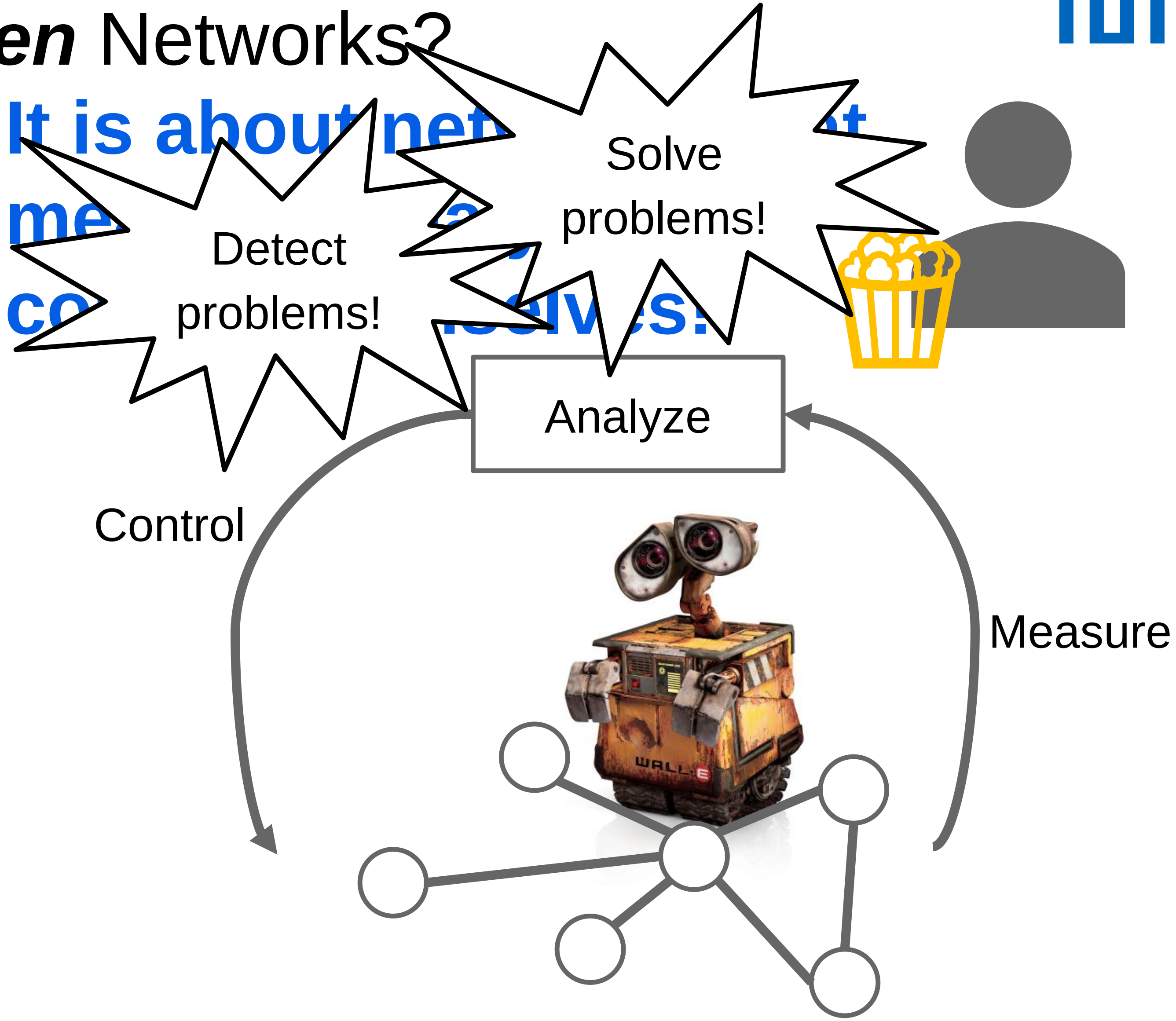
It is about networks that
me… alyze and
co… elves!



N. Feamster and J. Rexford, "Why (and How) Networks Should Run Themselves," CoRR, vol. abs/1710.11583, 2017.

Self-***Driving*** and Data ***Driven*** Networks?

It is not about cars!

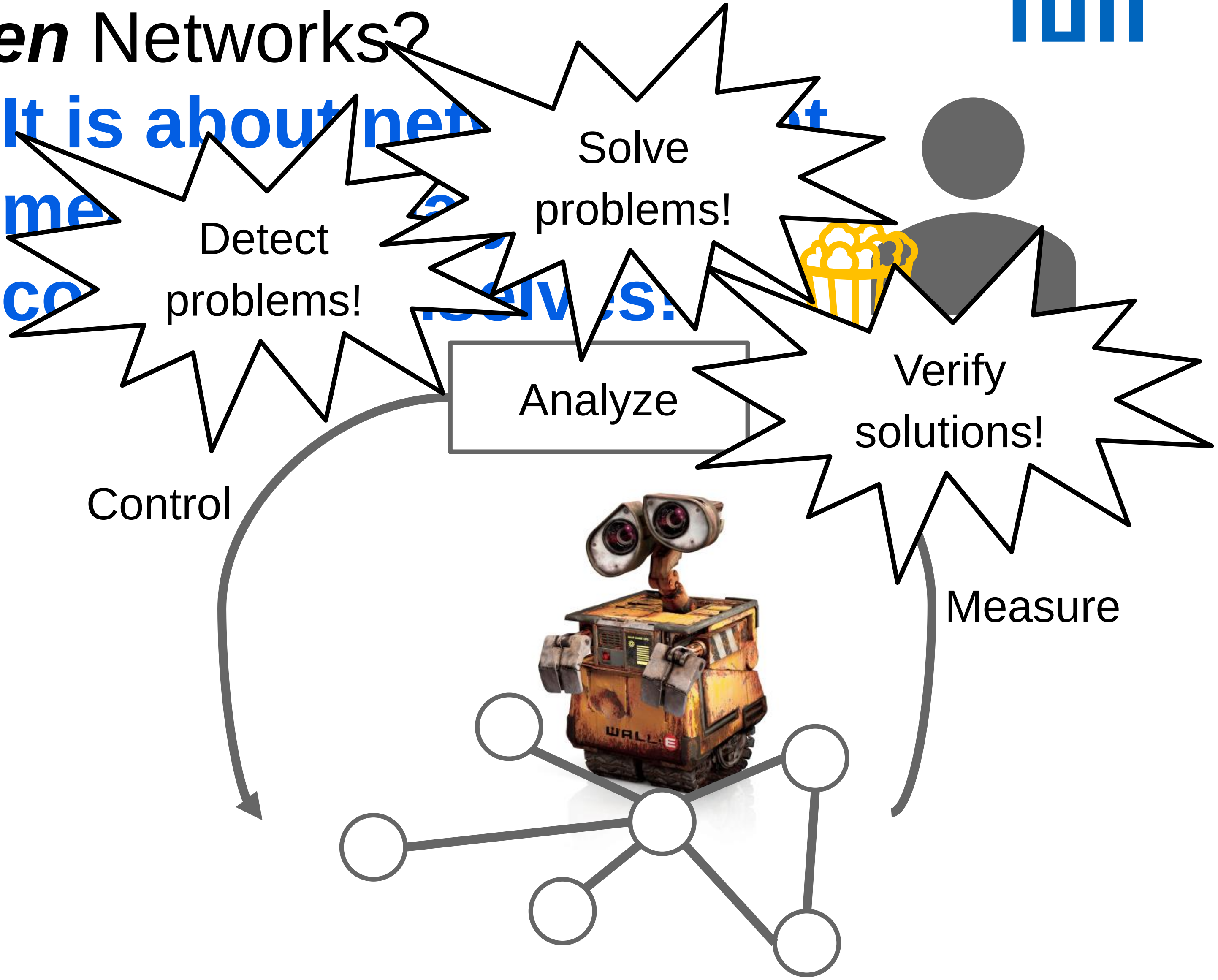


N. Feamster and J. Rexford, “Why (and How) Networks Should Run Themselves,” CoRR, vol. abs/1710.11583, 2017.

Self-***Driving*** and Data ***Driven*** Networks?

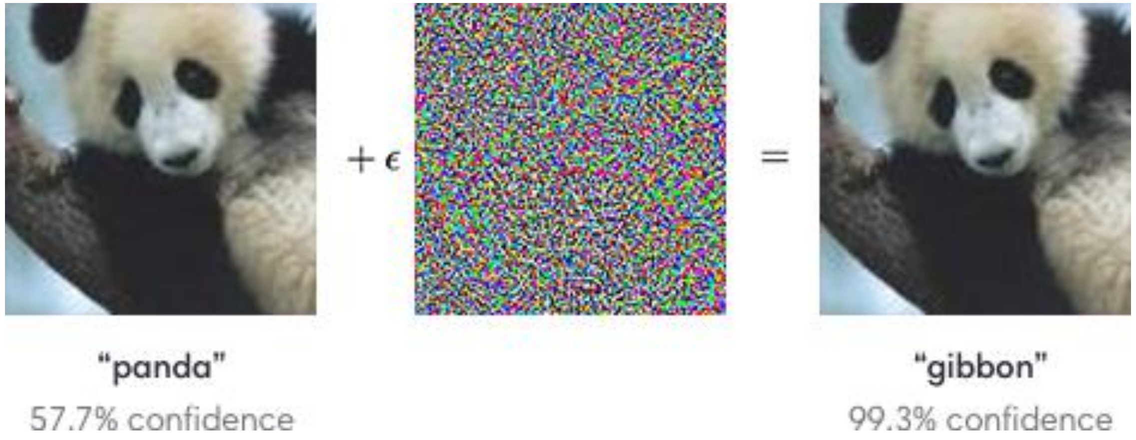
It is not about cars!

It is about net
me
co selves!



N. Feamster and J. Rexford, "Why (and How) Networks Should Run Themselves," CoRR, vol. abs/1710.11583, 2017.

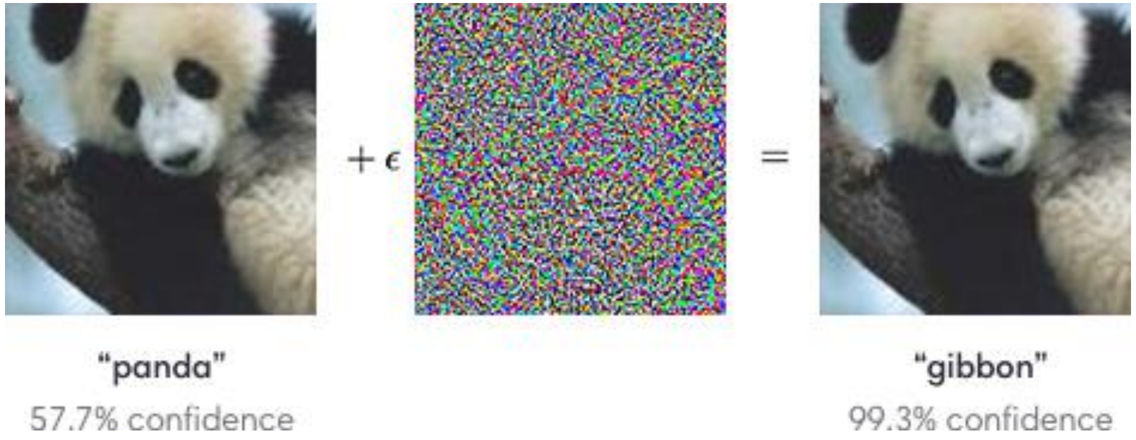
But Data-Driven Systems Can be Tricked



In typical ML applications

Source: https://openai.com/blog/adversarial-example-research/

But Data-Driven Systems Can be Tricked



(Self) Driving Under the Influence:
Intoxicating Adversarial Network Inputs

Roland Meier(1), Thomas Holterbach(1), Stephan Keck(1), Matthias Stähli(1), Vincent Lenders(2), Ankit Singla(1), Laurent Vanbever(1)

ACM HotNets 2019

(1) ETH zürich
(2) Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
armasuisse

In typical ML applications **… and in networking**

Adversarial Input Only Critical for Machine Learning?

Adversarial Input

Machine Learning-based solution

Adversarial Input Only Critical for Machine Learning?

Adversarial Input

Machine Learning-based solution

… but this is also true for existing solutions by human!

Adversarial Input Only Critical for Machine Learning?

Adversarial Input

Machine Learning-based solution

Adversarial Input

Solution designed by human

… but this is also true for existing solutions by human!

Adversarial Input Only Critical for Machine Learning?

Adversarial Input

Machine Learning-based solution

Adversarial Input

Solution designed by human

… but this is also true for existing solutions by human!

Adversarial input is not only critical for self-driving networks …
It’s already a problem!

Adversarial Input Only Critical for Machine Learning?

Adversarial Input

Machine Learning-based solution

Adversarial Input

Solution designed by human

Why?

… but this is also true for existing solutions by human!

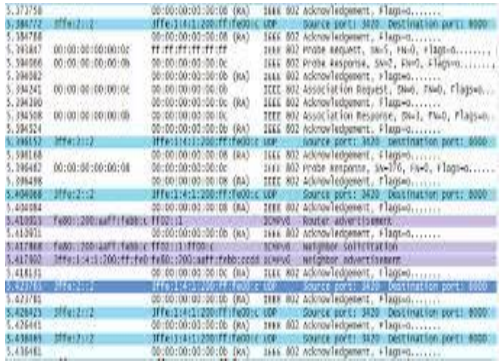
Adversarial input is not only critical for self-driving networks … It’s already a problem!

Benchmarking Network Algorithms, Architectures etc…

The Traditional Way …

Benchmarking Network Algorithms, Architectures etc…

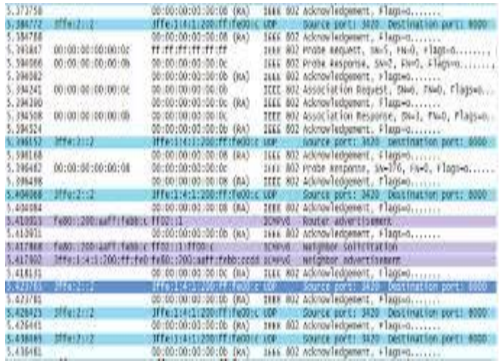
The Traditional Way …



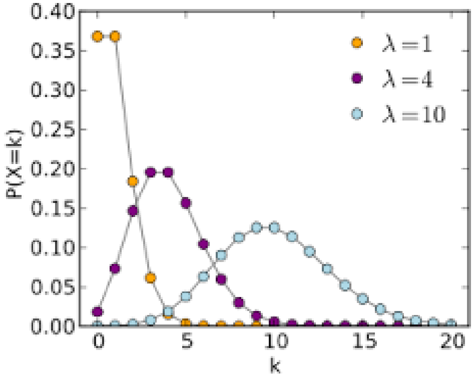
Traces

Benchmarking Network Algorithms, Architectures etc…

The Traditional Way …



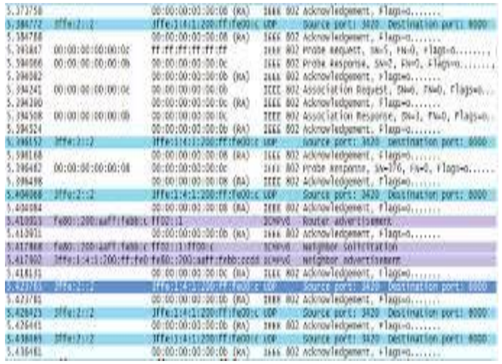
Traces



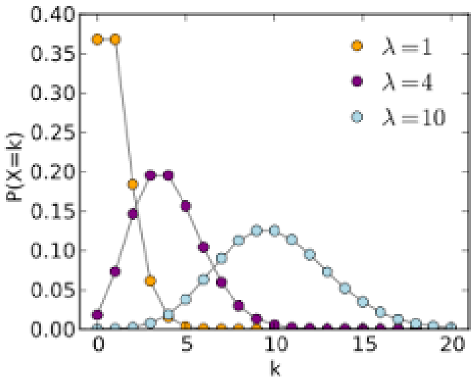
Models

Benchmarking Network Algorithms, Architectures etc…

The Traditional Way …



Traces



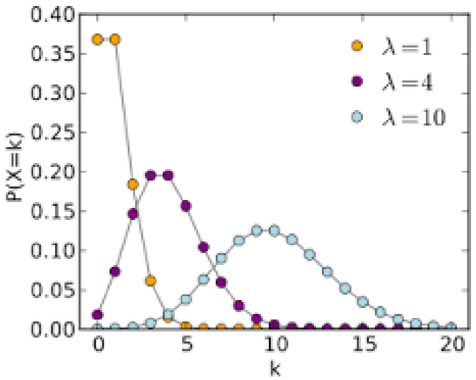
Models

Human‘s Best Guesses

Benchmarking Network Algorithms, Architectures etc…

The Traditional Way …

Traces

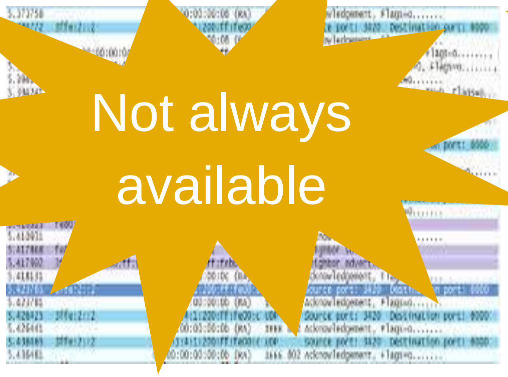


Models

Human‘s Best Guesses

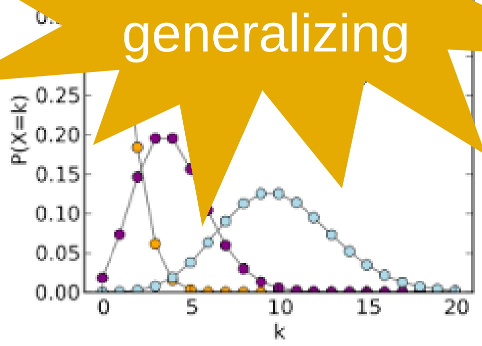
Benchmarking Network Algorithms, Architectures etc…

The Traditional Way …



Not always available

Traces



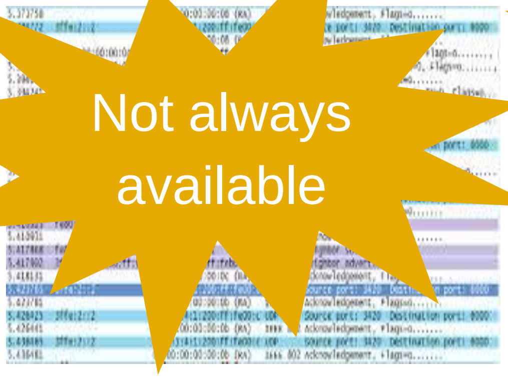
Not generalizing

Models

Human‘s Best Guesses

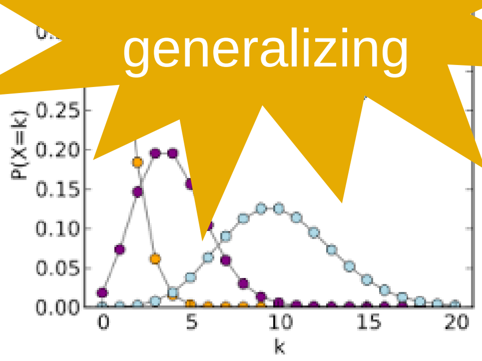
Benchmarking Network Algorithms, Architectures etc…

The Traditional Way …



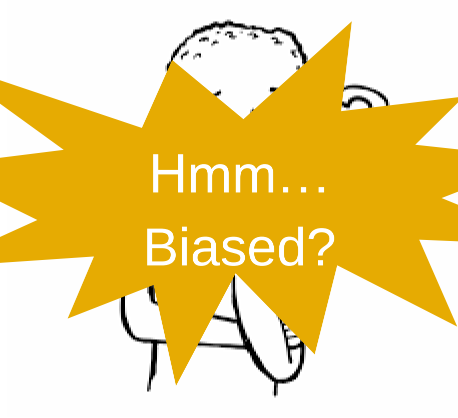
Not always available

Traces



Not generalizing

Models

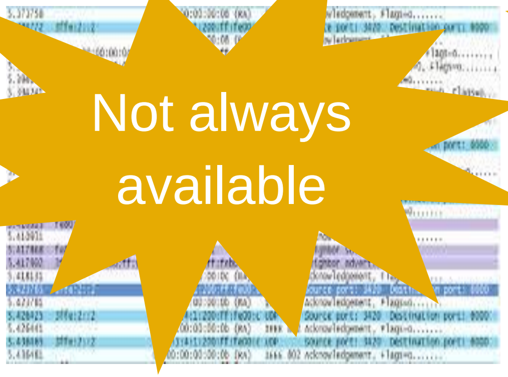


Hmm… Biased?

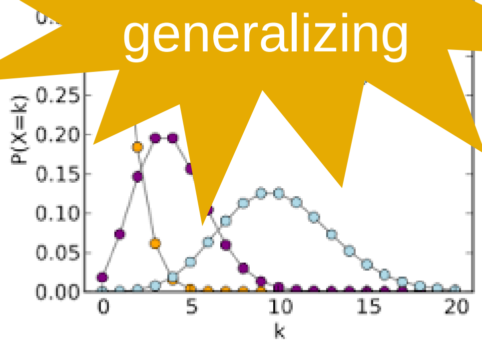
Human‘s Best Guesses

Benchmarking Network Algorithms, Architectures etc…

The Traditional Way …



Traces



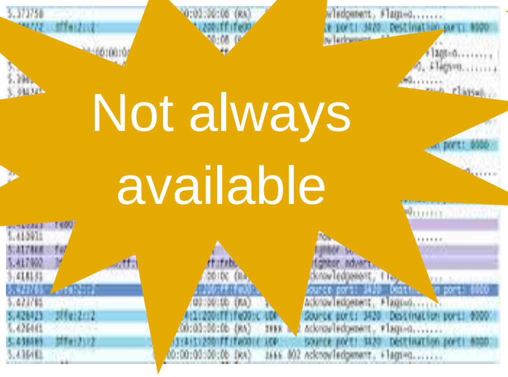
Models

Human‘s Best Guesses

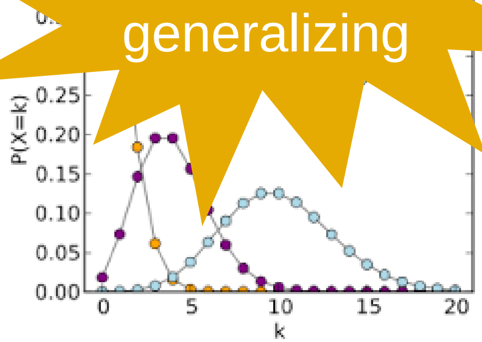
Data-Driven

Benchmarking Network Algorithms, Architectures etc…

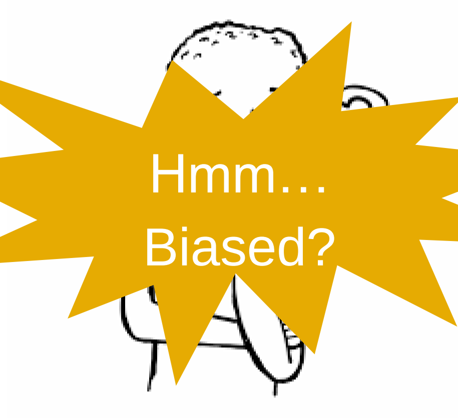
The Traditional Way …



Traces



Models

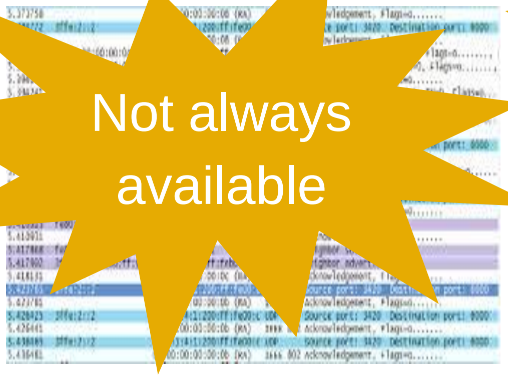


Human‘s Best Guesses

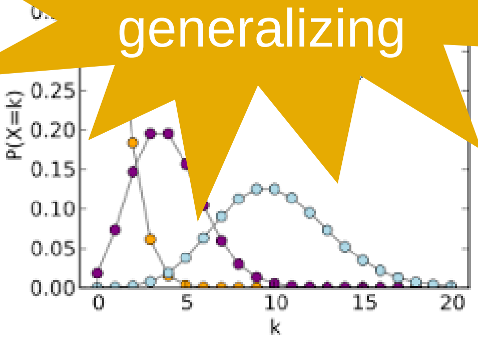
Data-Driven

Benchmarking Network Algorithms, Architectures etc…

The Traditional Way …



Traces



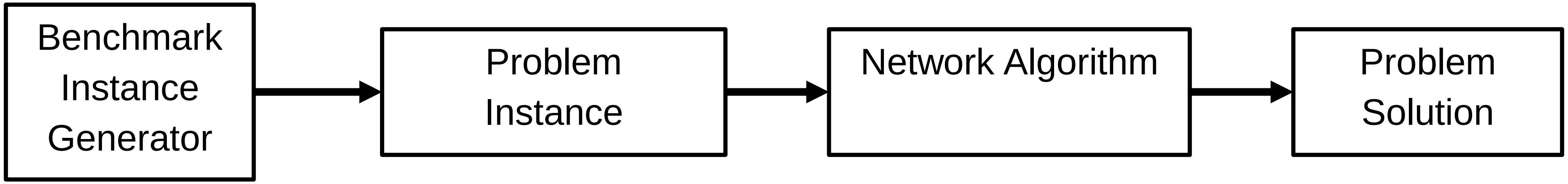
Models

Human‘s Best Guesses

Data-Driven

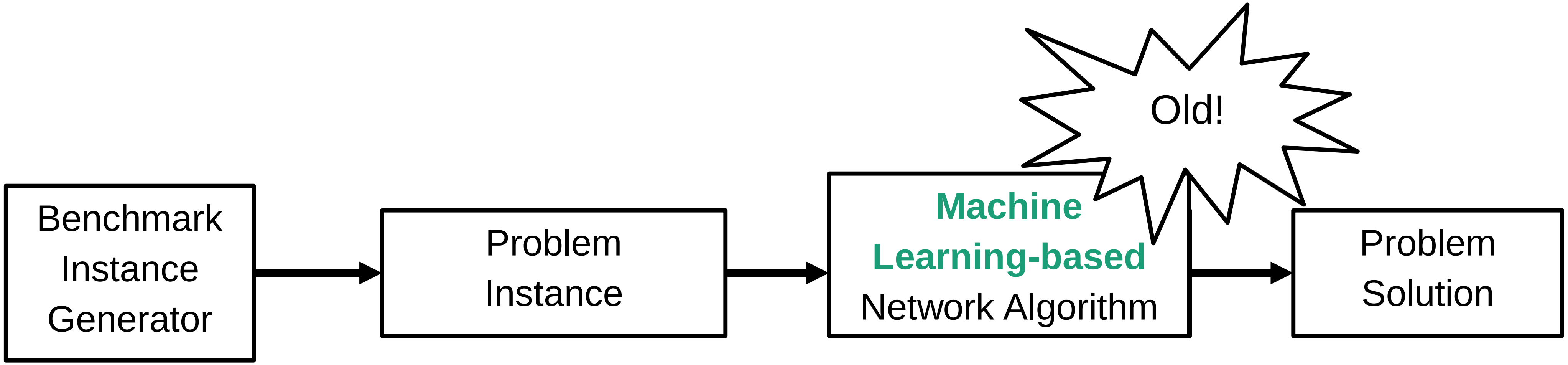
Our idea: Use ML to automatically find adversarial input to benchmark legacy and self-driving networks

Towards Automated Network Optimization and Design



The Traditional Way!

Towards Automated Network Optimization and Design



O’zapf t is [BIG DAMA’17]

Empowerement [SelfDN’18]

ISMAEL [TNSM’19]

Towards Automated Network Optimization and Design

ML/AI

Benchmark Instance Generator

→ **Adversarial** Problem Instance → **Machine Learning-based** Network Algorithm → Problem Solution

challenge

NetBOA [NetAI'19]

TOXIN [CoNEXT'19]

Towards Automated Network Optimization and Design

ML/AI → Benchmark Instance Generator → **Adversarial** Problem Instance → (challenge) → **Machine Learning-based** Network Algorithm → Problem Solution

NetBOA [NetAI'19]

TOXIN [CoNEXT'19]

ML/AI vs ML/AI and Human

Towards Automated Network Optimization and Design

This talk!

ML/AI

Benchmark Instance Generator

→ **Adversarial** Problem Instance → **Machine Learning-based** Network Algorithm → Problem Solution

challenge

NetBOA [NetAI'19]

TOXIN [CoNEXT'19]

ML/AI vs ML/AI and Human

Towards Automated Network Optimization and Design

This talk!

ML/AI

Benchmark Instance Generator

Fast and efficient!

NetBOA [NetAI'19]

TOXIN [CoNEXT'19]

Adversarial Problem Instance → (challenge) → **Machine Learning-based** Network Algorithm → Problem Solution

ML/AI vs ML/AI and Human

Data-Driven Adversarial Network Benchmarking in Data Centers: (1) NetBOA and (2) TOXIN

NetBOA: Self-Driving Network Benchmarking

Johannes Zerwas, Patrick Kalmbach, Laurenz Henkel
Technical University of Munich, Germany

Gábor Rétvári
Budapest University of Technology and Economics, Hungary

Wolfgang Kellerer, Andreas Blenk
Technical University of Munich, Germany

Stefan Schmid
Faculty of Computer Science, University of Vienna, Austria

ABSTRACT

Communication networks have not only become a critical infrastructure of our digital society, but are also increasingly complex and hence error-prone. This has recently motivated the study of more automated and "self-driving" networks: networks which measure, analyze, and control themselves in an adaptive manner, reacting to changes in the environment. In particular, such networks hence require a mechanism to recognize potential performance issues.

This paper presents NetBOA, an adaptive and "data-driven" approach to measure network performance, allowing the network to identify bottlenecks and to perform automated what-if analysis, exploring improved network configurations. As a case study, we demonstrate how the NetBOA approach can be used to benchmark a popular software switch, Open vSwitch. We report on our implementation and evaluation, and show that NetBOA can find performance issues efficiently, compared to a non-data-driven ap-

1 INTRODUCTION

Motivated by the complex, manual, and error-prone operation of today's communication networks, as well as the increasing dependability requirements in terms of availability and performance, the network community is currently very much engaged in developing more automated approaches to manage and operate networks. A particularly interesting vision in this context are *self-driving networks* [10, 17]: rather than aiming for specific optimizations for certain protocols and objectives, networks should learn to drive themselves, maximizing *high-level* goals (such as end-to-end latency), in a "context-aware", *data-driven* manner. At the heart of such self-driving networks hence lies the ability to adaptively measure, analyze, and control themselves. While over the last years, many interesting first approaches have been proposed related to how self-driving networks can control themselves [4, 10, 16], less is known today about how self-driving networks can analyze and

Adversarial Network Algorithm Benchmarking

Sebastian Lettner
TU München
sebastian.lettner@tum.de

Andreas Blenk
TU München
andreas.blenk@tum.de

ABSTRACT

Most research papers should have one thing in common: a clear and expressive evaluation of proposed solutions to problems. However, evaluating solutions is interestingly a challenging task: when using human-constructed examples or real-world data, it is difficult to assess to which degree the data represents the input spectrum also of future demands. Moreover, evaluations which fail to show generalization might hide algorithm weak-spots, which could eventually lead to reliability and security issues later on. To solve this problem we propose Toxin, a framework for automated, data-driven benchmarking of, e.g., network algorithms. In a first proof-of-concept implementation, we use Toxin to generate challenging traffic data-sets for a data center networking use case.

CCS CONCEPTS

• **Networks** → *Traffic engineering algorithms*; *Network simulations*; *Network performance analysis*; • **Computing methodologies** → *Machine learning*; Artificial intelligence.

KEYWORDS

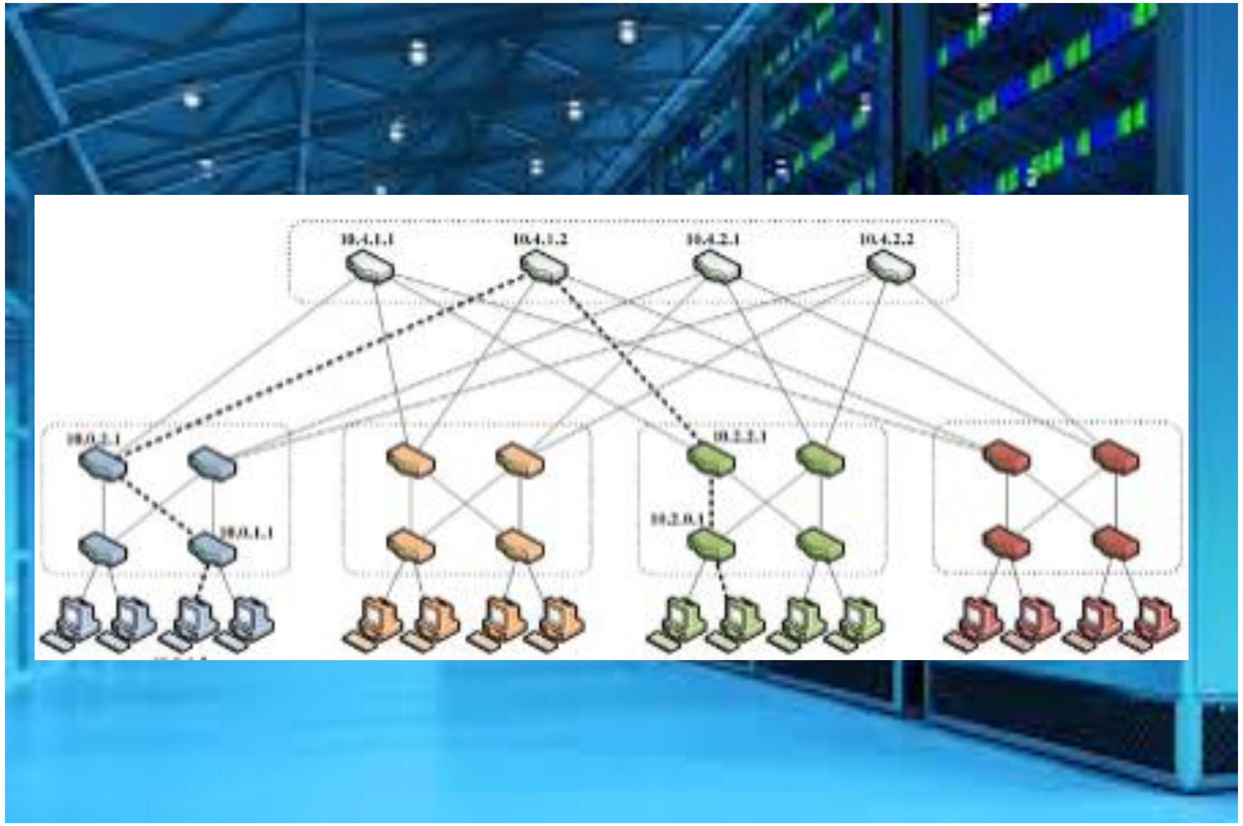
adversarial traffic generation, artificial intelligence, data center

Unfortunately, obtaining challenging input data is a problem of its own. Even human experts are often not able to construct inputs exposing these weaknesses [5, 7, 12], not at least because of the high effort it takes. As a consequence, evaluations might sometimes be biased and actually fail to show generalization. This is, however, problematic since overlooked performance issues can have negative implications not only on the reliability but also on the security of the system [9, 14] because it could open the door for exploitation.

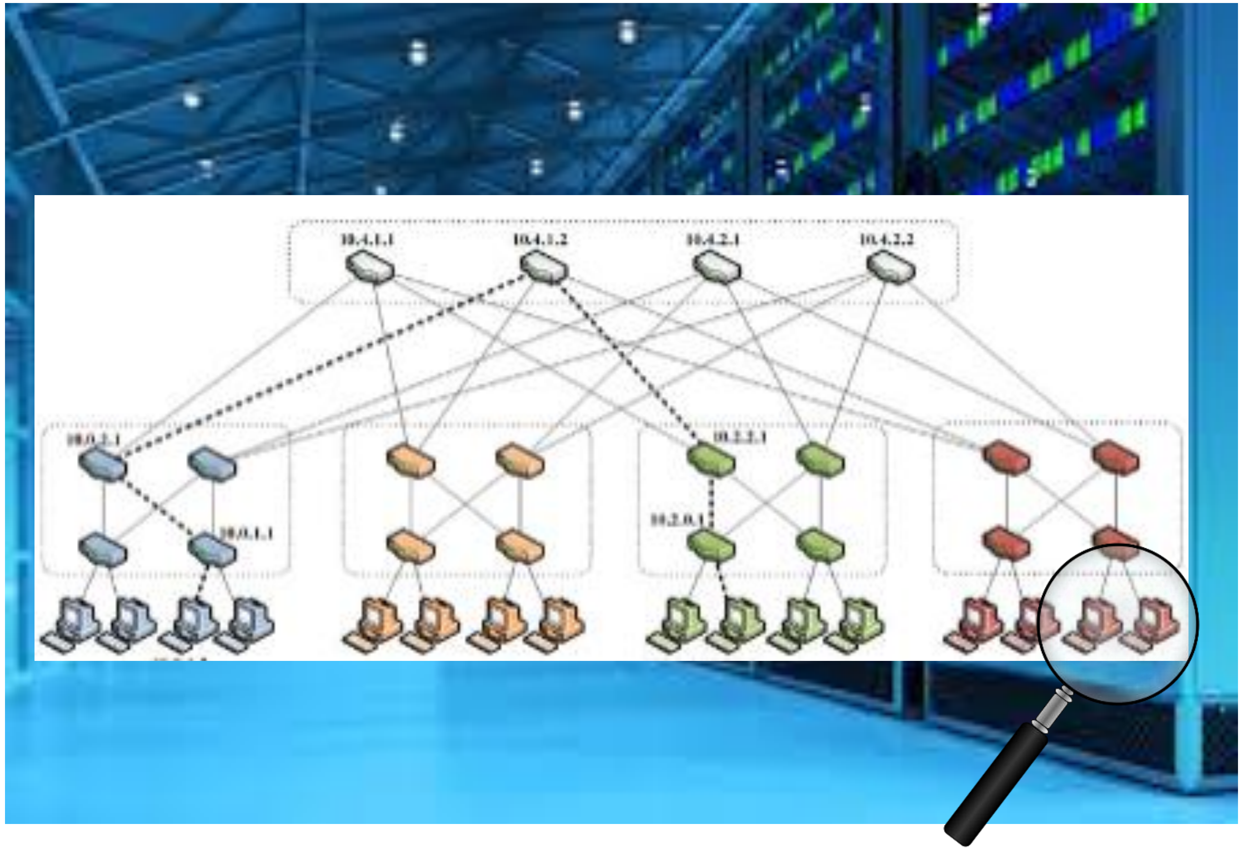
To address this problem we propose Toxin, an automated, data-driven benchmarking framework for data center network algorithms. We demonstrate that creating challenging evaluation data sets is a suitable task for machine learning and artificial intelligence. Those machine generated data-sets consist, e.g., of traffic matrices (demands), which are trained to maximize certain network metrics, e.g. the Flow Completion Time (FCT) in data centers. Using an automated, data-aware and unified way of benchmarking (i.e., attacking) algorithms, evaluation becomes more representative and even reproducibility might be simplified.

Previous work on algorithm complexity attacks has already shown methods for generating challenging, often called adversary, algorithms inputs [8, 10, 13, 17, 18]. With the help of these inputs the authors were able to improve algorithm performance and close

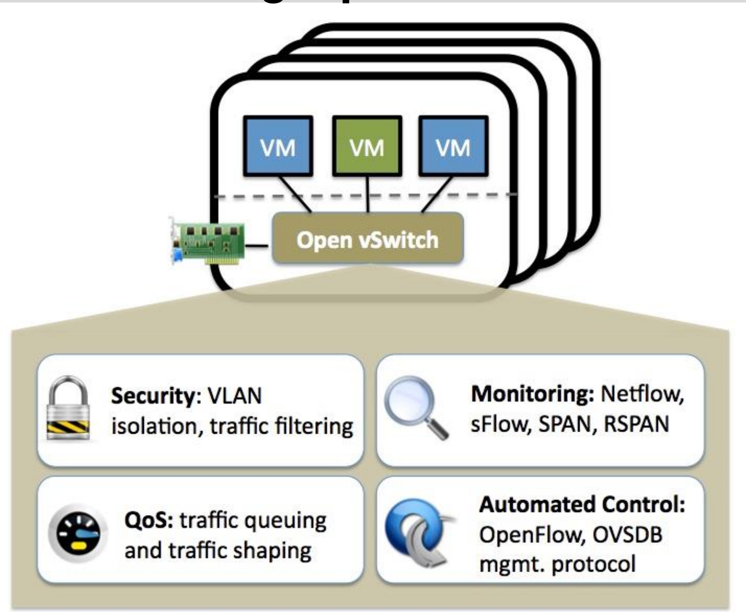
Use Cases of This Talk: Data Center Network Benchmarking



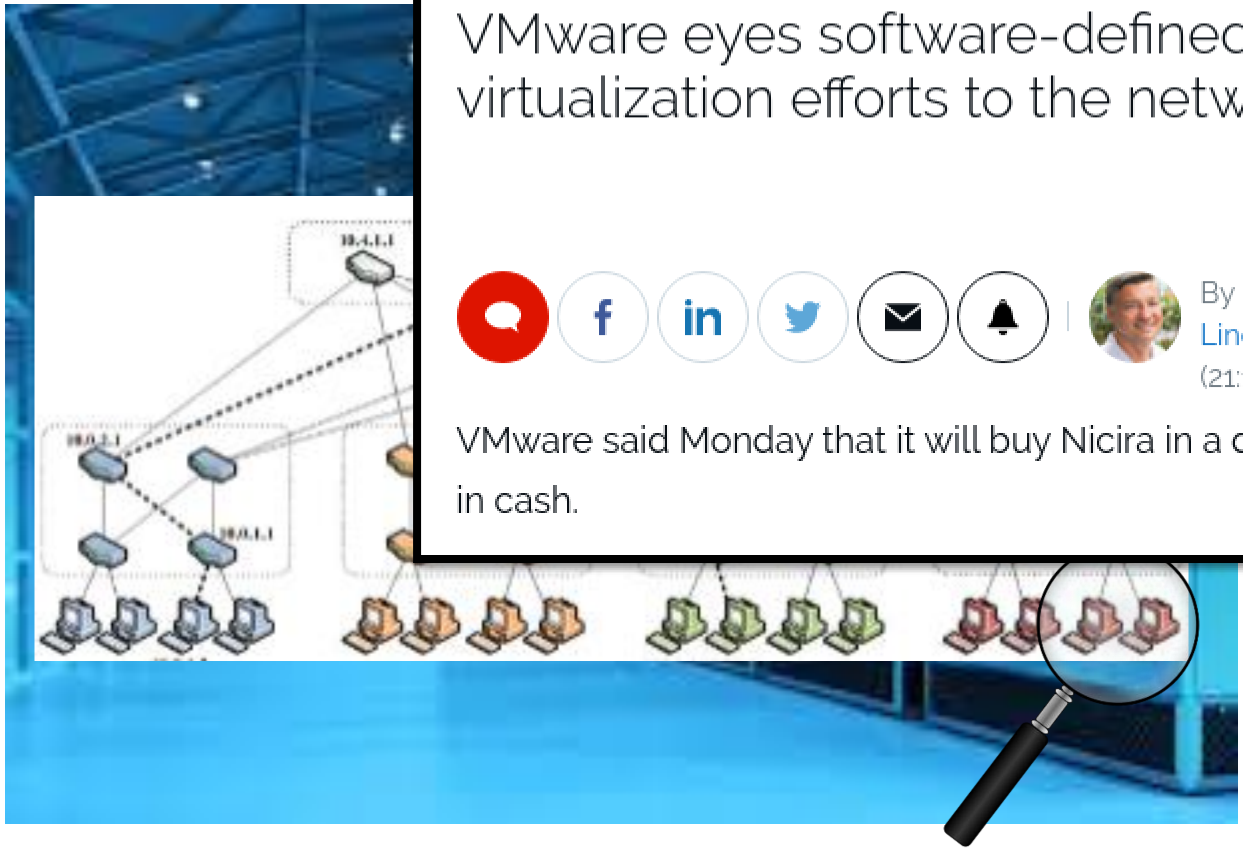
Use Cases of This Talk: Data Center Network Benchmarking



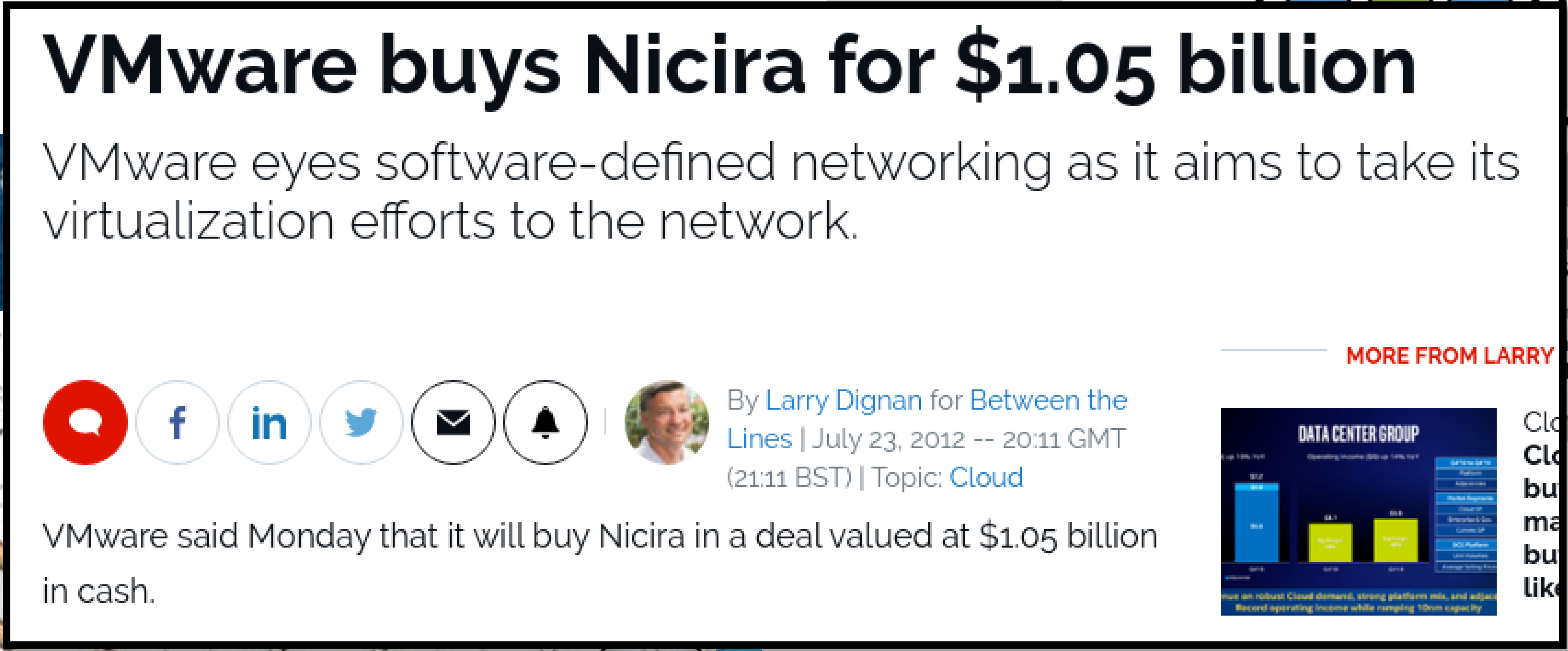
(1) Benchmarking Open vSwitch: NetBOA



Use Cases of This Talk: Data Center Network Benchmarking



(1) Benchmarking Open vSwitch: NetBOA



VMware buys Nicira for $1.05 billion

VMware eyes software-defined networking as it aims to take its virtualization efforts to the network.

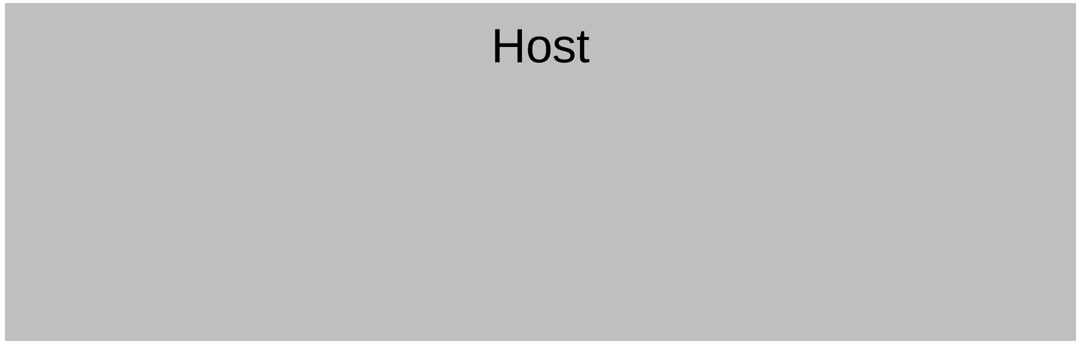
By Larry Dignan for Between the Lines | July 23, 2012 -- 20:11 GMT (21:11 BST) | Topic: Cloud

VMware said Monday that it will buy Nicira in a deal valued at $1.05 billion in cash.

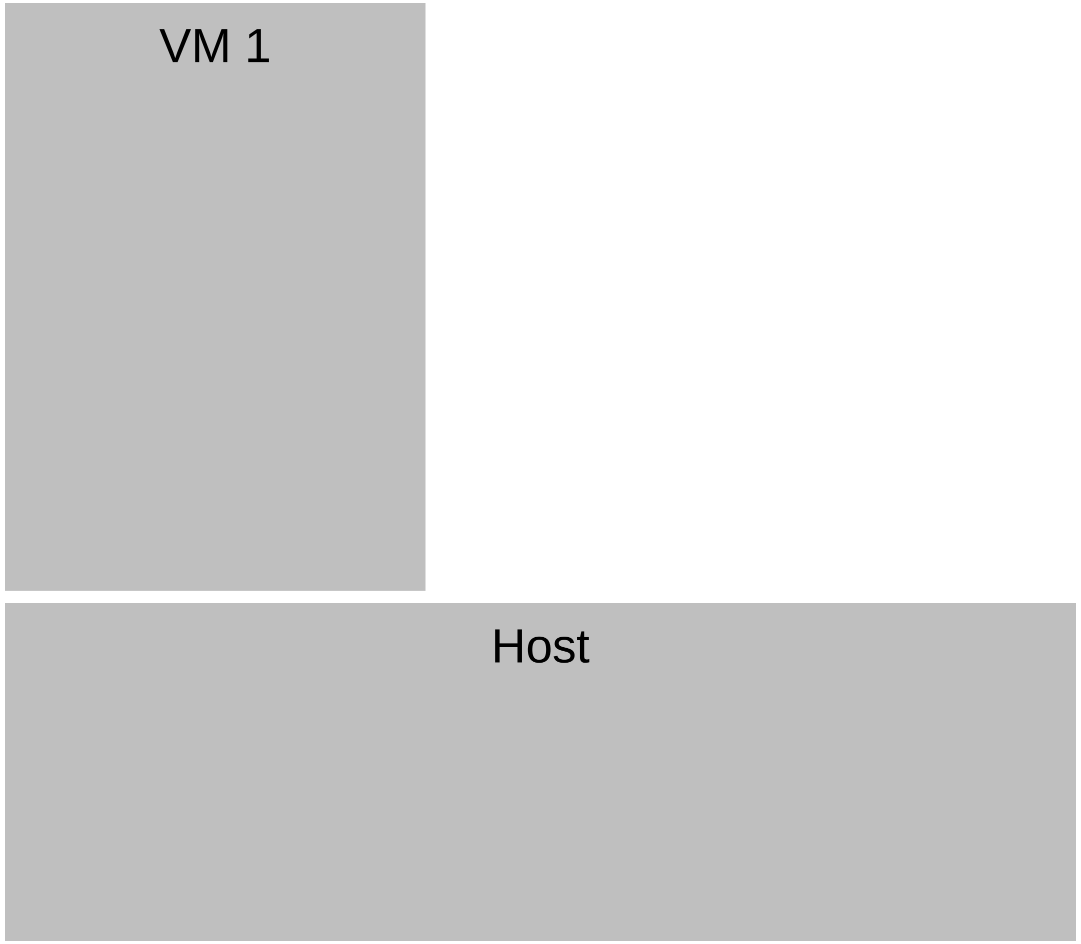
MORE FROM LARRY

Network Traffic Generation in a Real Testbed: Setup

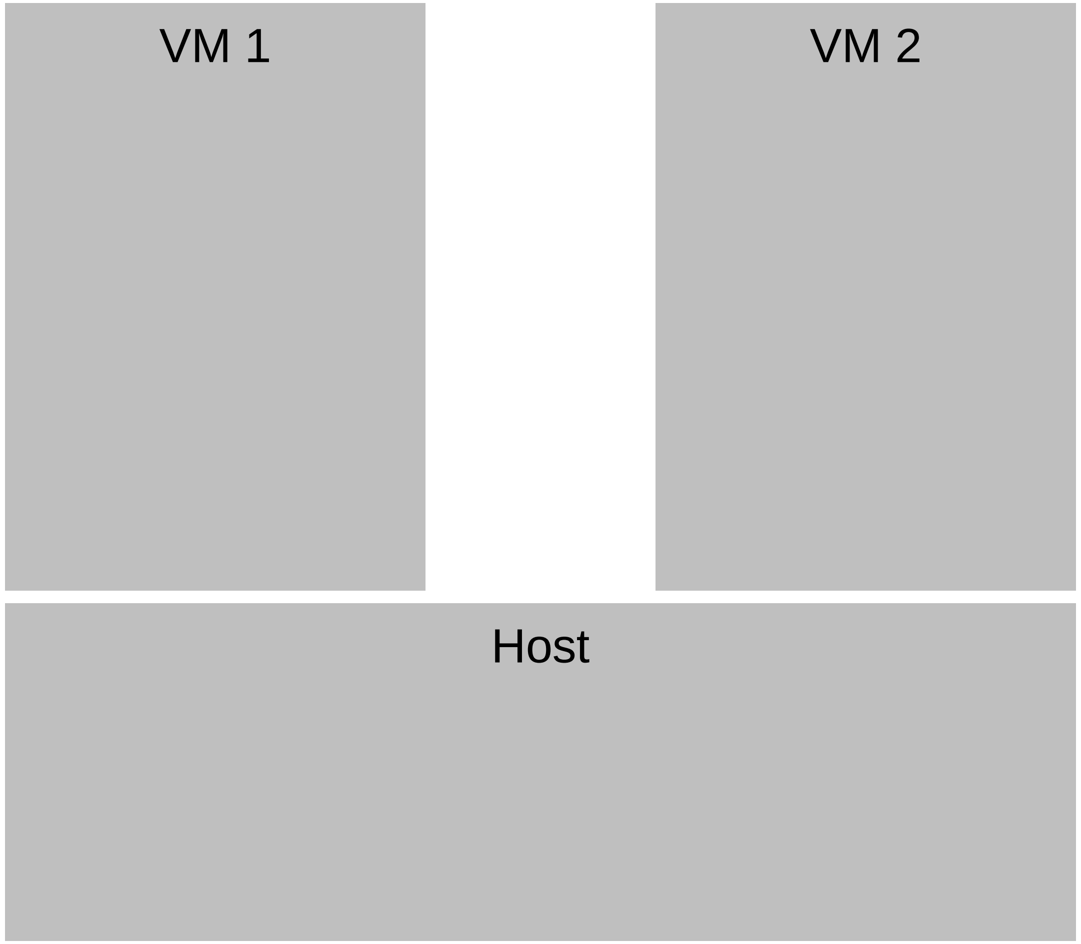
Network Traffic Generation in a Real Testbed: Setup



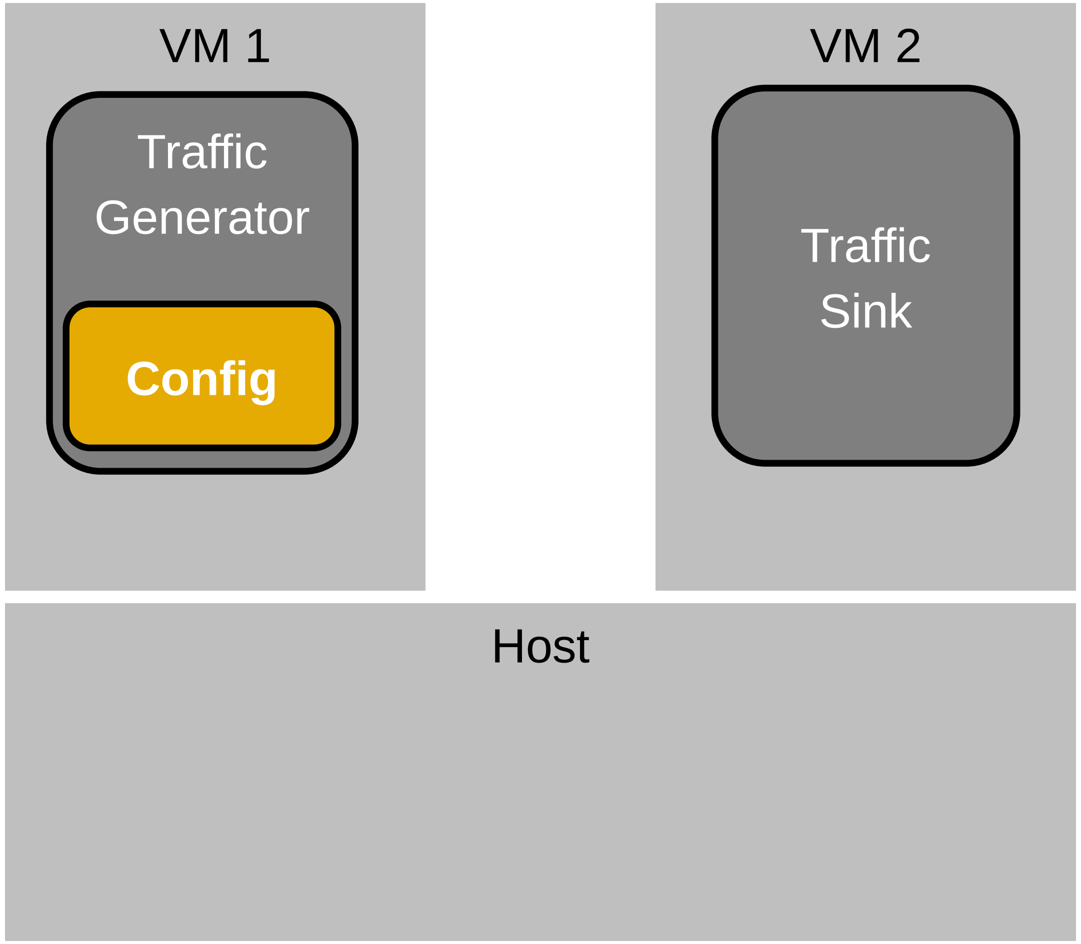
Network Traffic Generation in a Real Testbed: Setup



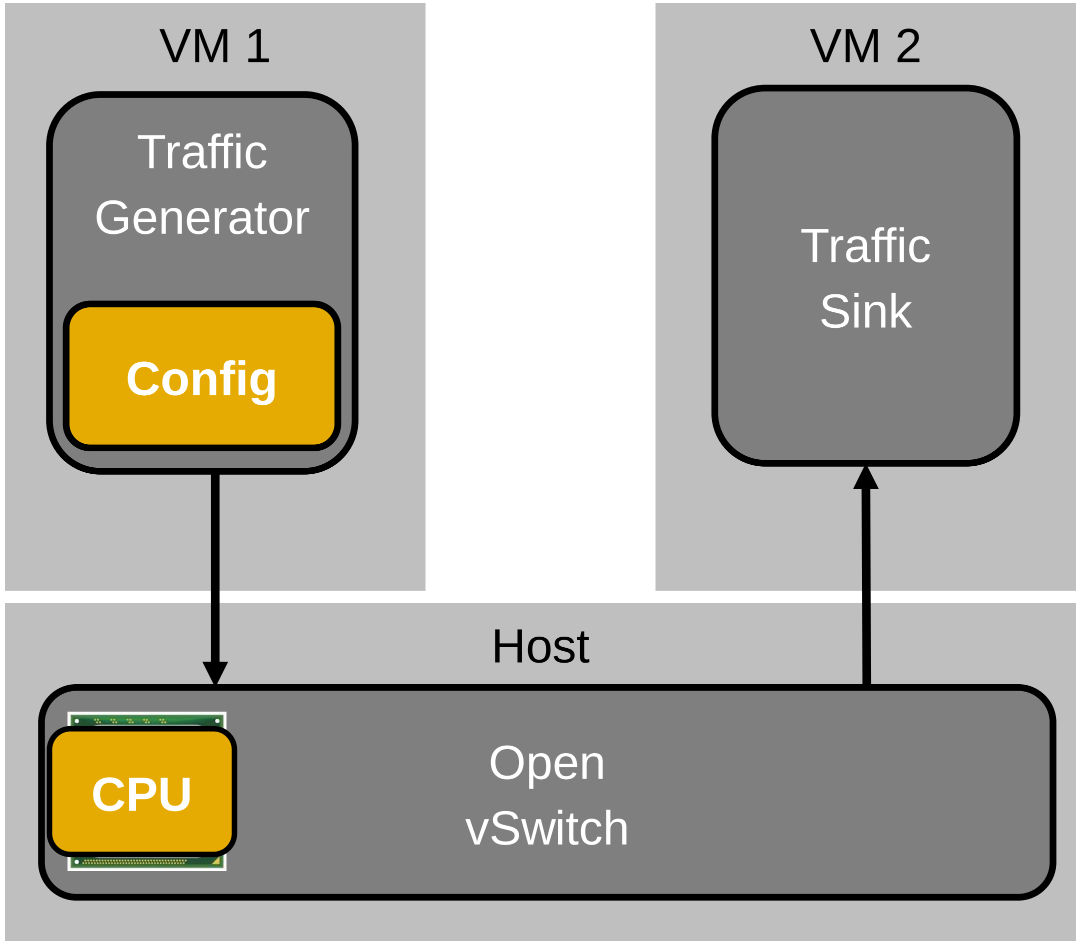
Network Traffic Generation in a Real Testbed: Setup



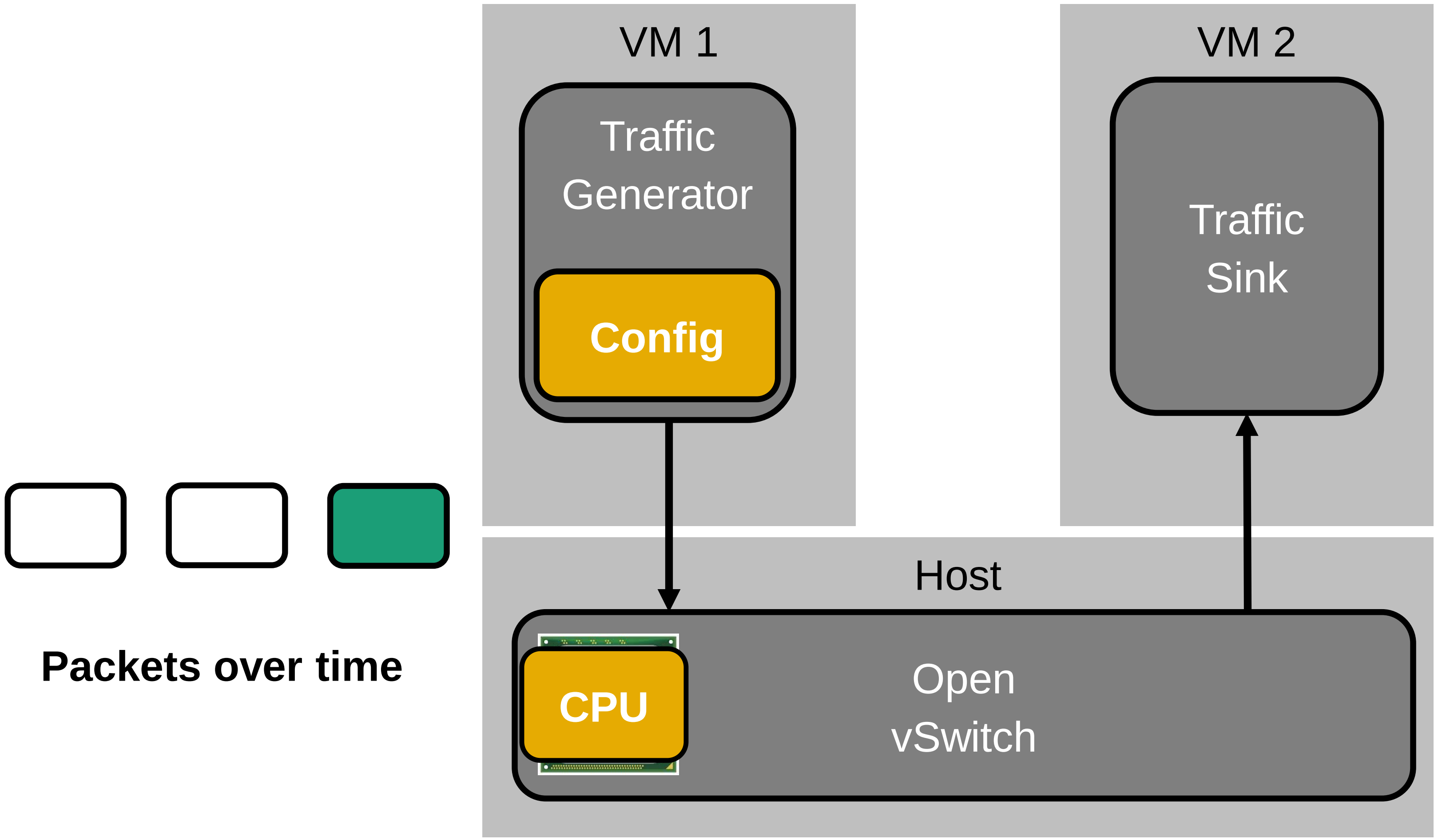
Network Traffic Generation in a Real Testbed: Setup



Network Traffic Generation in a Real Testbed: Setup

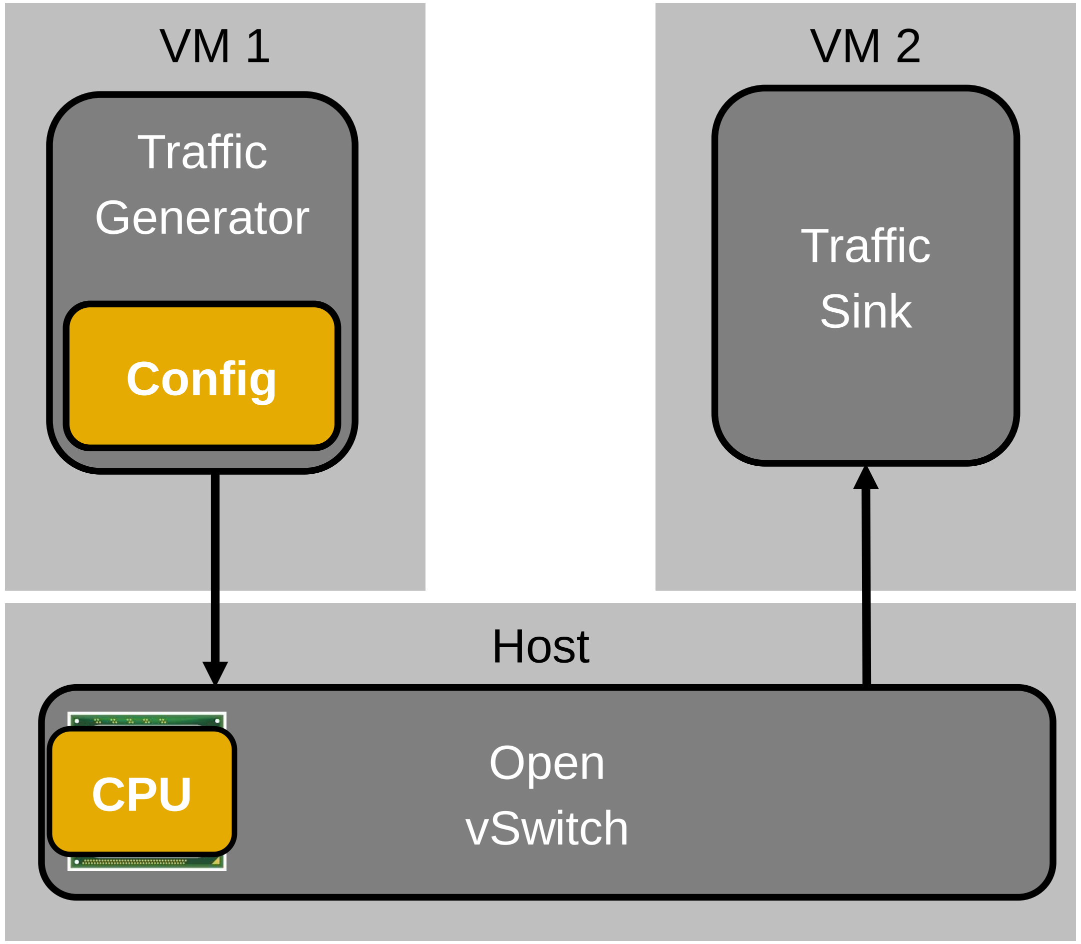


Network Traffic Generation in a Real Testbed: Setup



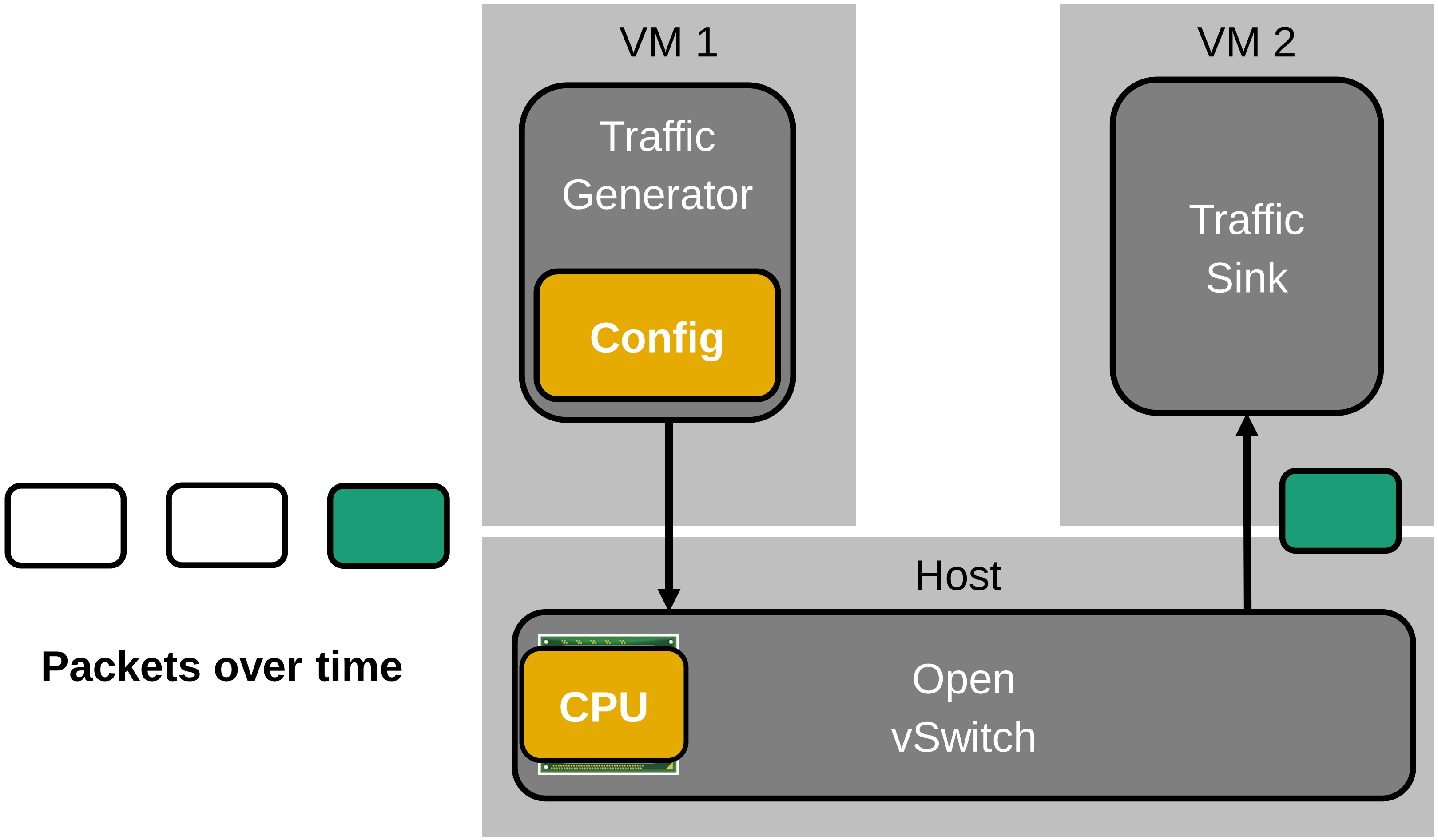
Network Traffic Generation in a Real Testbed: Setup

Packets over time



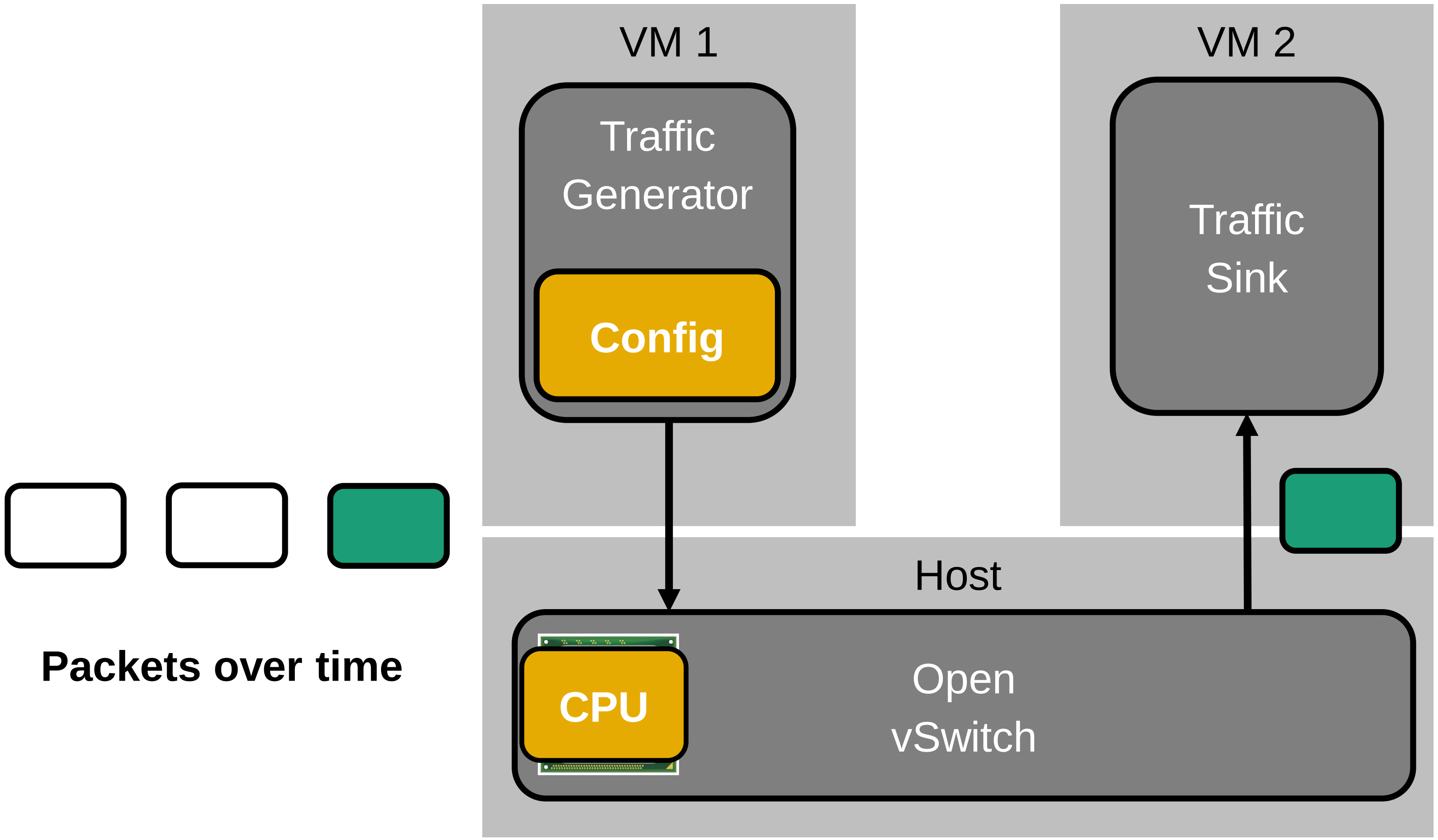
Match	Rule
	Forward
*	DROP

Network Traffic Generation in a Real Testbed: Setup



Match	Rule
	Forward
*	DROP

Network Traffic Generation in a Real Testbed: Setup

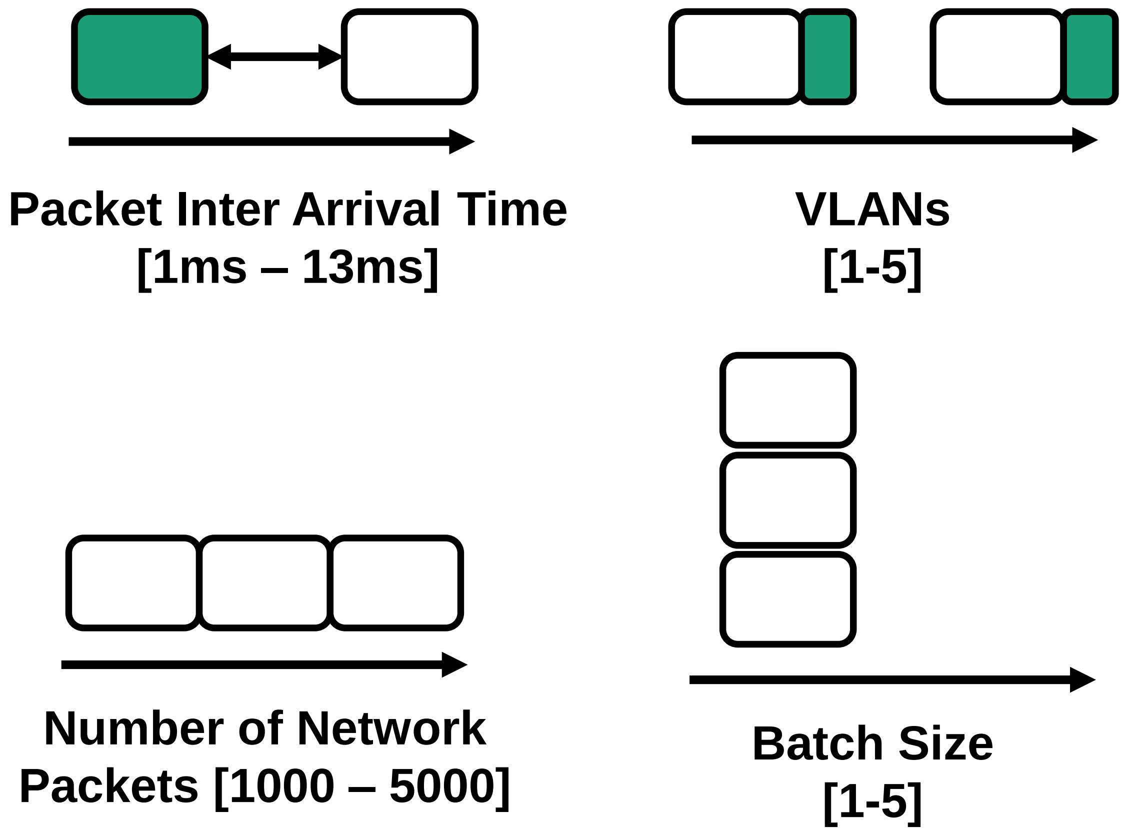


Match	Rule
	Forward
*	DROP

Goal: Find network traffic configuration that maximizes CPU load

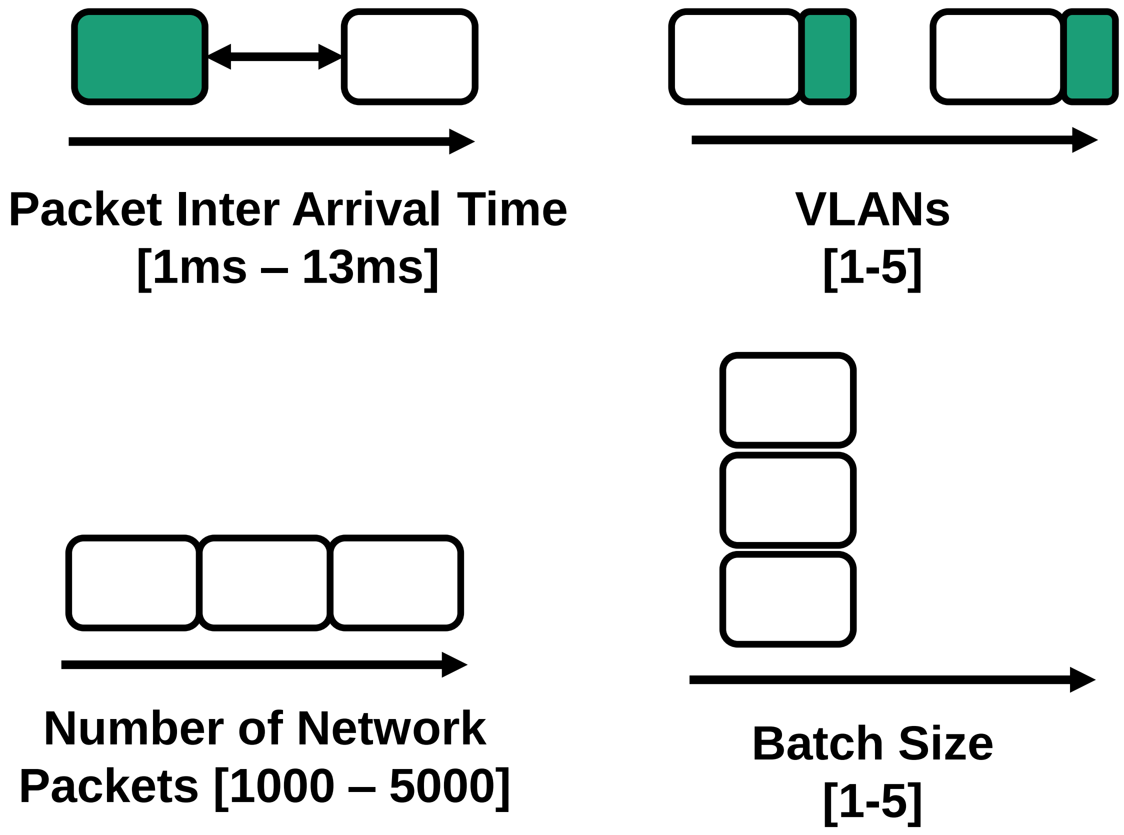
Network Benchmarking is Challenging: Complex and Huge Configuration Space

How many packets to send? How should headers look like? What protocol to use? When to send packets? Etc.

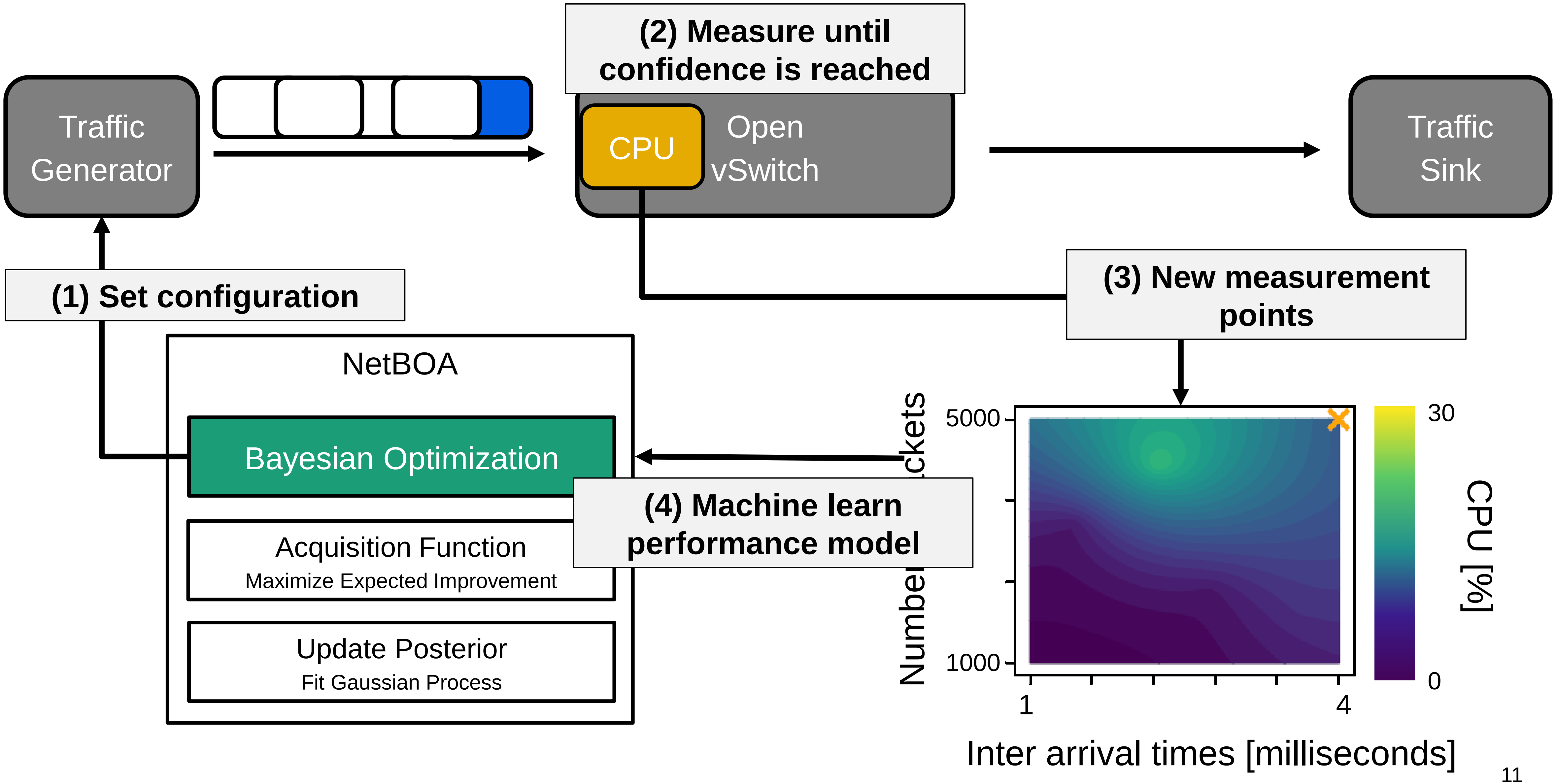


Network Benchmarking is Challenging: Complex and Huge Configuration Space

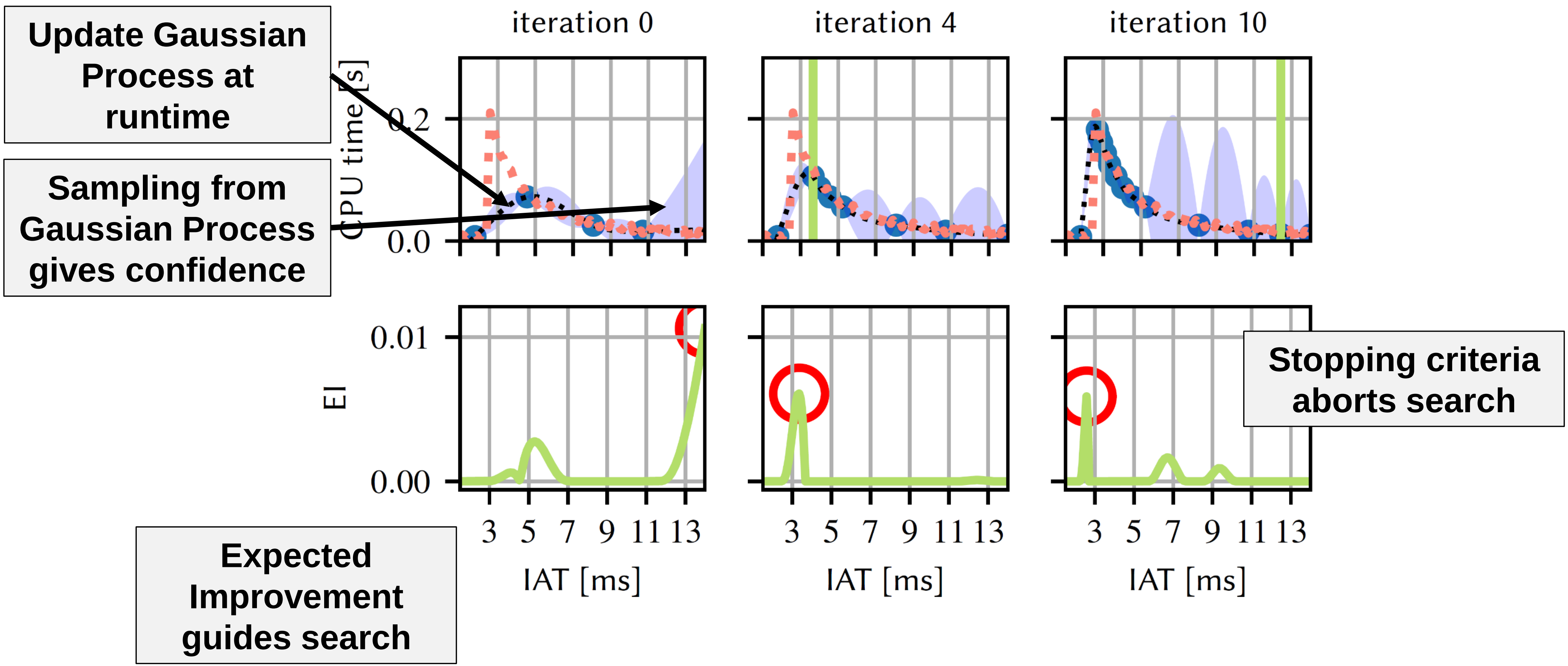
How many packets to send? How should headers look like? What protocol to use? When to send packets? Etc.



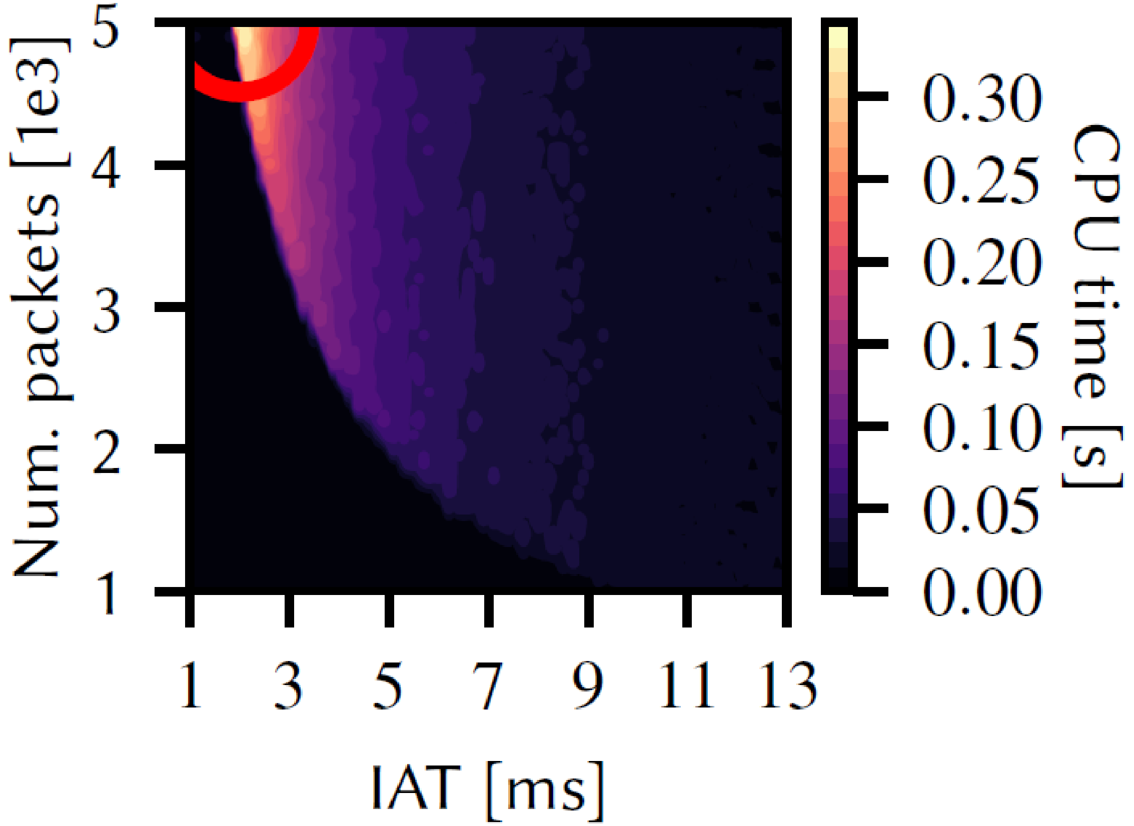
NetBOA: The Bayesian Optimization Measurement Loop



Bayesian Optimization: NetBOA for Inter Arrival Time (IAT) Parameter

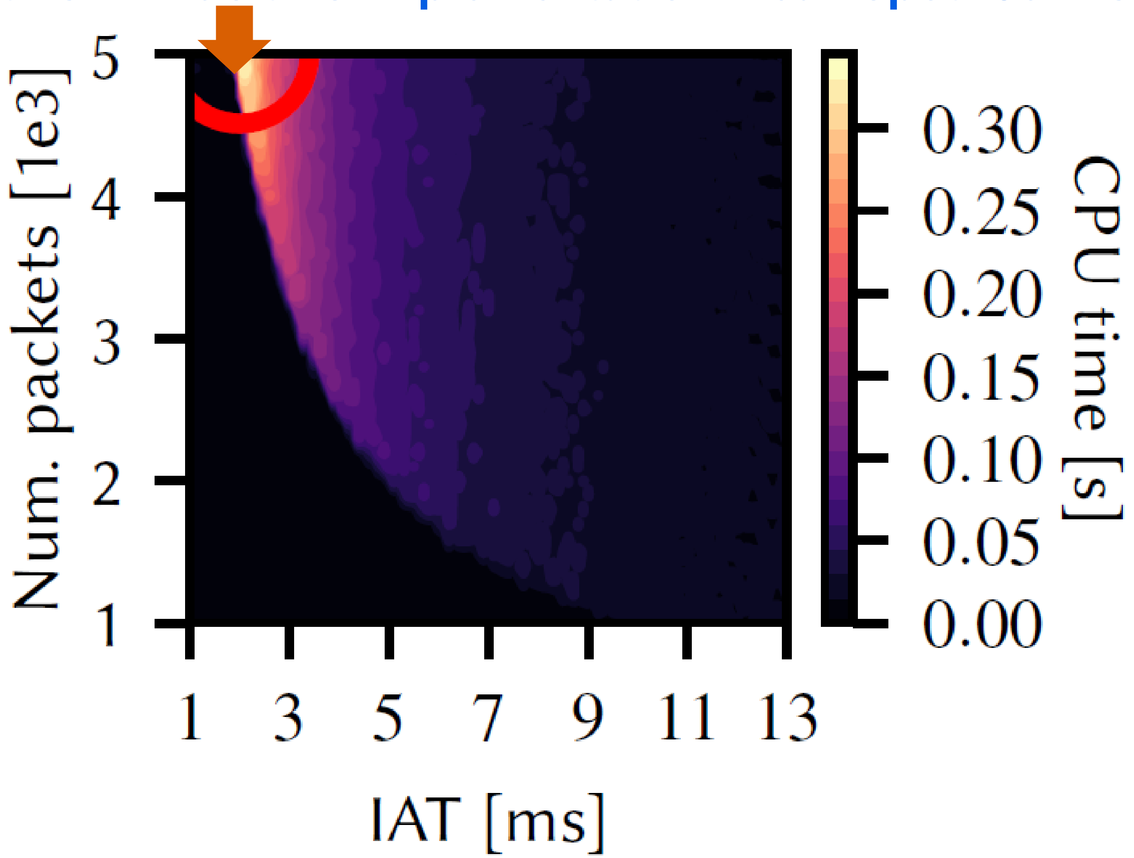


OVS Performance for Number of Packets and Inter-arrival Times



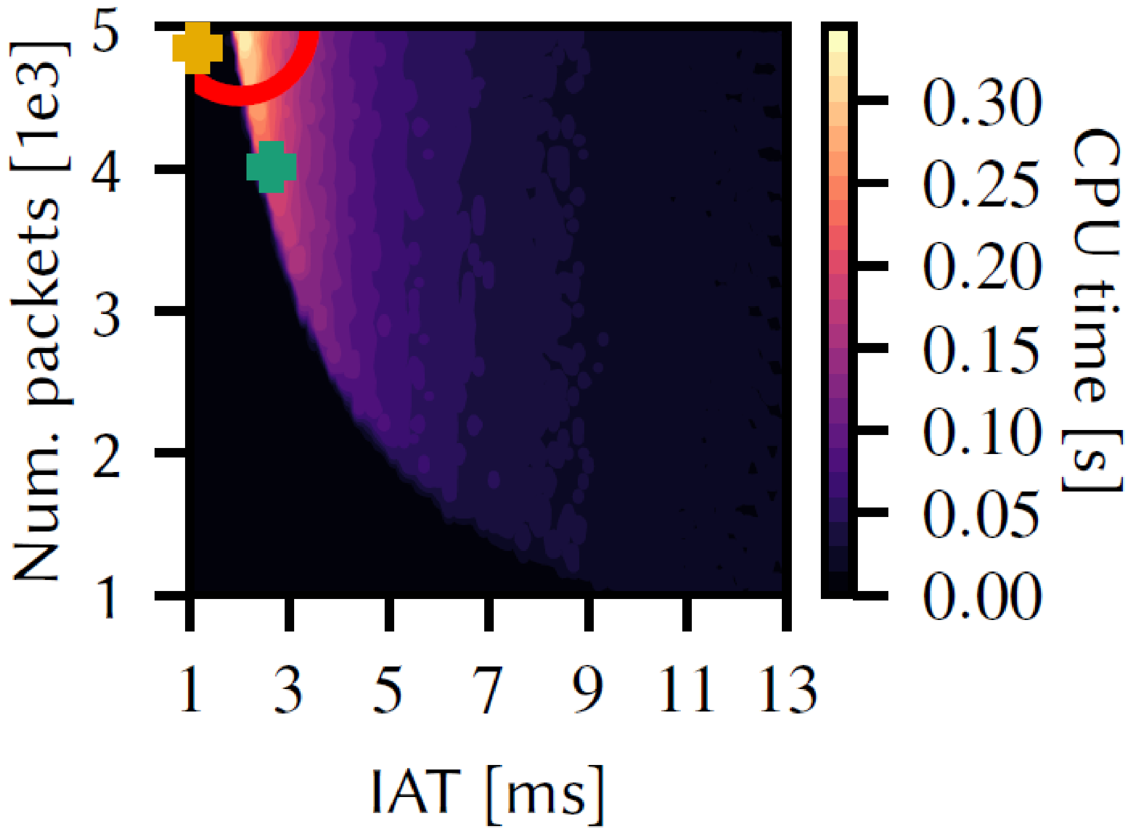
OVS Performance for Number of Packets and Inter-arrival Times

NetBOA finds this implementation weak spot! 30% CPU increase!

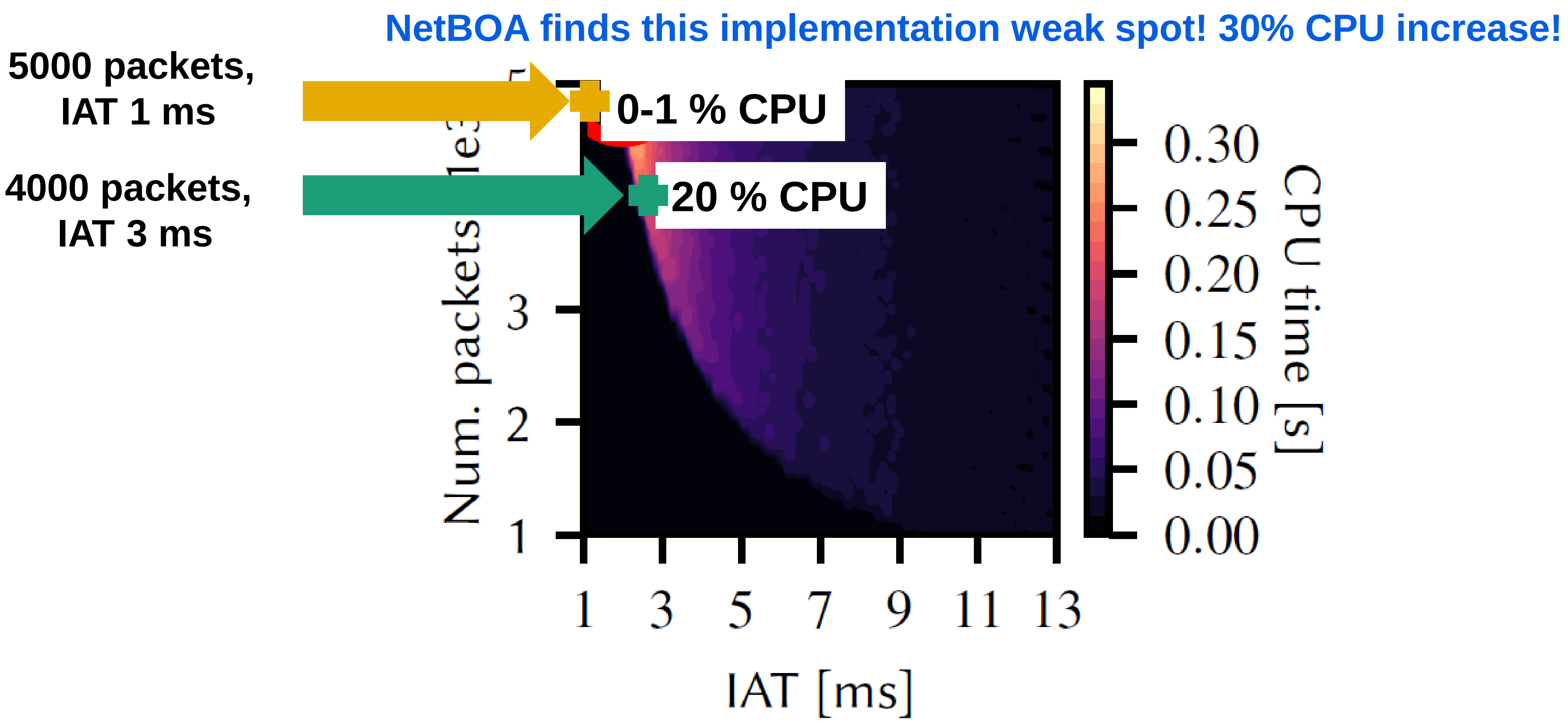


OVS Performance for Number of Packets and Inter-arrival Times

NetBOA finds this implementation weak spot! 30% CPU increase!

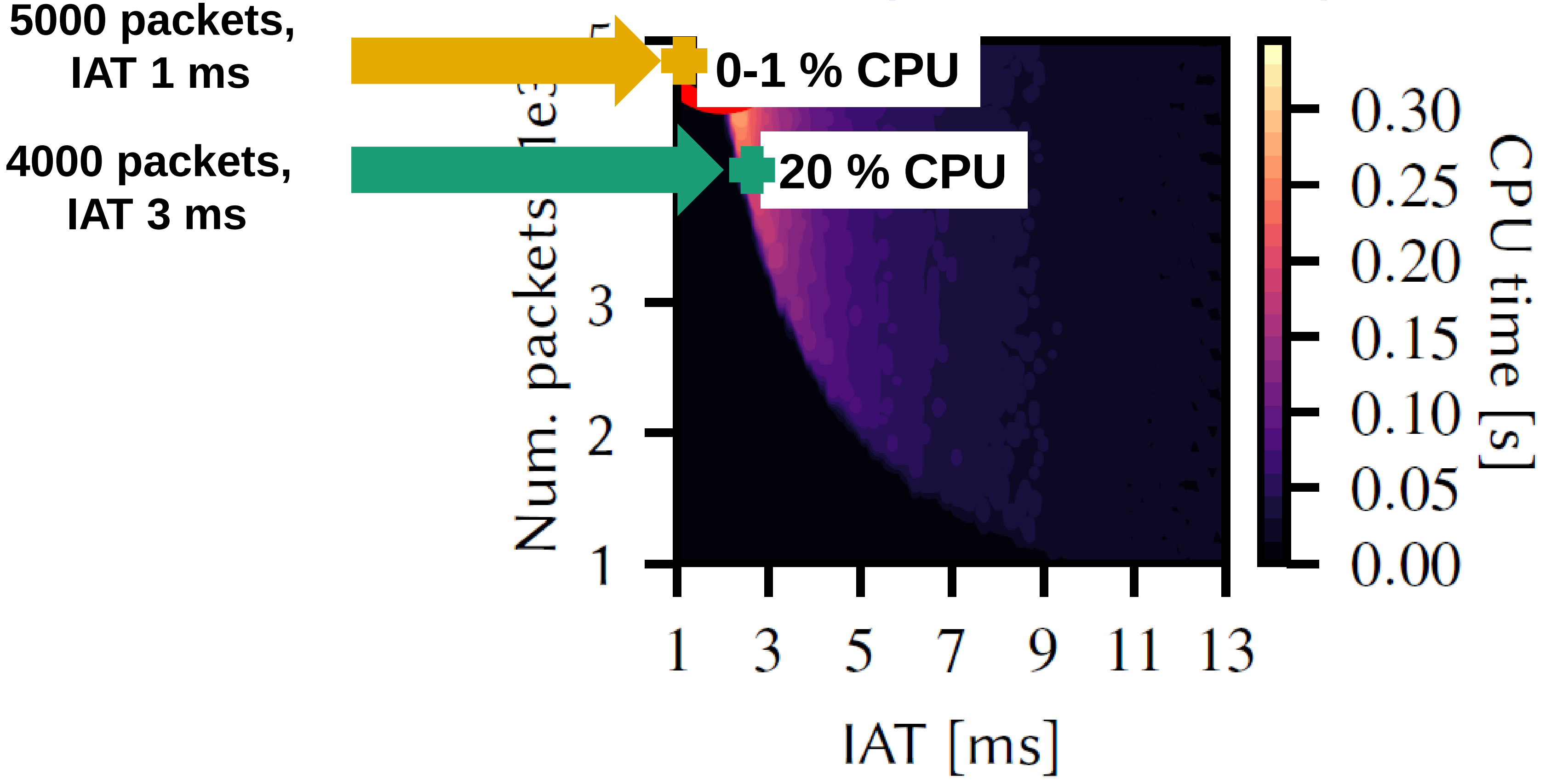


OVS Performance for Number of Packets and Inter-arrival Times



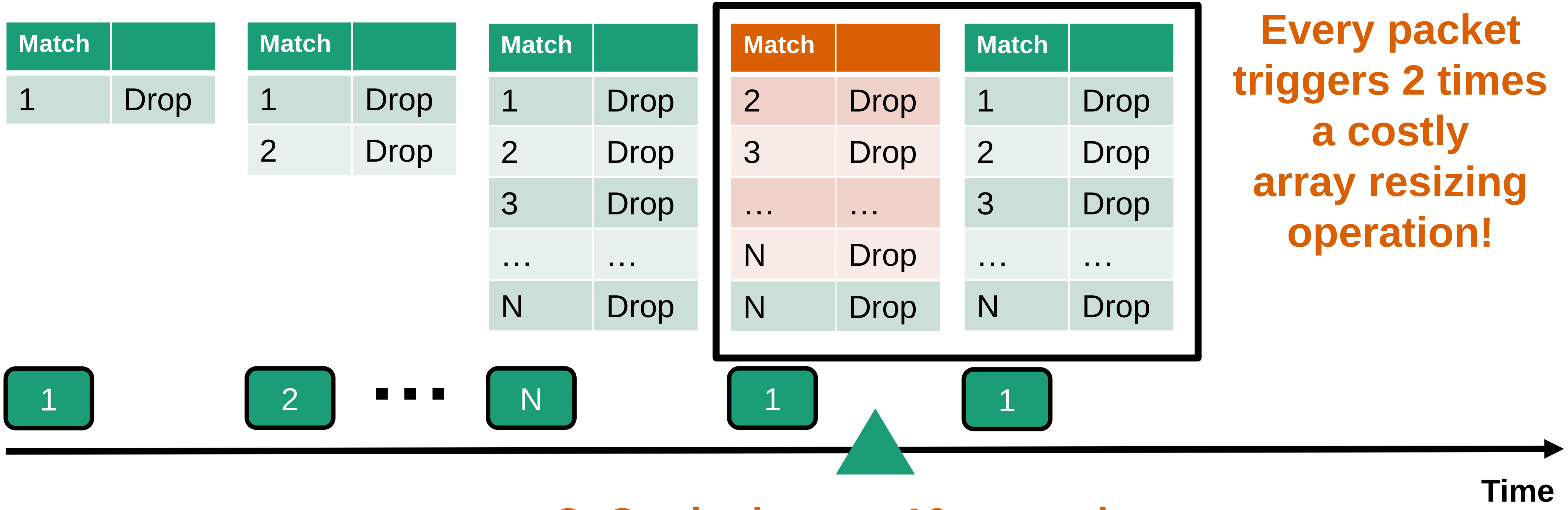
OVS Performance for Number of Packets and Inter-arrival Times

NetBOA finds this implementation weak spot! 30% CPU increase!



- Performance models are non-trivial
- **Surprising:** Sending less network packets over time can lead to significantly higher CPU

Why? Let Us Look At OvS Behavior!

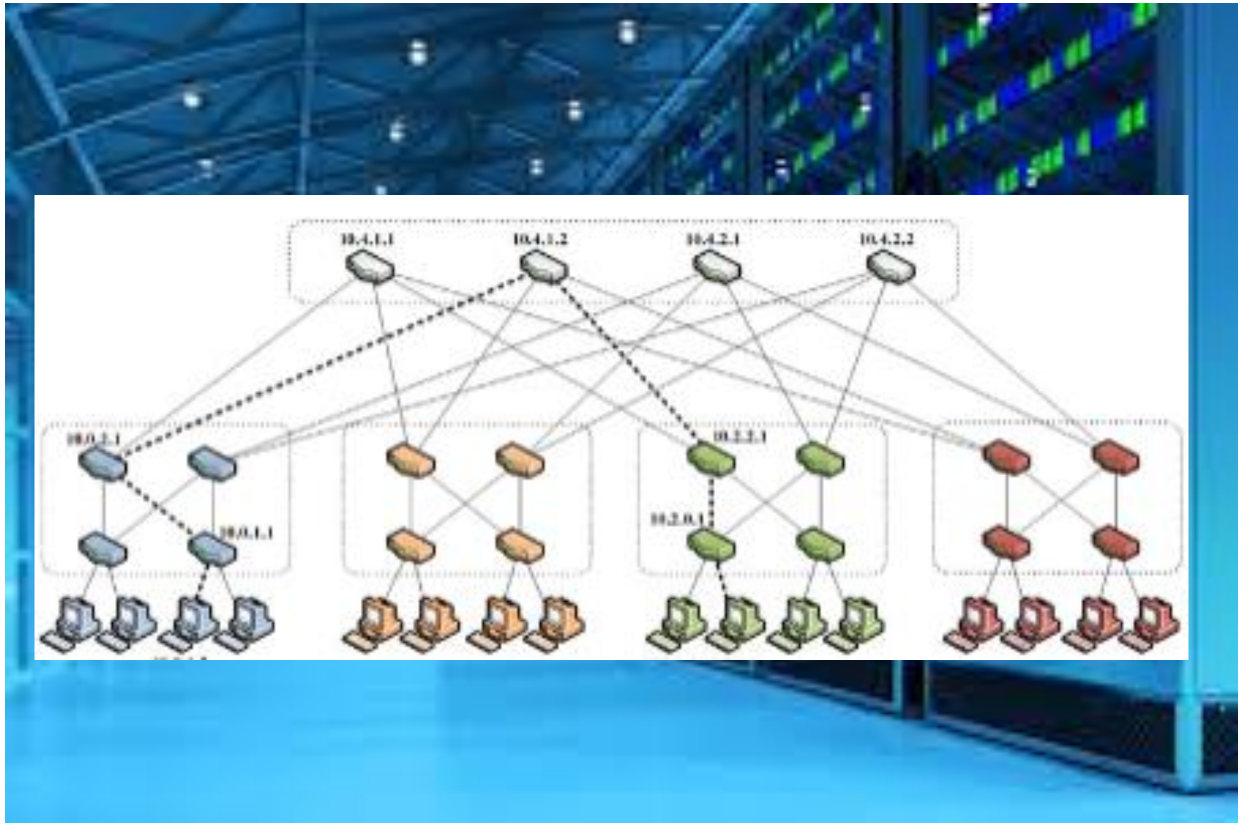


OvS rule timeout 10 seconds

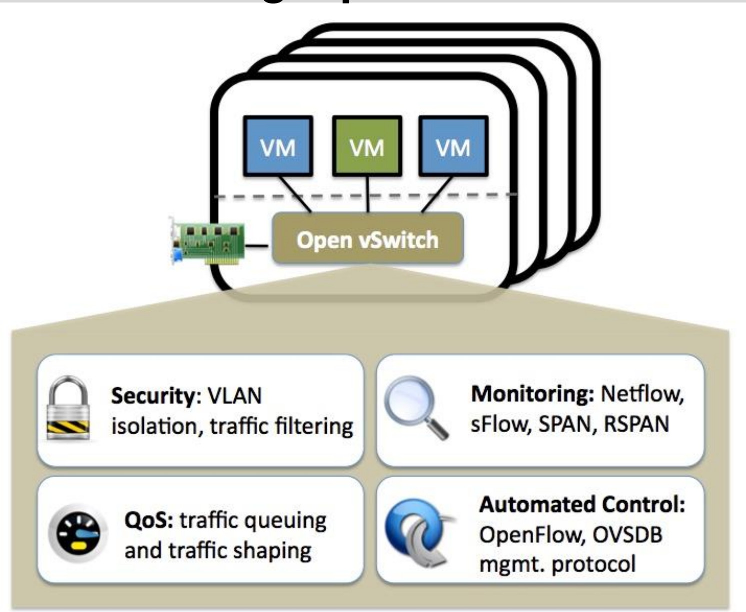
- We are using the OvS switch with the **Megaflow Cache enabled**
- For instance for 5000 packets: We trigger roughly every >2 ms a flow insertion + removal

→ **Forcing OvS to continuously run through the array + resizing it**

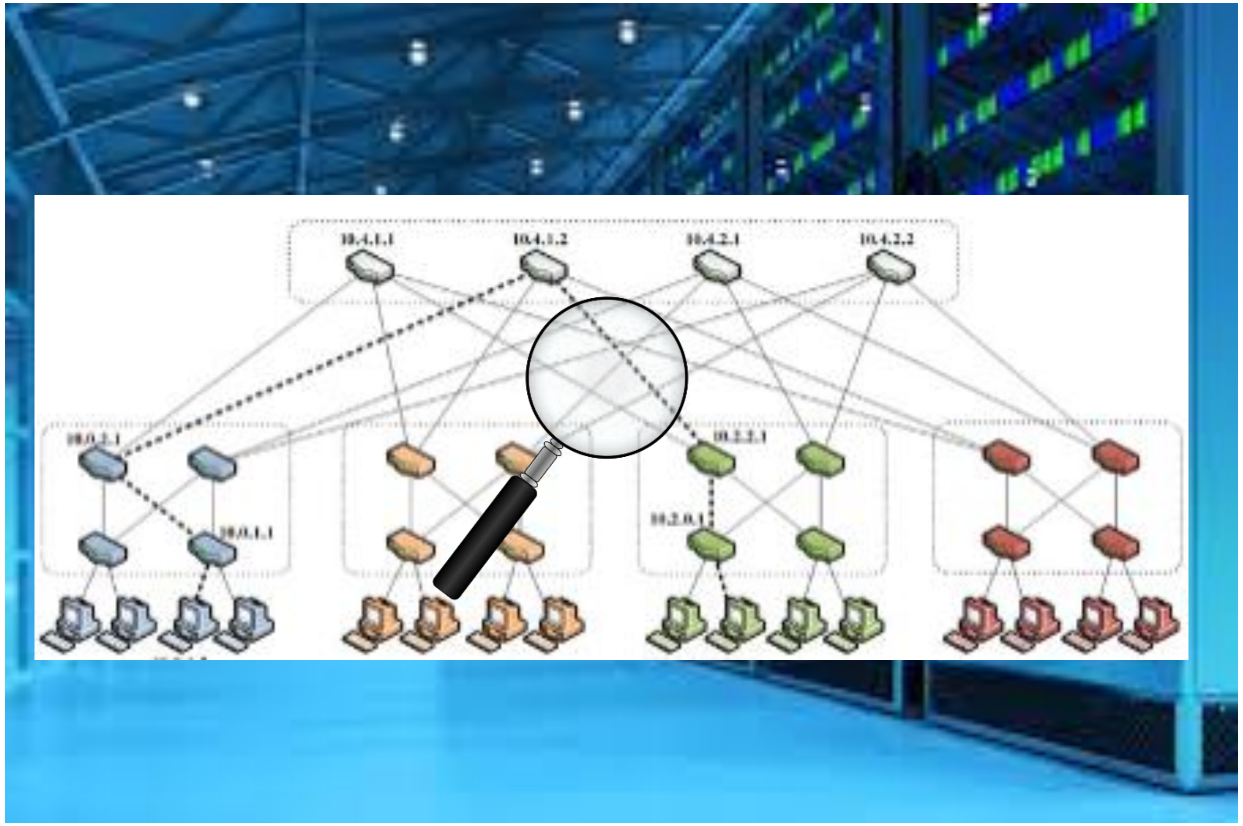
Use Cases of this talk: data center network benchmarking



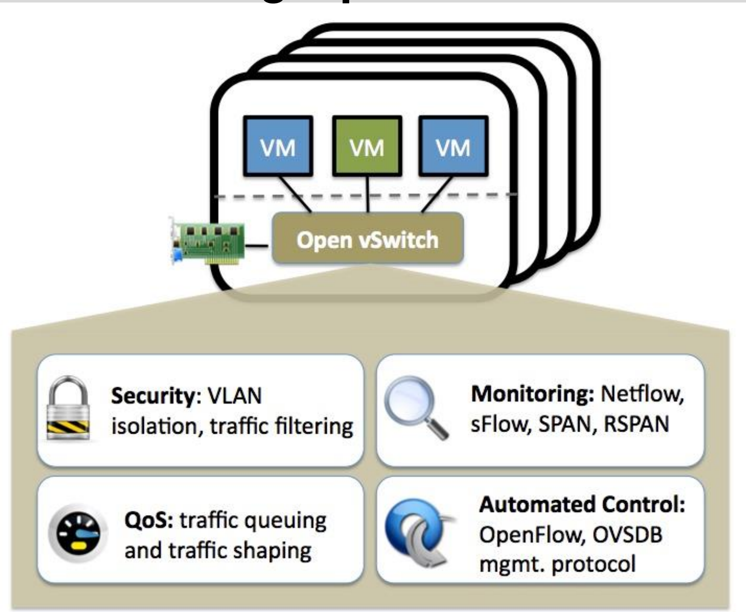
(1) Benchmarking Open vSwitch: NetBOA



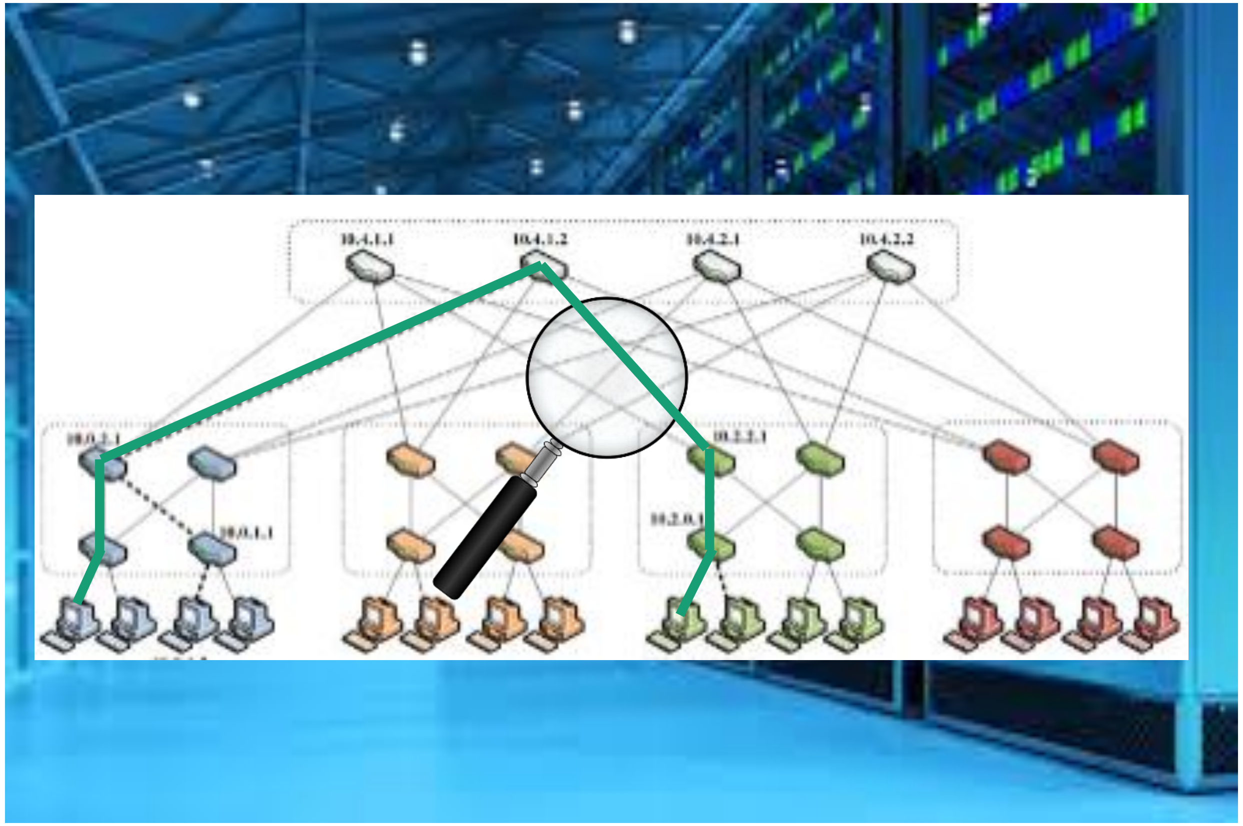
Use Cases of this talk: data center network benchmarking



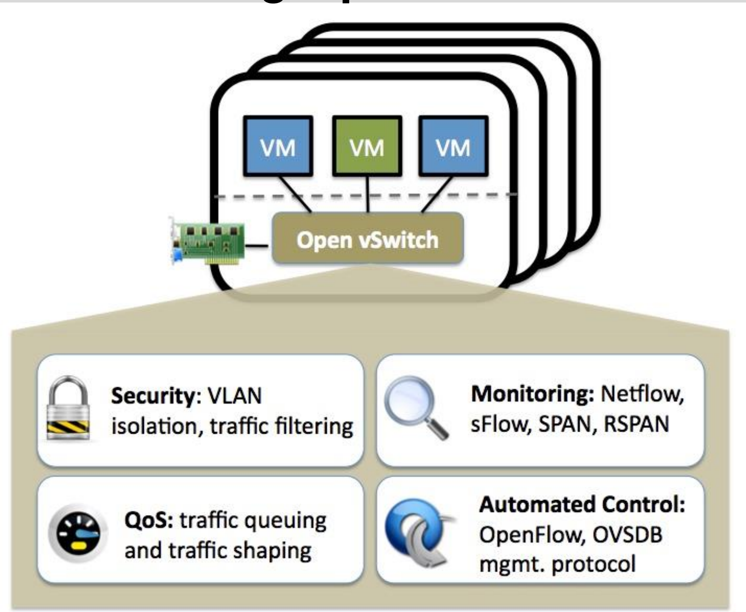
(1) Benchmarking Open vSwitch: NetBOA



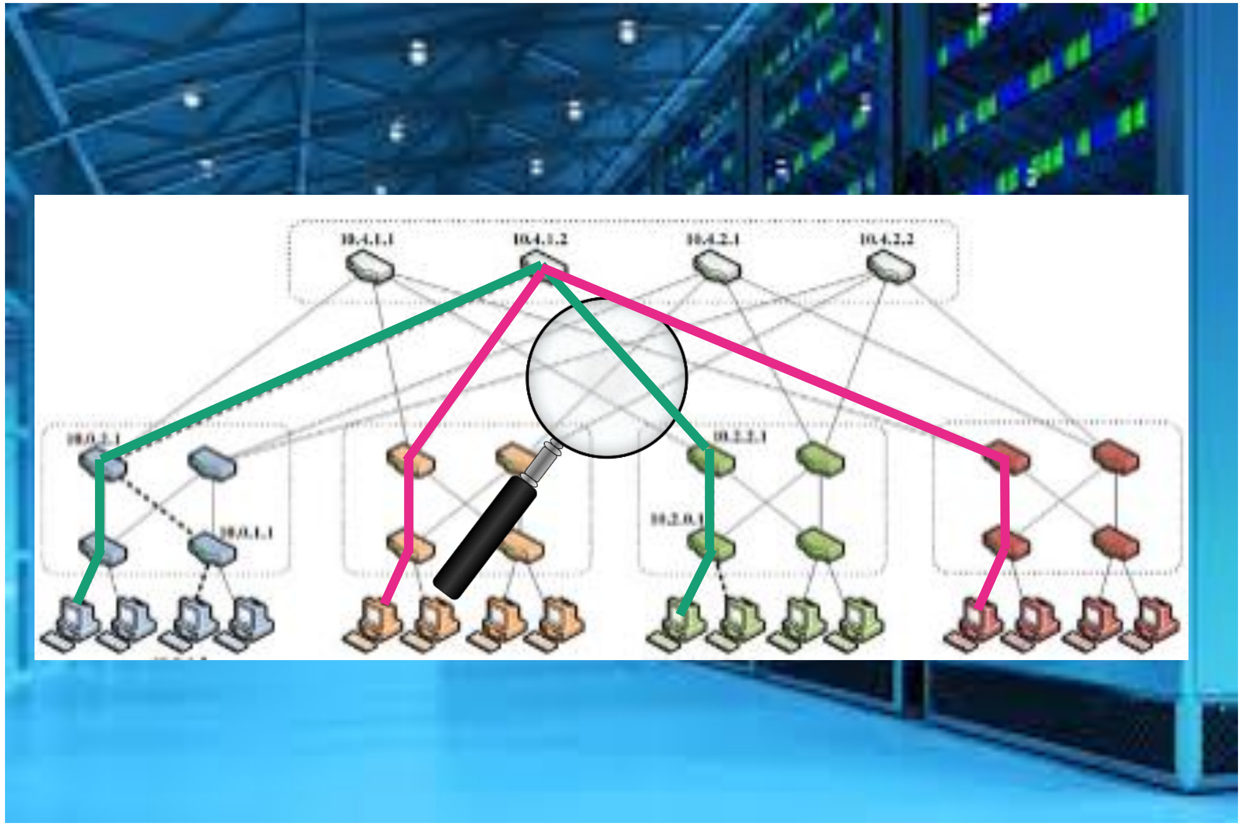
Use Cases of this talk: data center network benchmarking



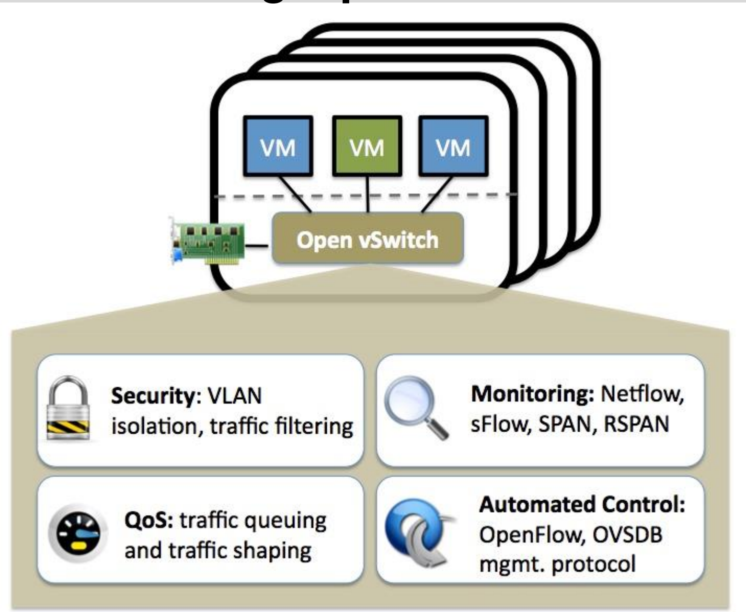
(1) Benchmarking Open vSwitch: NetBOA



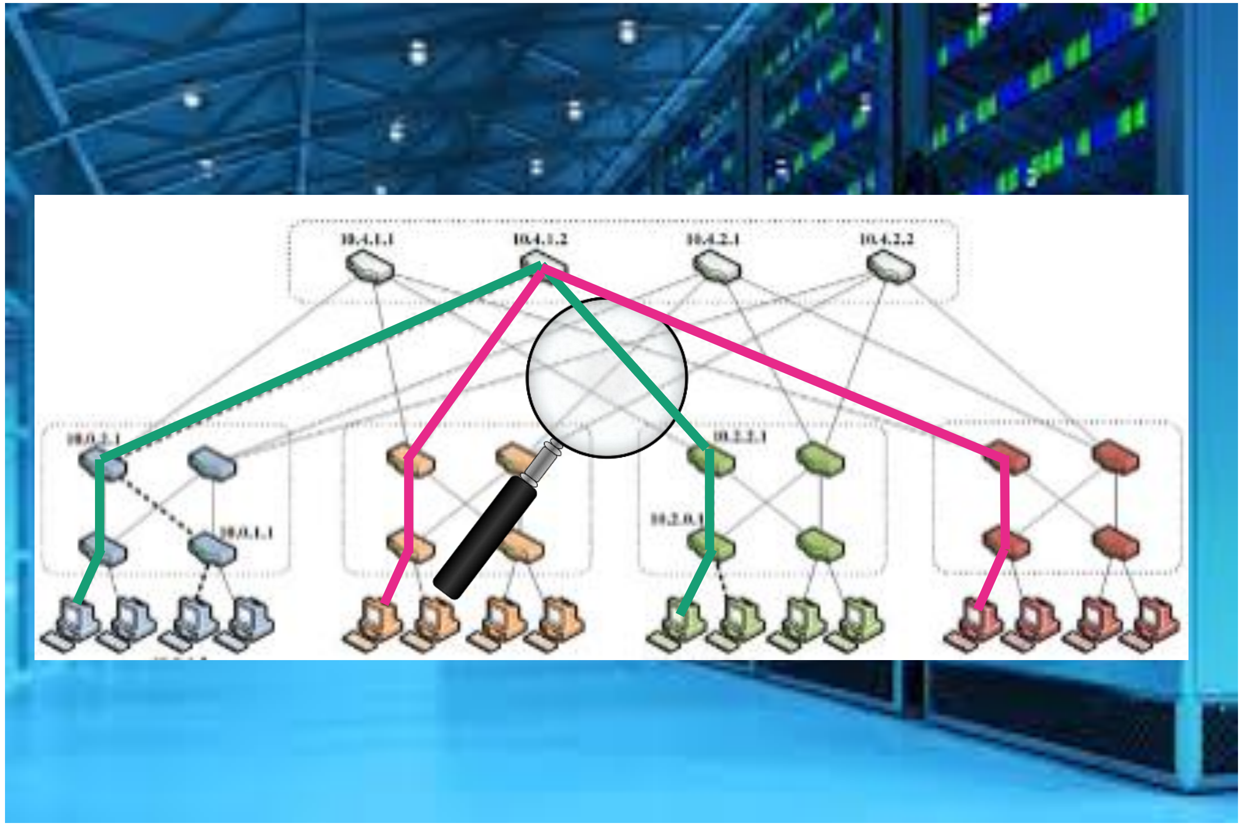
Use Cases of this talk: data center network benchmarking



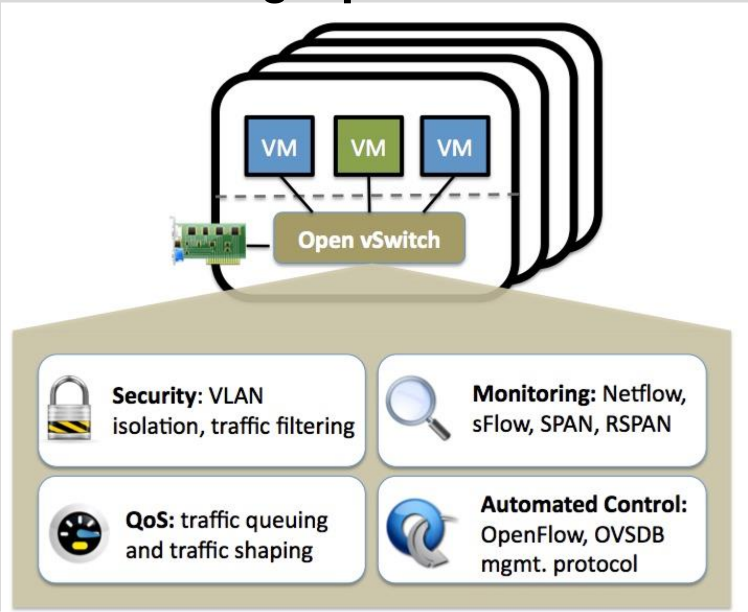
(1) Benchmarking Open vSwitch: NetBOA



Use Cases of this talk: data center network benchmarking



(1) Benchmarking Open vSwitch: NetBOA



(2) Benchmarking Data Center Traffic Scheduling Algorithms: TOXIN

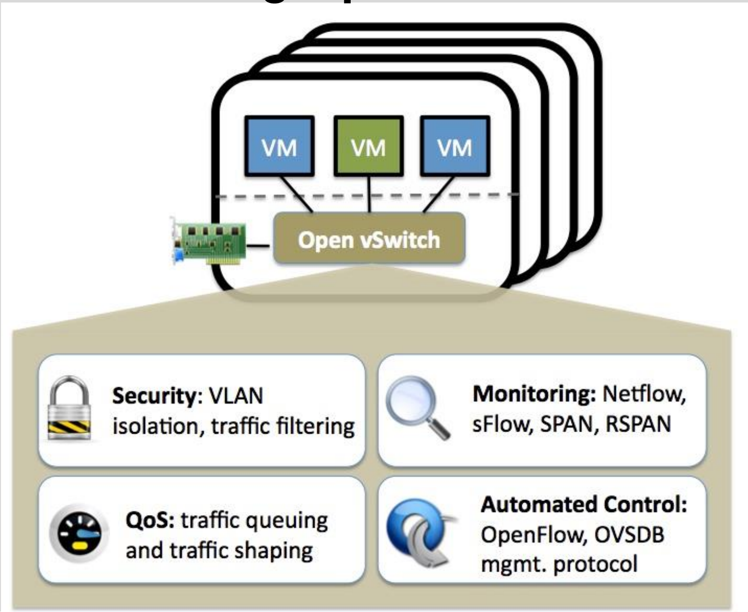
Network flows arrive over time

⇩

Routing algorithm takes decision over time

Use Cases of this talk: data center network benchmarking

(1) Benchmarking Open vSwitch: NetBOA



(2) Benchmarking Data Center Traffic Scheduling Algorithms: TOXIN

Network flows arrive over time

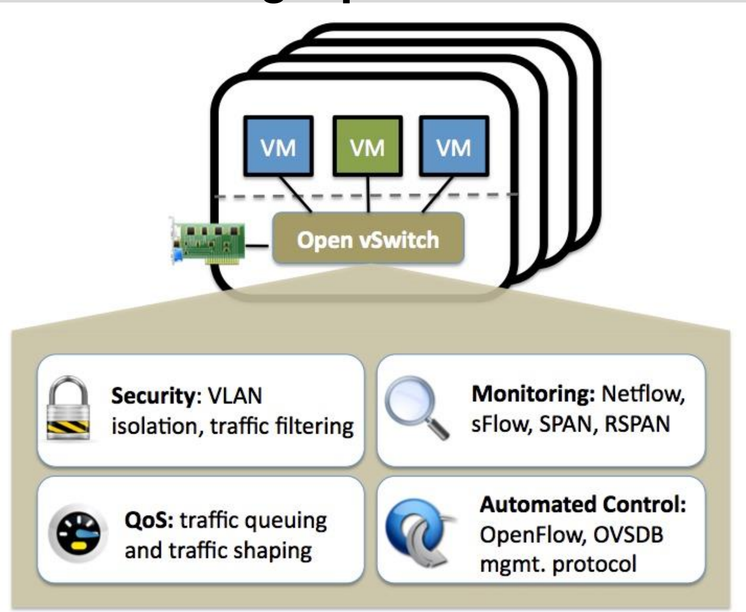
⇩

Routing algorithm takes decision over time

Use Cases of this talk: data center network benchmarking

Milliseconds matter!!!

(1) Benchmarking Open vSwitch: NetBOA



(2) Benchmarking Data Center Traffic Scheduling Algorithms: TOXIN

Network flows arrive over time

↓

Routing algorithm takes decision over time

Problem Scope: Flow Completion Time in Data Centers

The input: set of flows (a population)

	F1	F2	F3	F4	F5	F6
Arrival Time	12ms	14ms	17ms	18ms	21ms	24ms
Source	3	4	13	2	3	12
Destination	14	12	7	7	1	6
Volume	10Mbit	400Mbit	90Mbit	200MBit	9Mbit	110Mbit

Problem Scope: Flow Completion Time in Data Centers

The input: set of flows (a population)

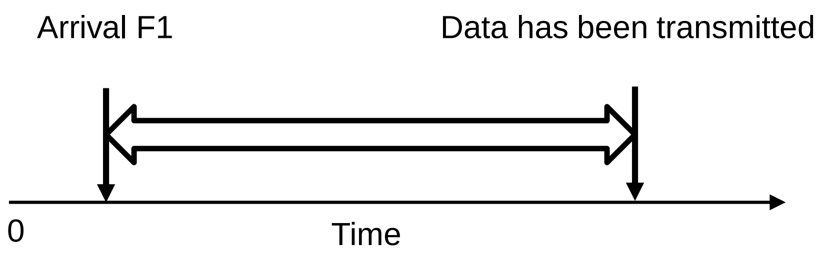
	F1	F2	F3	F4	F5	F6
Arrival Time	12ms	14ms	17ms	18ms	21ms	24ms
Source	3	4	13	2	3	12
Destination	14	12	7	7	1	6
Volume	10Mbit	400Mbit	90Mbit	200MBit	9Mbit	110Mbit

Problem Scope: Flow Completion Time in Data Centers

The input: set of flows (a population)

	F1	F2	F3	F4	F5	F6
Arrival Time	12ms	14ms	17ms	18ms	21ms	24ms
Source	3	4	13	2	3	12
Destination	14	12	7	7	1	6
Volume	10Mbit	400Mbit	90Mbit	200MBit	9Mbit	110Mbit

FCT: Flow Completion Time

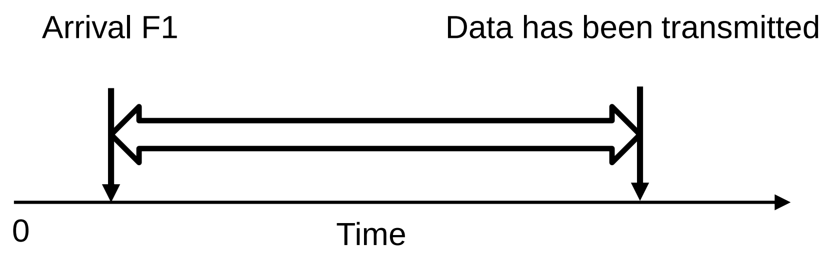


Problem Scope: Flow Completion Time in Data Centers

The input: set of flows (a population)

	F1	F2	F3	F4	F5	F6
Arrival Time	12ms	14ms	17ms	18ms	21ms	24ms
Source	3	4	13	2	3	12
Destination	14	12	7	7	1	6
Volume	10Mbit	400Mbit	90Mbit	200MBit	9Mbit	110Mbit

FCT: Flow Completion Time



Find the order of volumes such that:

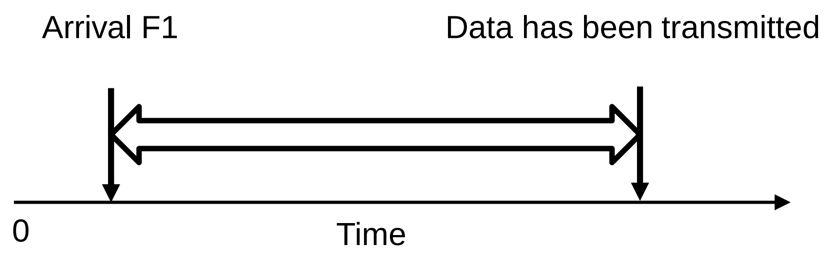
$$\underset{F_N}{\operatorname{argmax}} \frac{1}{N} \sum_{i=1}^{N} FCT(f_i)$$

Problem Scope: Flow Completion Time in Data Centers

The input: set of flows (a population)

	F1	F2	F3	F4	F5	F6
Arrival Time	12ms	14ms	17ms	18ms	21ms	24ms
Source	3	4	13	2	3	12
Destination	14	12	7	7	1	6
Volume	10Mbit	400Mbit	90Mbit	200MBit	9Mbit	110Mbit

FCT: Flow Completion Time



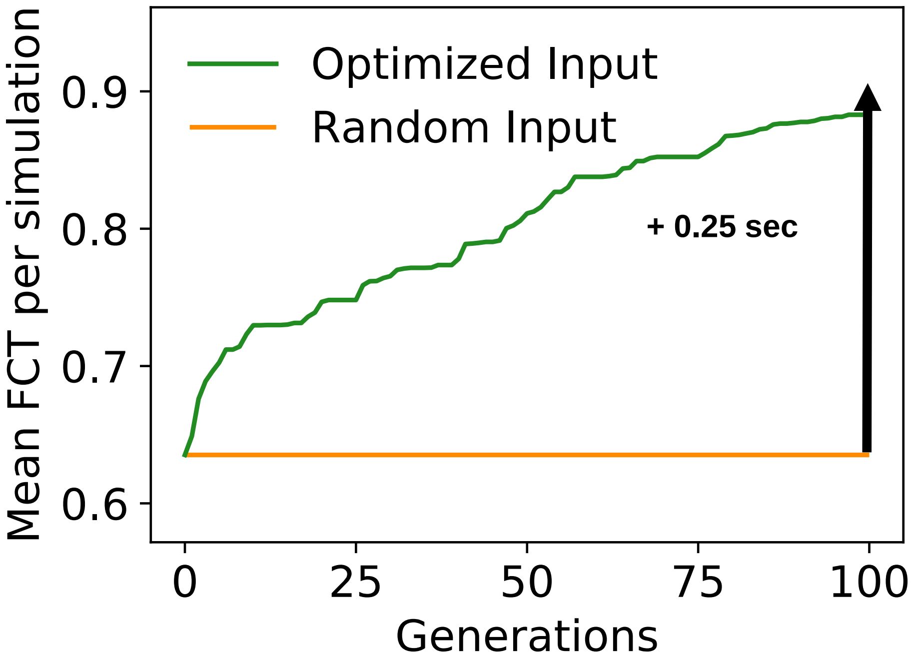
Find the order of volumes such that:

$$\underset{F_N}{\operatorname{argmax}} \frac{1}{N} \sum_{i=1}^{N} FCT(f_i)$$

TOXIN: Use genetic algorithms to find challenging flow input

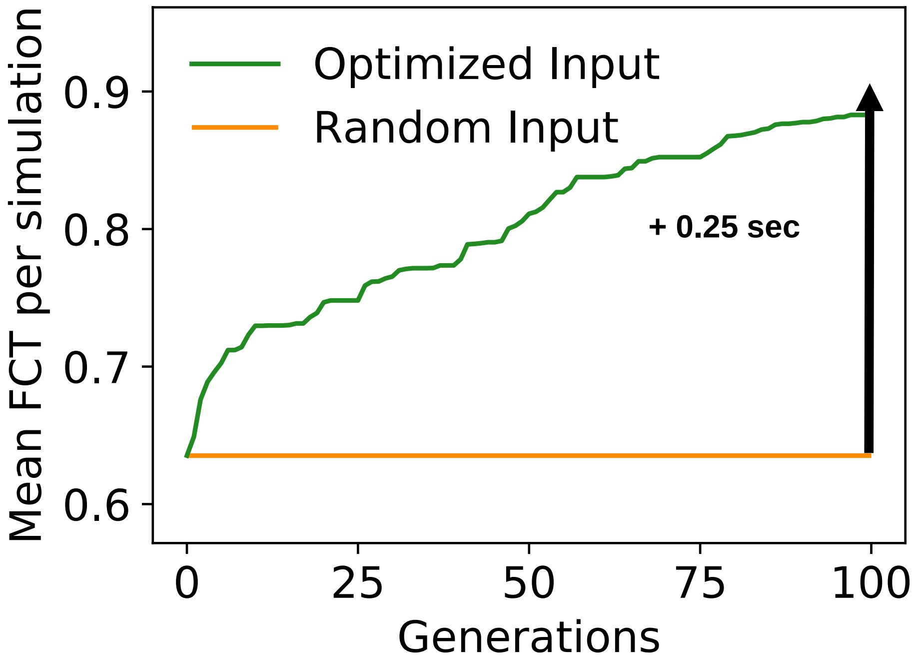
Flow-based Simulation Results of TOXIN for Data Center Networks

Population with N = 30 flows



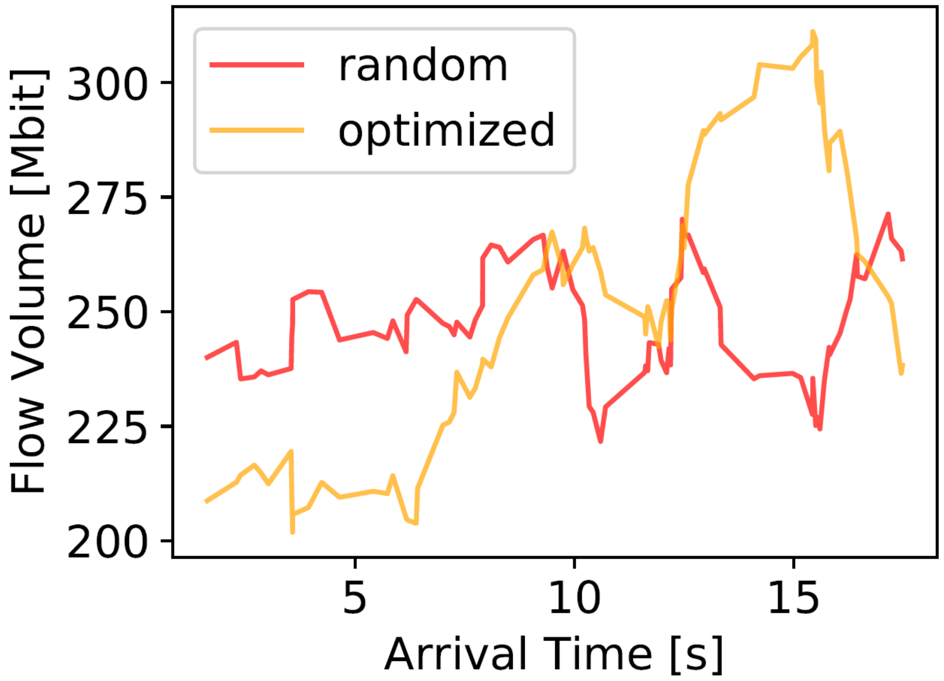
Flow-based Simulation Results of TOXIN for Data Center Networks

Population with N = 30 flows

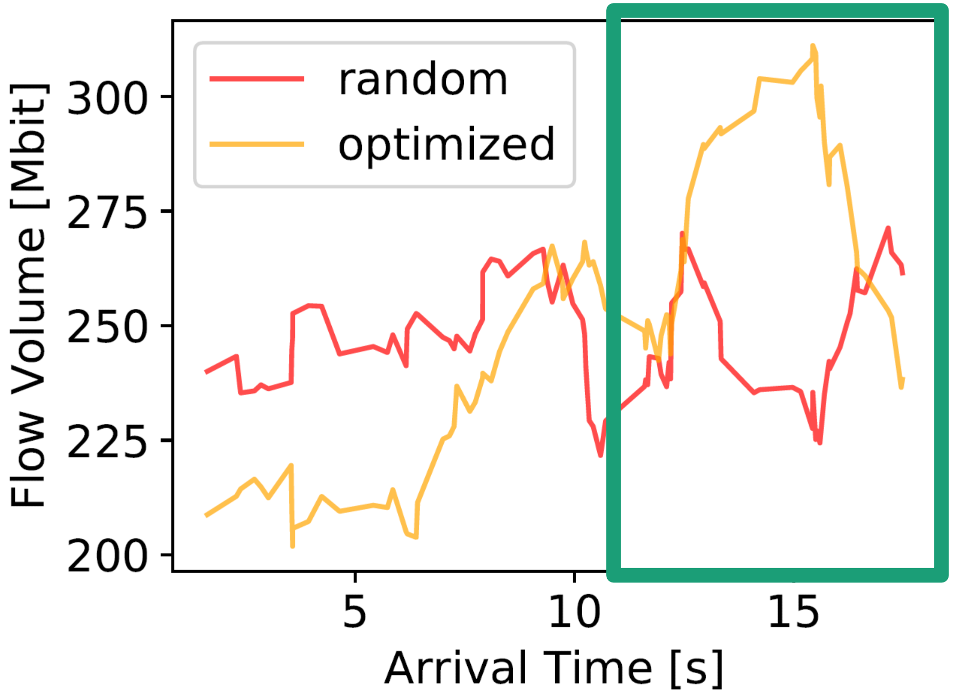


TOXIN creates more challenging input requests than random generation

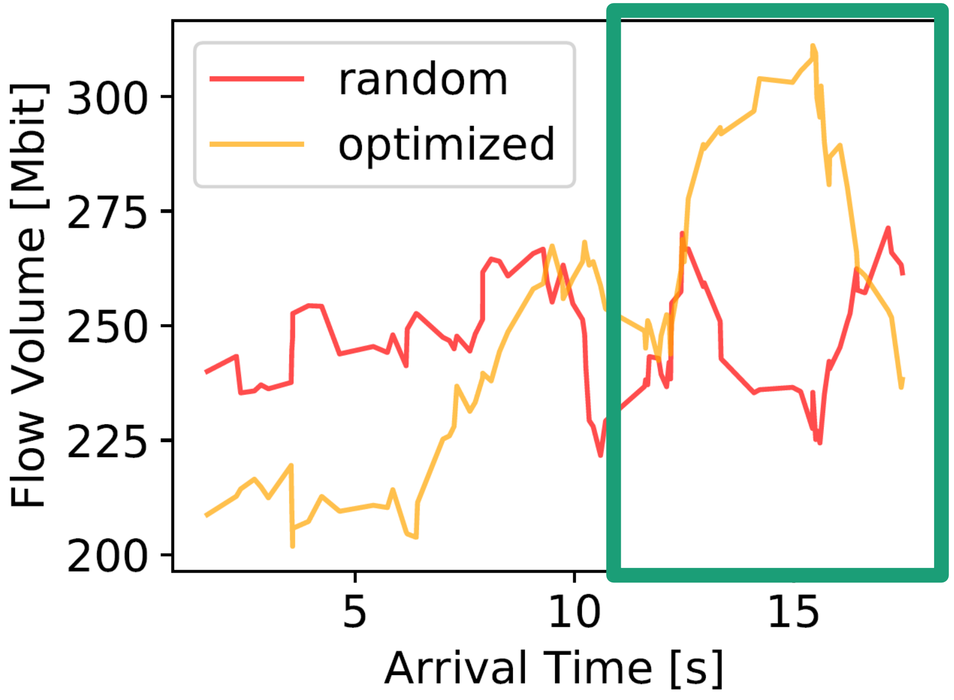
Flow Volume Over Time and Network Connections Over Time

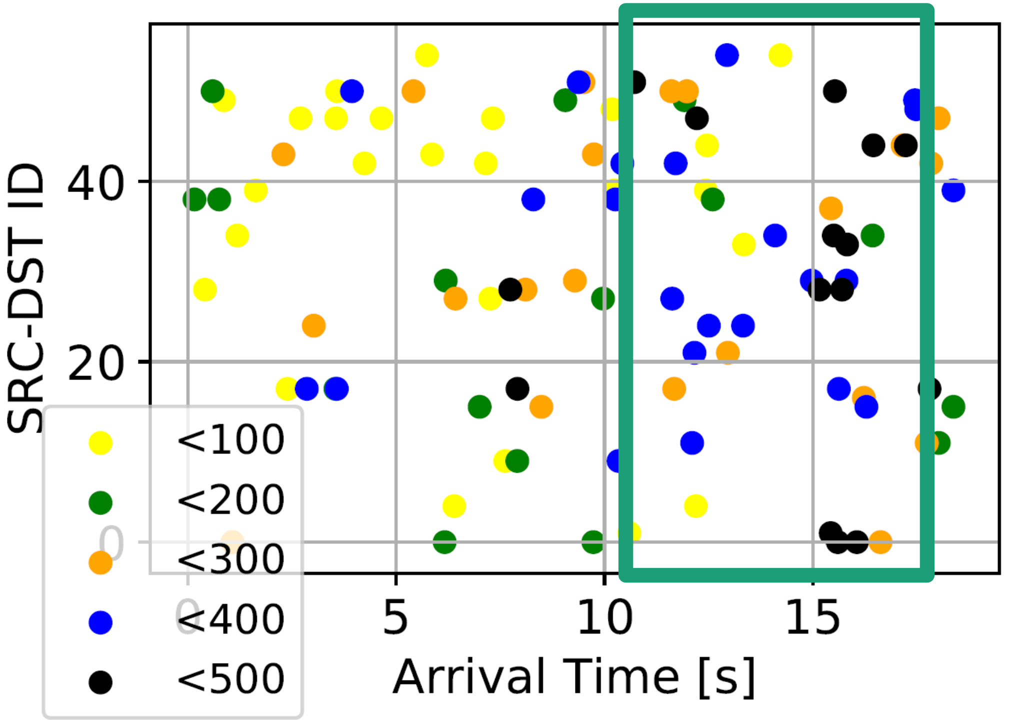


Flow Volume Over Time and Network Connections Over Time

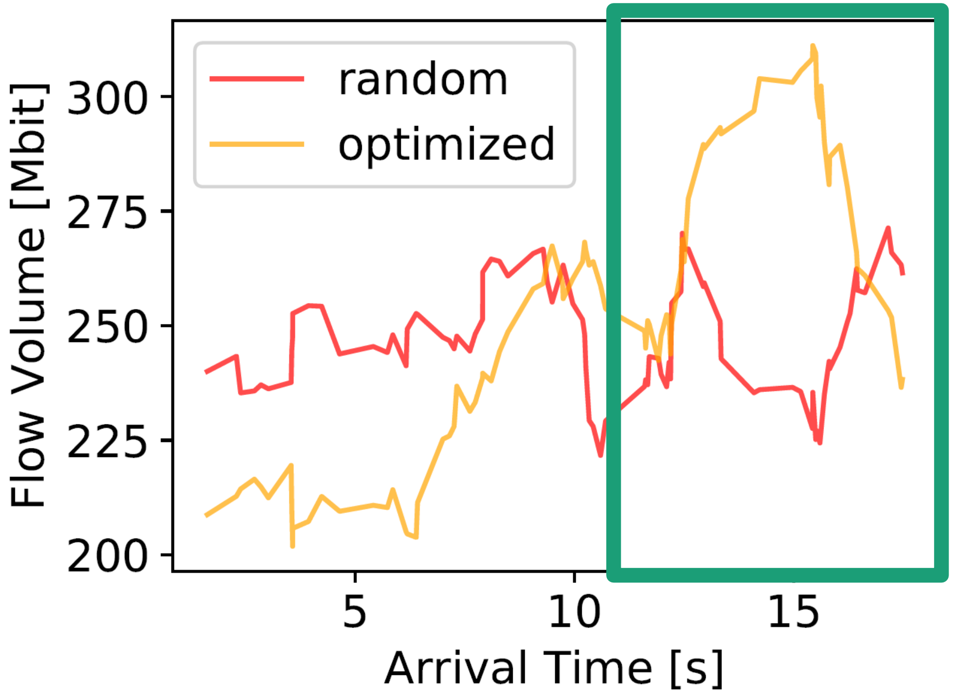


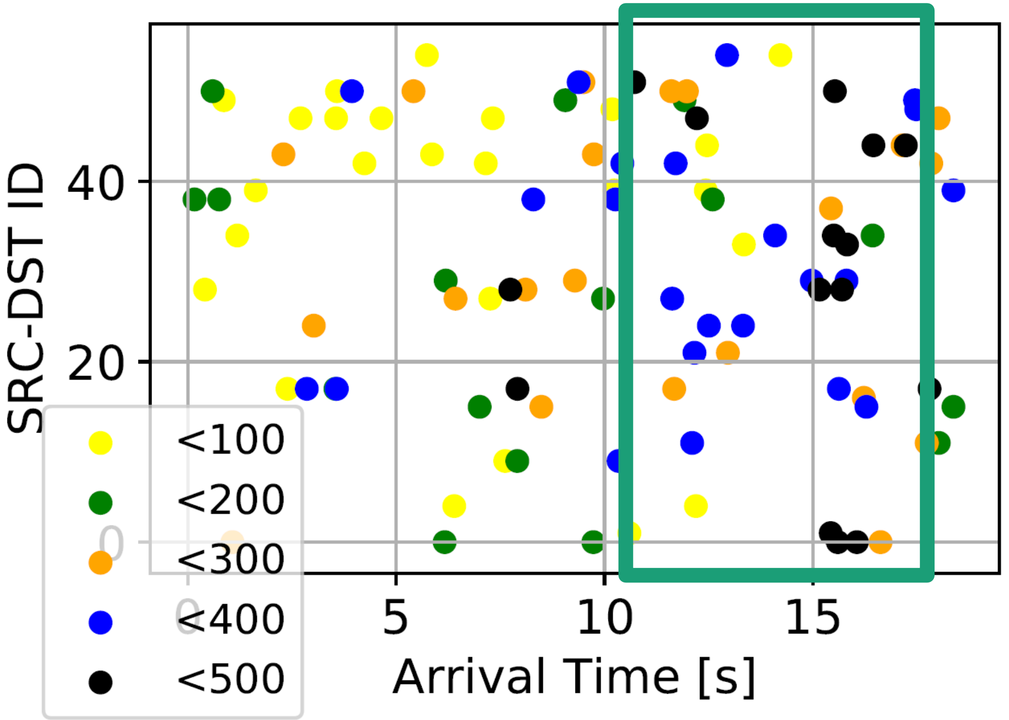
Flow Volume Over Time and Network Connections Over Time





Flow Volume Over Time and Network Connections Over Time





Larger flows go together on the same links! Makes sense …

Summary

Adversarial input can harm your systems!

This talk: Data-Driven approach to automatically generate adversarial input to find weak spots, security holes … to make your systems bullet-proof!

Information missing in this talk: measurement details, simulation details, details on the used machine learning and artificial intelligence algorithms, … anything else :D?

Use concepts like NetBOA and TOXIN to receive continuous feedback about your solutions/implementations

References

[BIG DAMA’17] Blenk, Andreas; Kalmbach, Patrick; Schmid, Stefan; Kellerer, Wolfgang: o'zapft is: Tap Your Network Algorithm's Big Data! ACM SIGCOMM 2017 Workshop on Big Data Analytics and Machine Learning for Data Communication Networks (Big-DAMA), 2017

[SelfDN’18] Kalmbach, Patrick; Zerwas, Johannes; Babarczi, Péter; Blenk, Andreas; Kellerer, Wolfgang; Schmid, Stefan: Empowering Self-Driving Networks. Proceedings of the Afternoon Workshop on Self-Driving Networks - SelfDN 2018, ACM Press, 2018

[NetAI’19] Zerwas, Johannes; Kalmbach, Patrick; Henkel, Laurenz; Retvari, Gabor; Kellerer, Wolfgang; Blenk, Andreas; Schmid, Stefan: NetBOA: Self-Driving Network Benchmarking. ACM SIGCOMM 2019 Workshop on Network Meets AI & ML (NetAI '19), 2019

[CoNEXT‘19] Lettner, Sebastian; Blenk, Andreas: Adversarial Network Algorithm Benchmarking. The 15th International Conference on emerging Networking EXperiments and Technologies (CoNEXT ’19 Companion), ACM, 2019

[TNSM‘19] Zerwas, Johannes; Kalmbach, Patrick; Schmid, Stefan; Blenk, Andreas: Ismael: Using Machine Learning To Predict Acceptance of Virtual Clusters in Data Centers. IEEE Transactions on Network and Service Management, 2019

Thank you!

Questions?