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

Adversarial Network Benchmarking

Andreas Blenk*

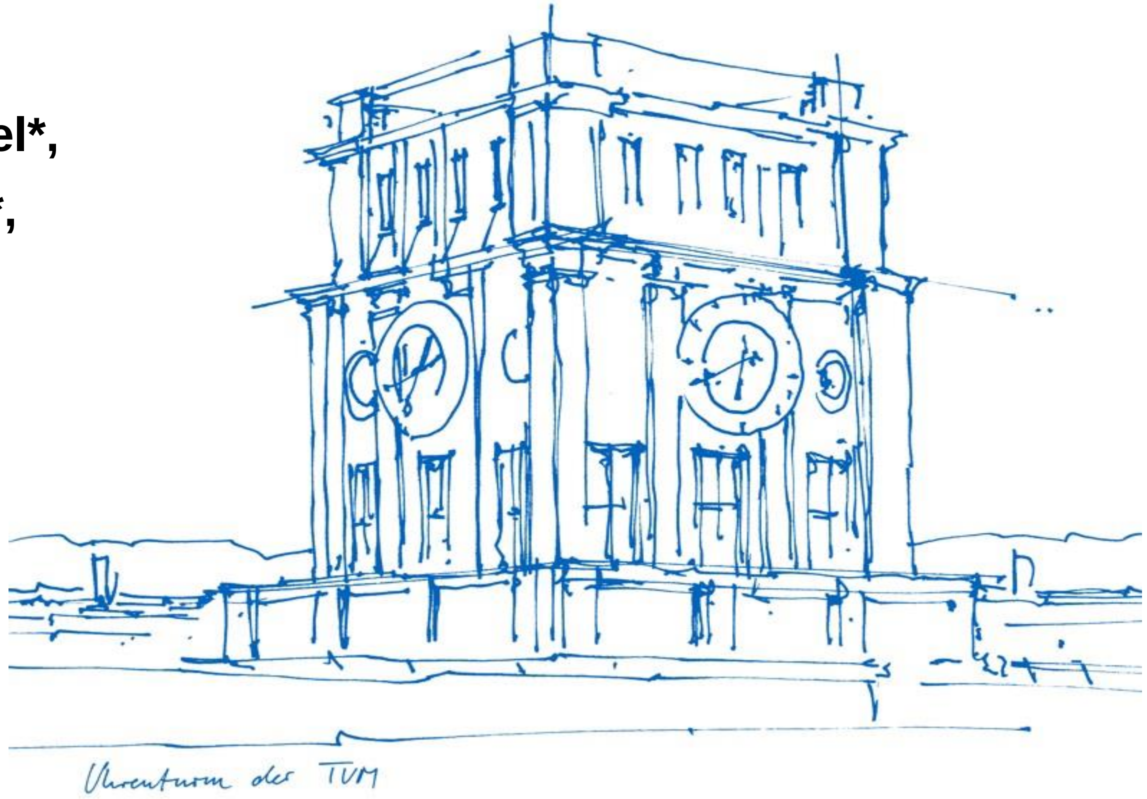
Joint work with:

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



Today’s Approach of Operating Networks?

Monitors → Network Problem

Optimizes → Solution

Designs → Performance Evaluation

With more complex networks need for automation!

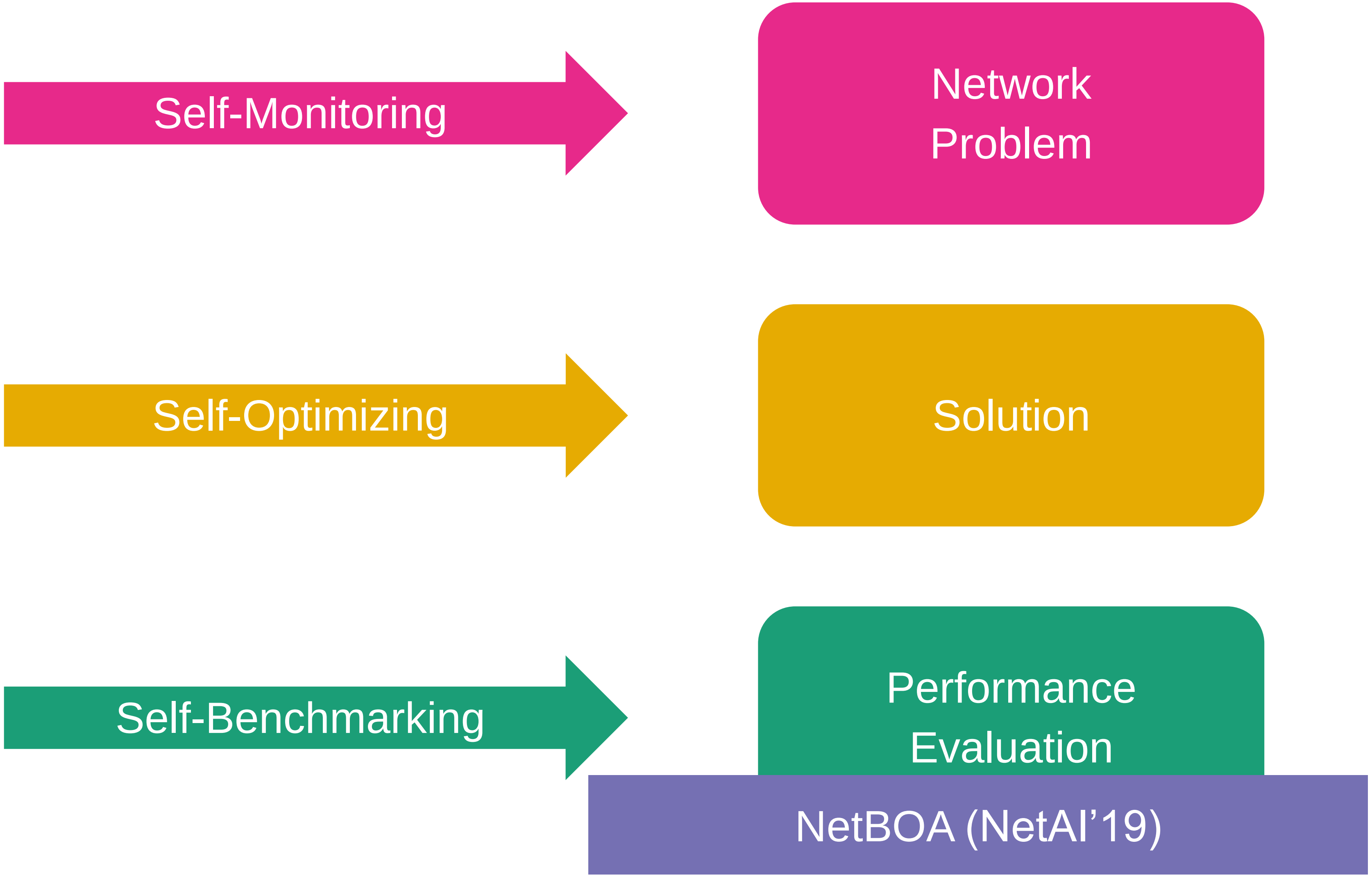
What Self-Driving Networks Should Do

What Self-Driving Networks Should Do

Source: https://www.pinterest.at/pin/318137161149129652/

What Self-Driving Networks Should Do

Source: https://www.pinterest.at/pin/318137161149129652/



Self-Monitoring → Network Problem

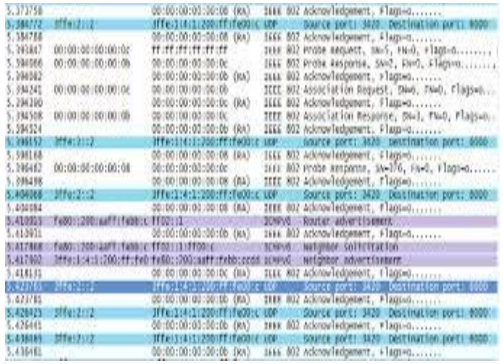
Self-Optimizing → Solution

Self-Benchmarking → Performance Evaluation

NetBOA (NetAI’19)

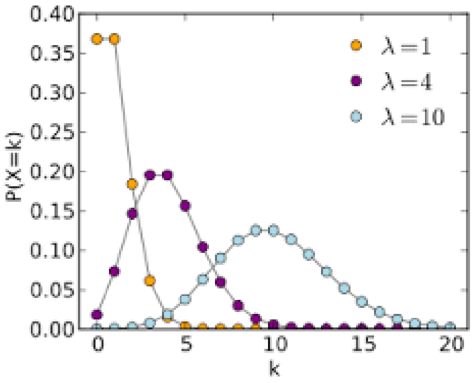
Benchmarking Network Algorithms, Architectures etc…

The Traditional Way …



```
5.373758                         00:00:00:00:00:0b (RA)   IEEE 802 Acknowledgement, Flags=........
5.384772  3ffe:2::2              3ffe:1:4:1:200:ff:fe00:c UDP      Source port: 3420  Destination port: 8000
5.384788                         00:00:00:00:00:0b (RA)   IEEE 802 Acknowledgement, Flags=........
5.391847  00:00:00:00:00:0c      ff:ff:ff:ff:ff:ff        IEEE 802 Probe Request, SN=5, FN=0, Flags=........,
5.394066  00:00:00:00:00:0b      00:00:00:00:00:0c        IEEE 802 Probe Response, SN=2, FN=0, Flags=........,
5.394082                         00:00:00:00:00:0b (RA)   IEEE 802 Acknowledgement, Flags=........
5.394241  00:00:00:00:00:0c      00:00:00:00:00:0b        IEEE 802 Association Request, SN=6, FN=0, Flags=..
5.394260                         00:00:00:00:00:0c (RA)   IEEE 802 Acknowledgement, Flags=........
5.394508  00:00:00:00:00:0b      00:00:00:00:00:0c        IEEE 802 Association Response, SN=3, FN=0, Flags=..
5.394524                         00:00:00:00:00:0b (RA)   IEEE 802 Acknowledgement, Flags=........
5.396152  3ffe:2::2              3ffe:1:4:1:200:ff:fe00:c UDP      Source port: 3420  Destination port: 8000
5.396168                         00:00:00:00:00:08 (RA)   IEEE 802 Acknowledgement, Flags=........
5.396482  00:00:00:00:00:08      00:00:00:00:00:0c        IEEE 802 Probe Response, SN=376, FN=0, Flags=........
5.396498                         00:00:00:00:00:08 (RA)   IEEE 802 Acknowledgement, Flags=........
5.404068  3ffe:2::2              3ffe:1:4:1:200:ff:fe00:c UDP      Source port: 3420  Destination port: 8000
5.404084                         00:00:00:00:00:08 (RA)   IEEE 802 Acknowledgement, Flags=........
5.410015  fe80::200:aff:febb:c   ff02::1                  ICMPv6   Router advertisement
5.410031                         00:00:00:00:00:0b (RA)   IEEE 802 Acknowledgement, Flags=........
5.417868  fe80::200:aff:febb:c   ff02::1:ff00:c           ICMPv6   Neighbor solicitation
5.417892  3ffe:1:4:1:200:ff:fe0  fe80::200:aff:febb:cccd  ICMPv6   Neighbor advertisement
5.418131                         00:00:00:00:00:0c (RA)   IEEE 802 Acknowledgement, Flags=........
5.423765  3ffe:2::2              3ffe:1:4:1:200:ff:fe00:c UDP      Source port: 3420  Destination port: 8000
5.423781                         00:00:00:00:00:0b (RA)   IEEE 802 Acknowledgement, Flags=........
5.426425  3ffe:2::2              3ffe:1:4:1:200:ff:fe00:c UDP      Source port: 3420  Destination port: 8000
5.426441                         00:00:00:00:00:0b (RA)   IEEE 802 Acknowledgement, Flags=........
5.436085  3ffe:2::2              3ffe:1:4:1:200:ff:fe00:c UDP      Source port: 3420  Destination port: 8000
5.436101                         00:00:00:00:00:0b (RA)   IEEE 802 Acknowledgement, Flags=........
```

Traces



Models

Human‘s Best Guesses

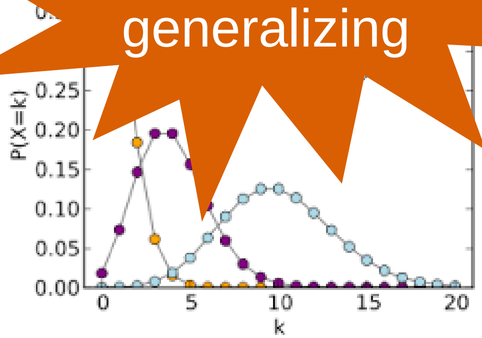
Data-Driven

Benchmarking Network Algorithms, Architectures etc…

The Traditional Way …

Not always available

Traces



Not generalizing

Models

Hmm… Biased?

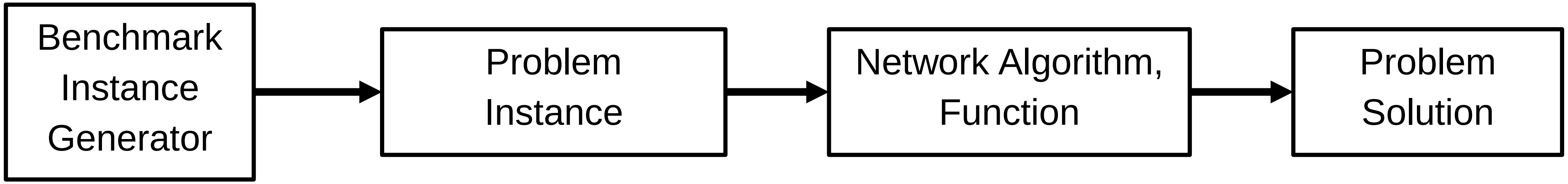
Human‘s Best Guesses

Alternative opponent?

Data-Driven

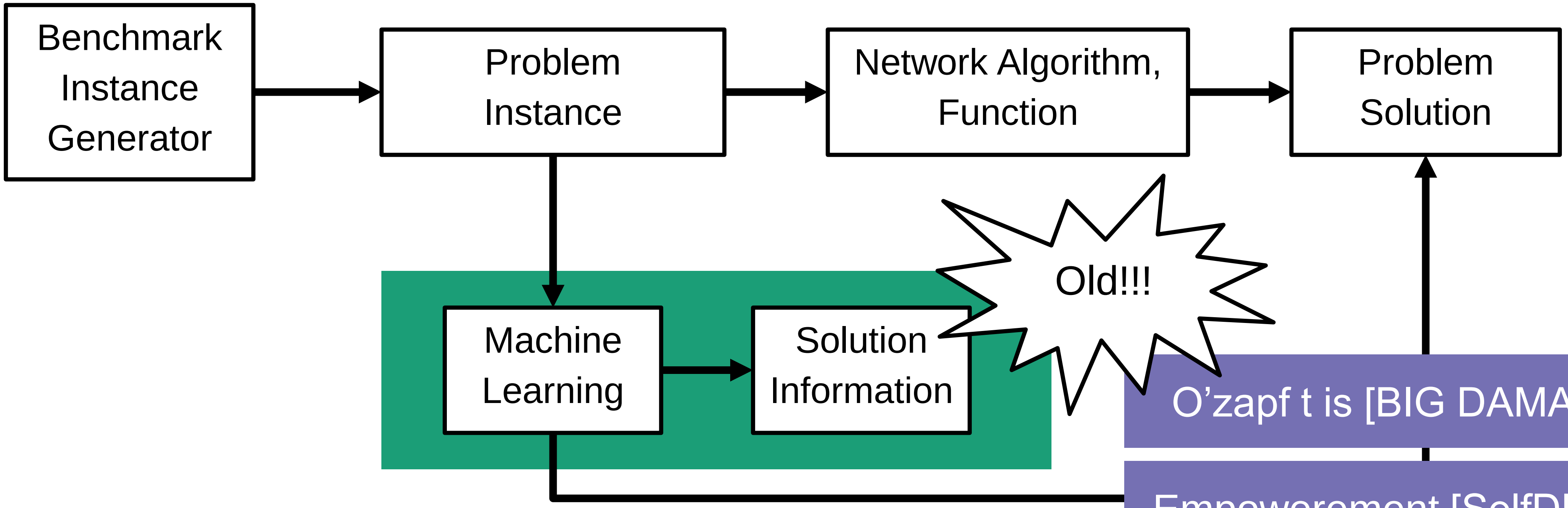
This Talk: Use Machine Learning to Benchmark Networks

Towards Automated Network Optimization and Design



The Traditional Way!

Towards Automated Network Optimization and Design



Benchmark Instance Generator

Problem Instance

Network Algorithm, Function

Problem Solution

Machine Learning

Solution Information

Old!!!

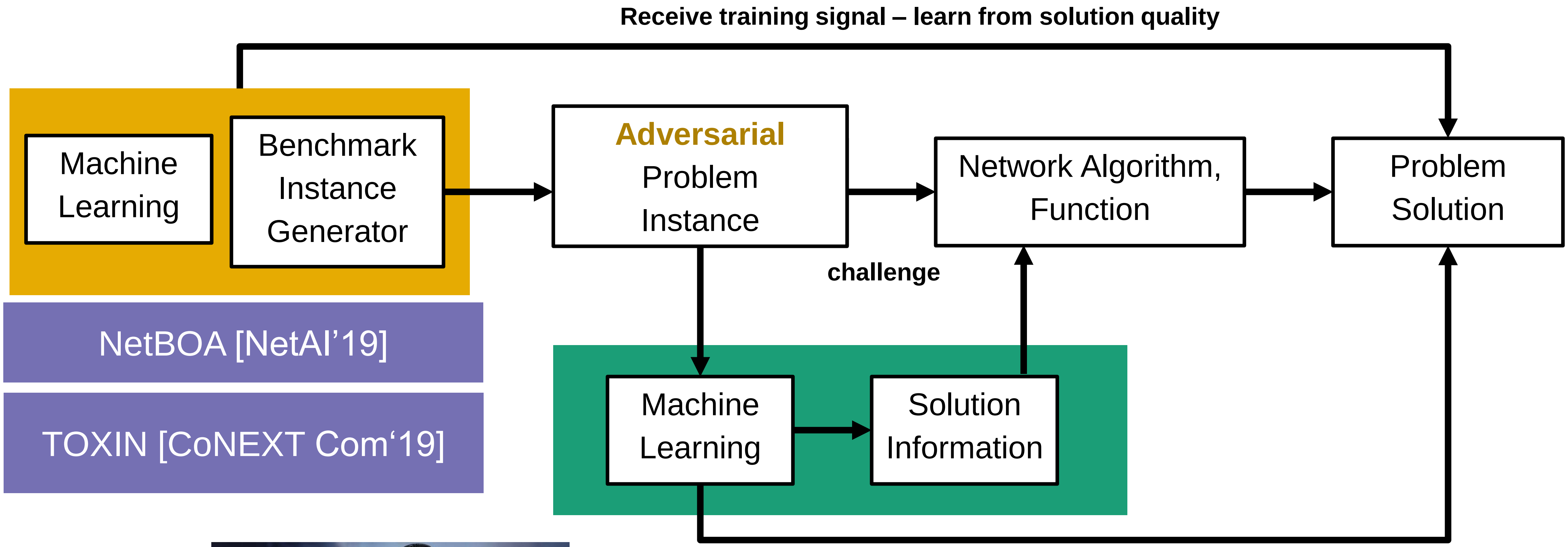
O’zapf t is [BIG DAMA’17]

Empowerement [SelfDN’18]

ISMAEL [TNSM’19]

Our ML/AI Way!

Towards Automated Network Optimization and Design


Receive training signal – learn from solution quality
Machine Learning
Benchmark Instance Generator
Adversarial Problem Instance
Network Algorithm, Function
Problem Solution
challenge
Machine Learning
Solution Information

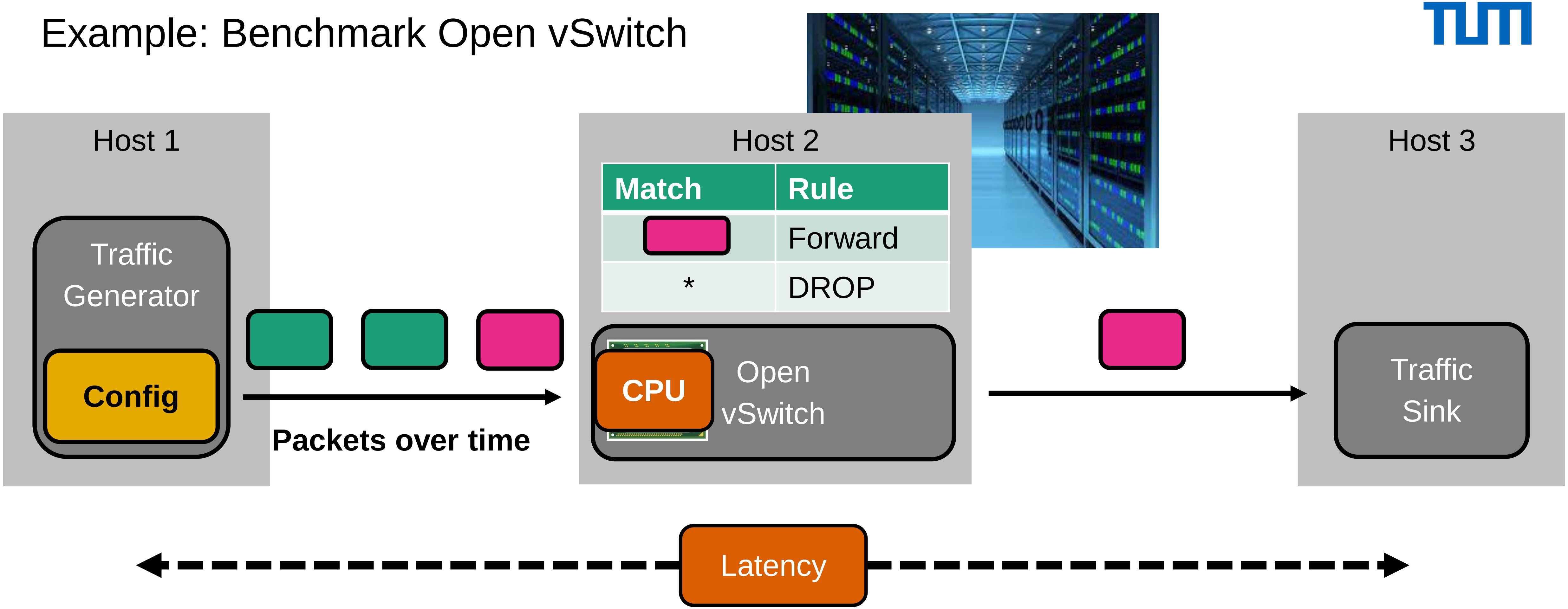
NetBOA [NetAI’19]

TOXIN [CoNEXT Com‘19]

Our ML/AI Way!

ML/AI vs ML/AI

Example: Benchmark Open vSwitch

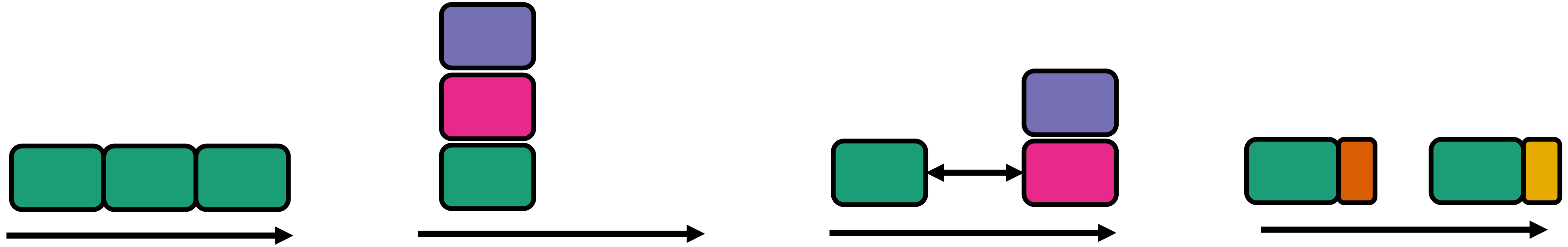


Match	Rule
🟥	Forward
*	DROP

Goal: Find Network Traffic Configuration that Maximizes CPU/Latency

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.



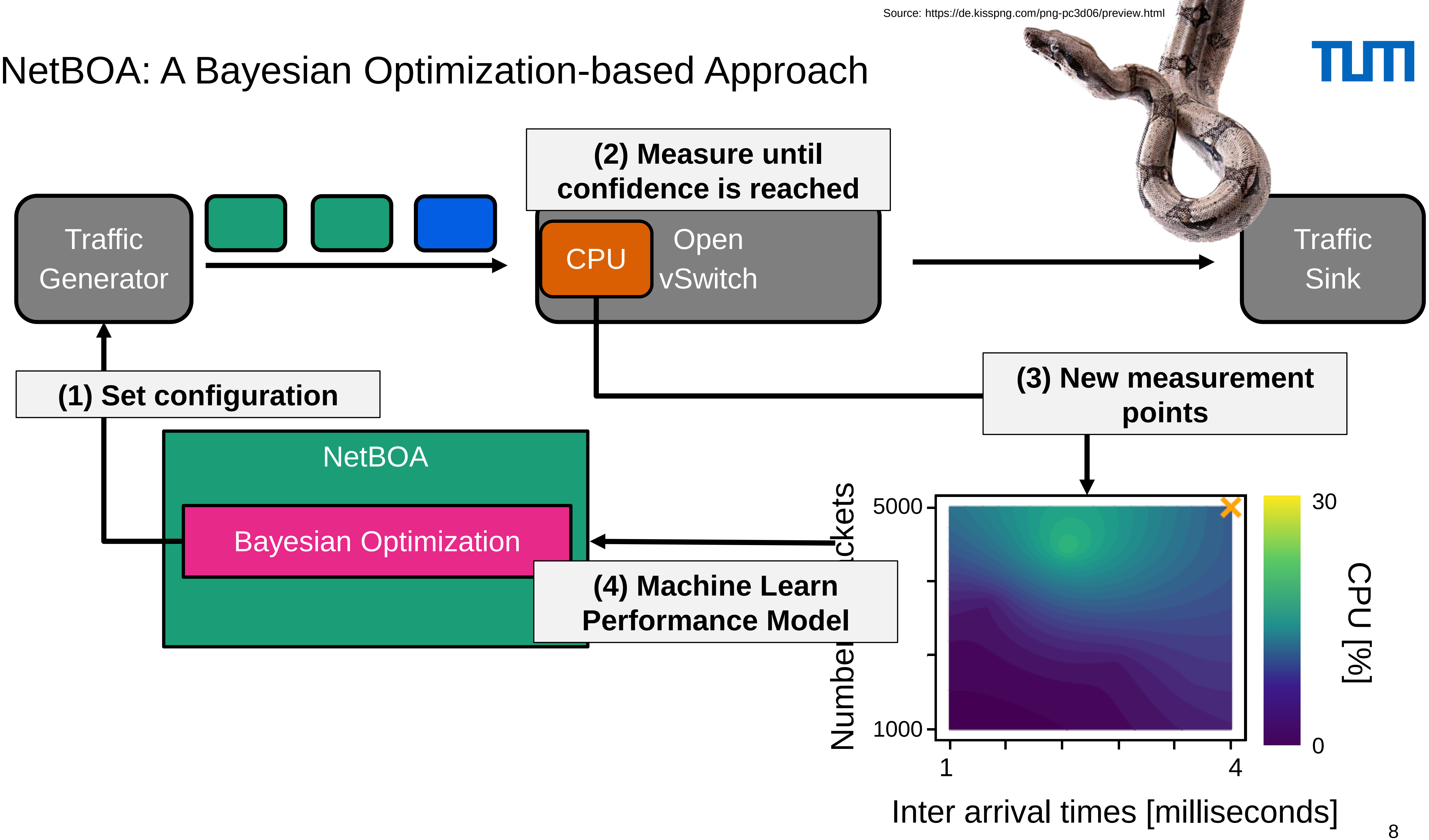
Number of Network Packets [1000 – 5000]

Batch Size [1-5]

Packet Inter Arrival Time [1ms – 13ms]

VLANs [1-5]

NetBOA: A Bayesian Optimization-based Approach



Source: https://de.kisspng.com/png-pc3d06/preview.html

Traffic Generator

Open vSwitch

CPU

Traffic Sink

(1) Set configuration

(2) Measure until confidence is reached

(3) New measurement points

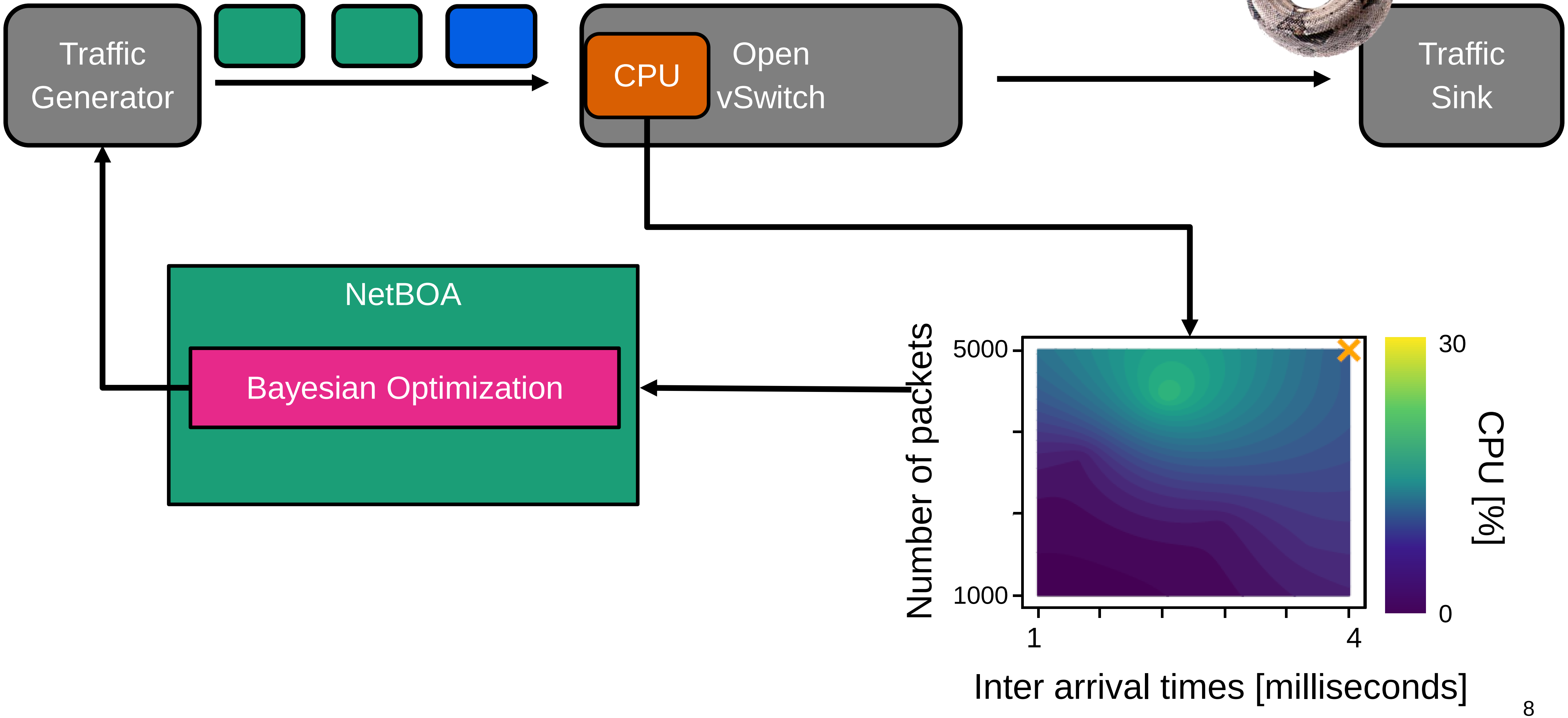
(4) Machine Learn Performance Model

NetBOA

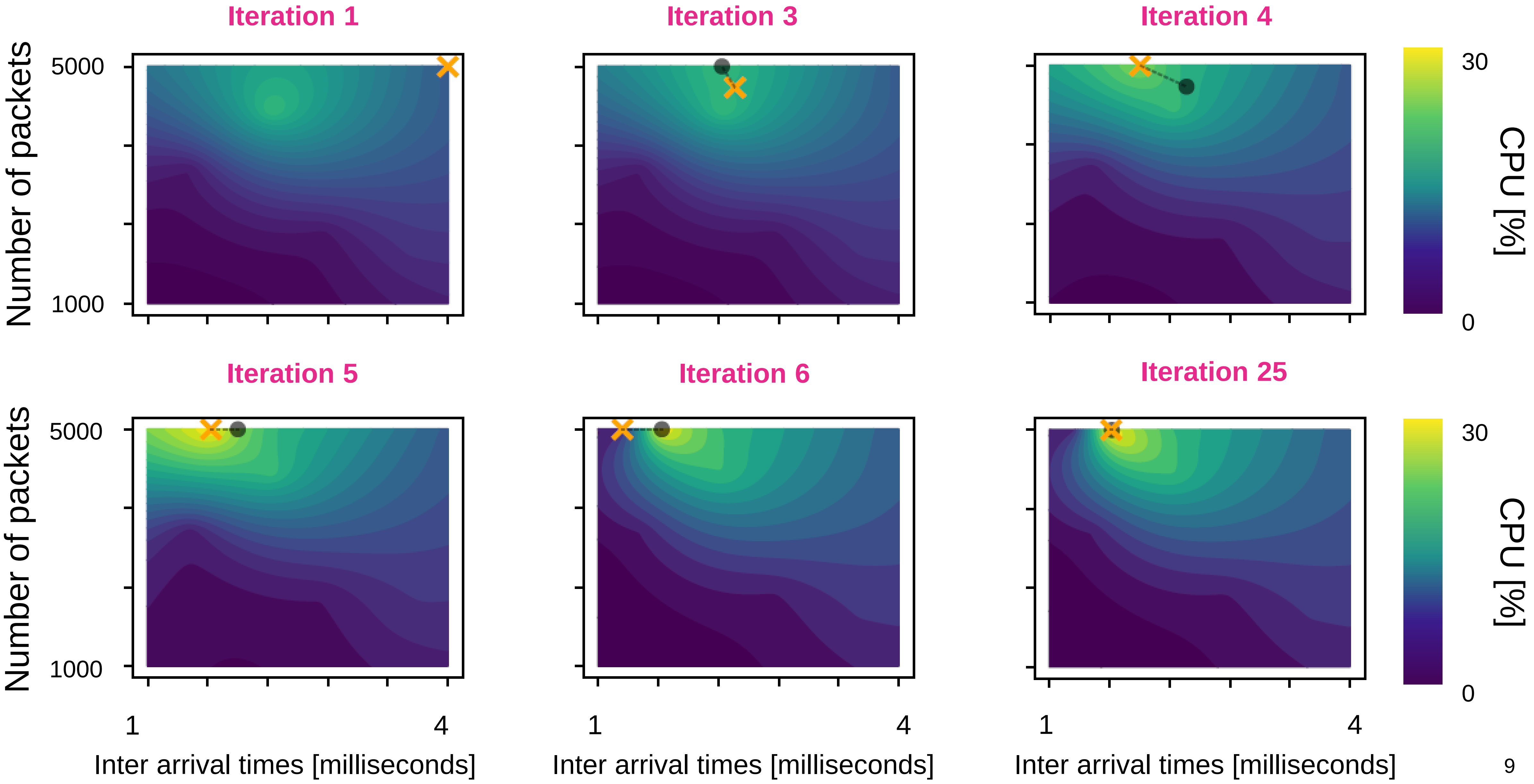
Bayesian Optimization

Source: https://de.kisspng.com/png-pc3d06/preview.html

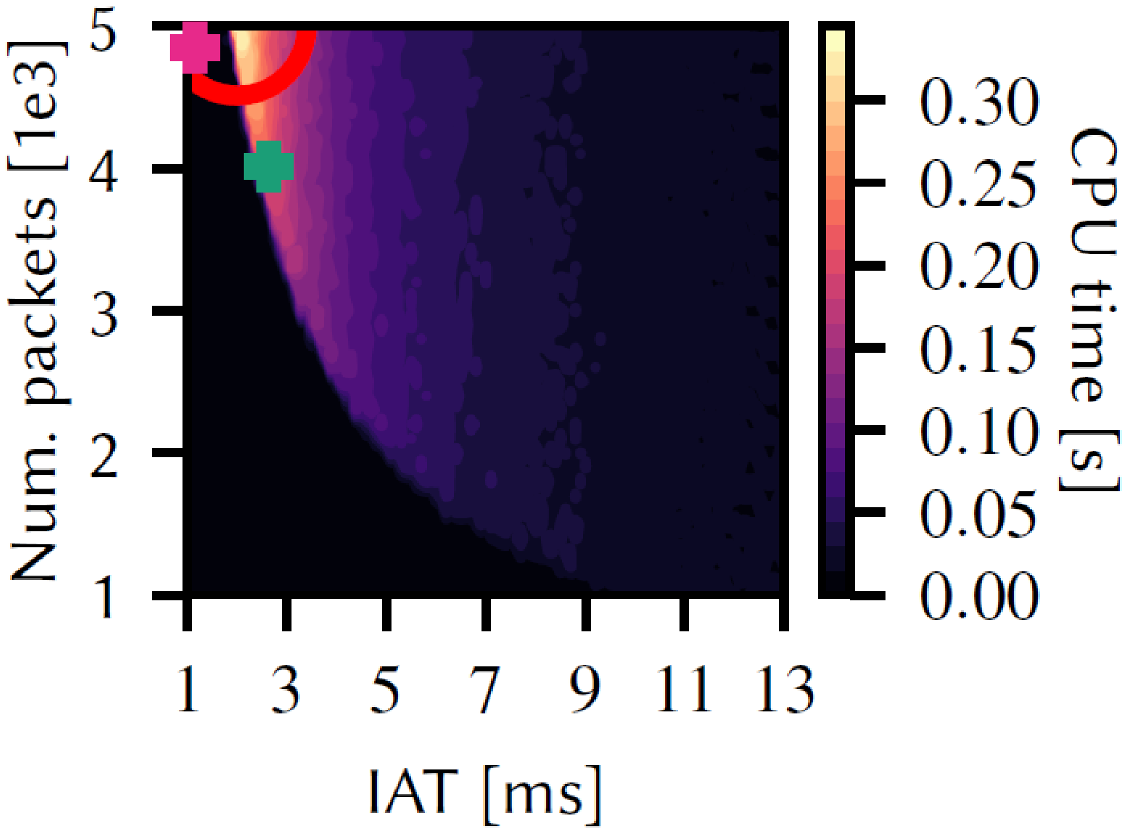
NetBOA: A Bayesian Optimization-based Approach



How NetBOA Explores the Performance Model



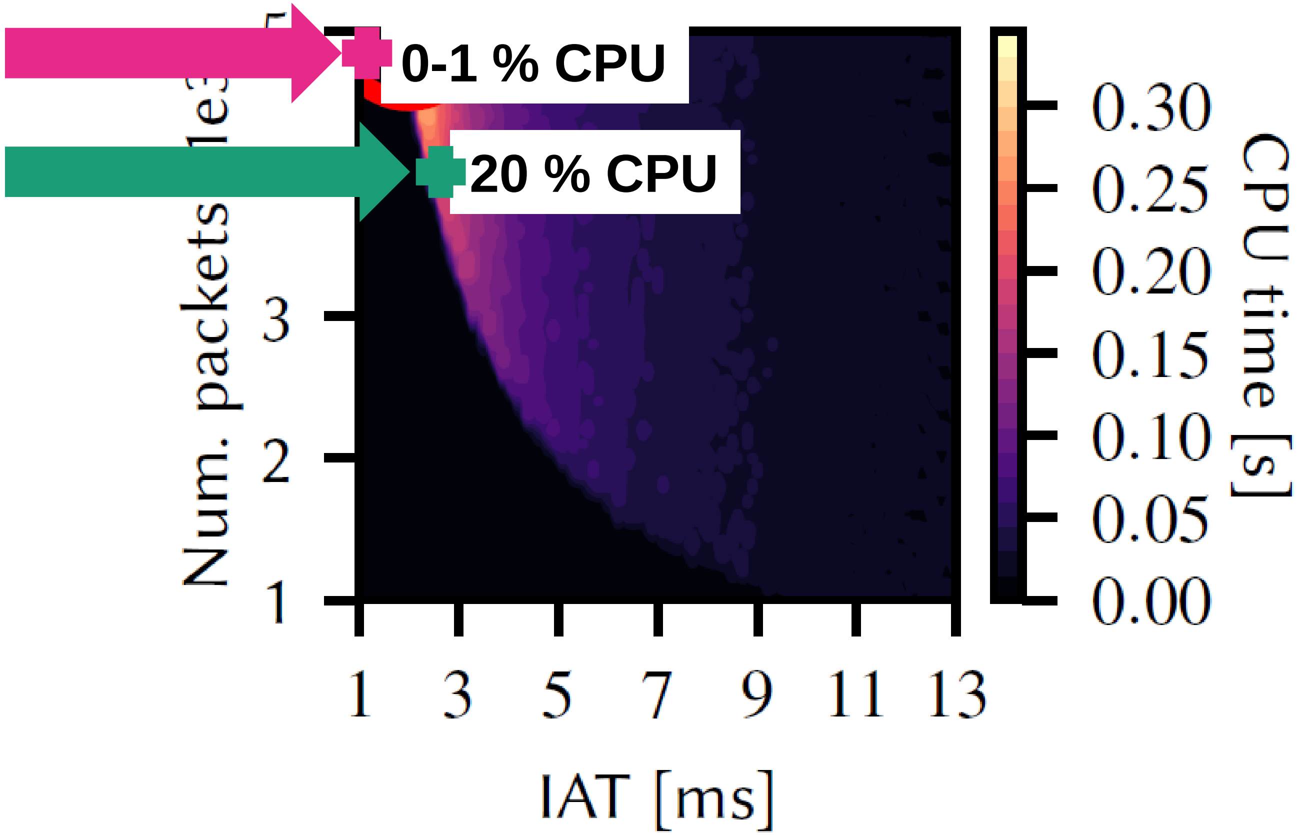
Grid Search for Two Parameters (Num. Packets and Inter Arrival Time)



Grid Search for Two Parameters (Num. Packets and Inter Arrival Time)

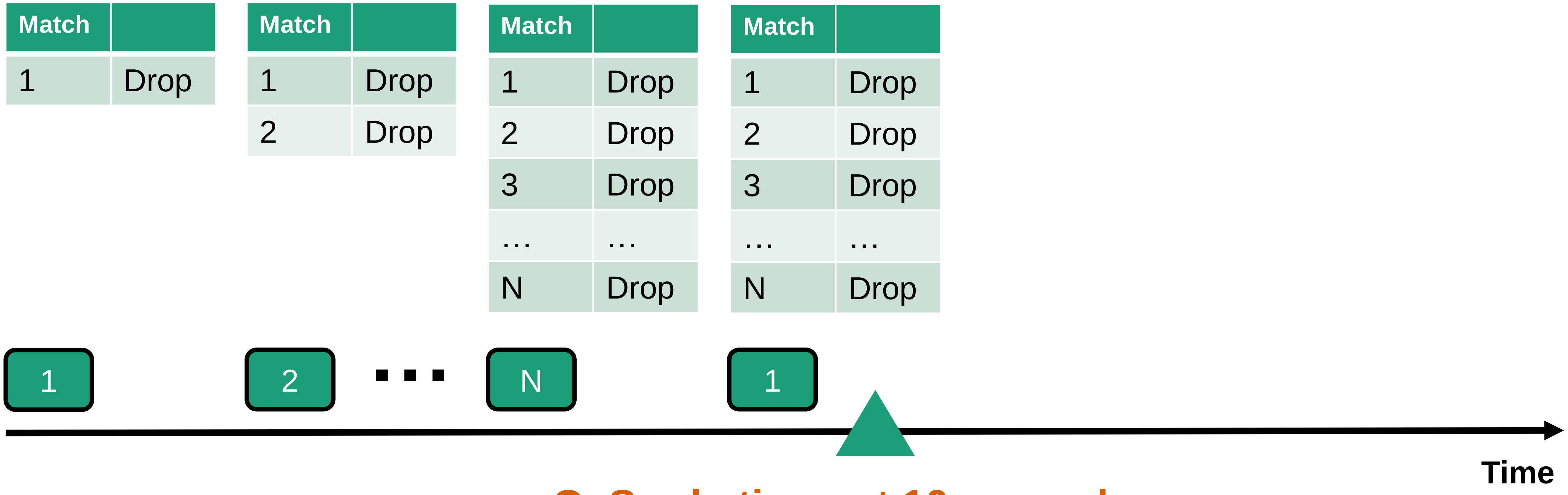
5000 packets,
IAT 1 ms

4000 packets,
IAT 3 ms



- Performance models are non-trivial
- **Surprising:** Sending less network packets over time can lead to significantly higher CPU
- **But**: Can we find such weak-spots automatically?

Why? Let Us Look At OvS Behavior!



Match	
1	Drop

Match	
1	Drop
2	Drop

Match	
1	Drop
2	Drop
3	Drop
…	…
N	Drop

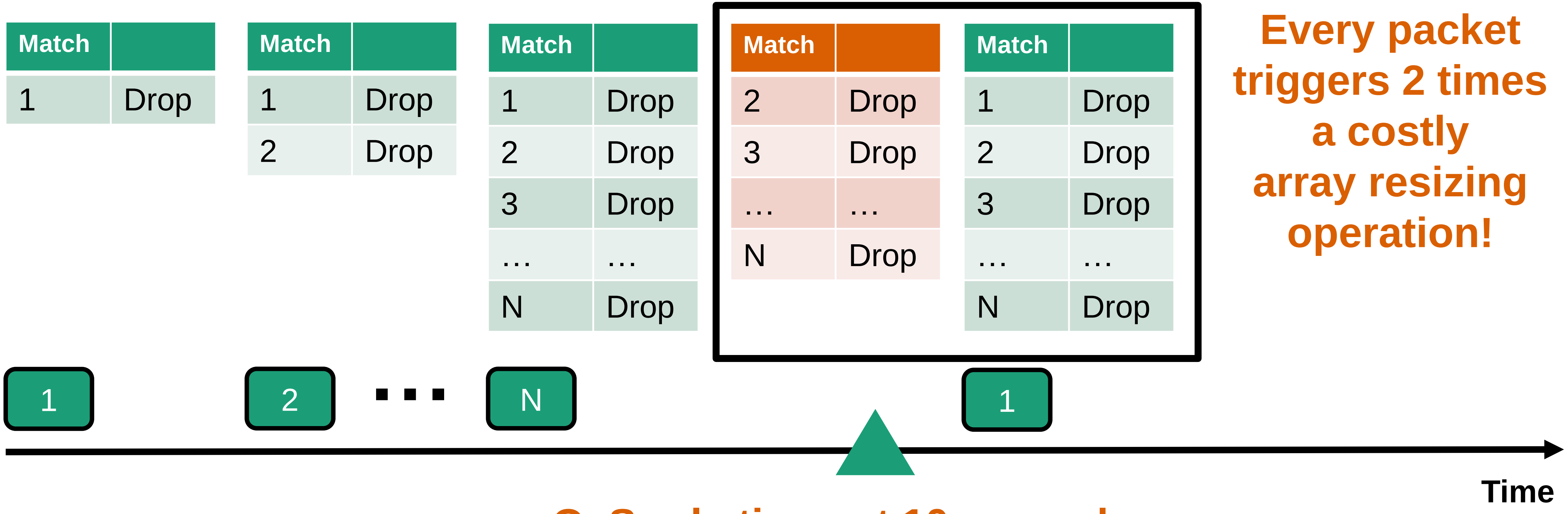
Match	
1	Drop
2	Drop
3	Drop
…	…
N	Drop

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**

Why? Let Us Look At OvS Behavior!



Match	
1	Drop

Match	
1	Drop
2	Drop

Match	
1	Drop
2	Drop
3	Drop
…	…
N	Drop

Match	
2	Drop
3	Drop
…	…
N	Drop

Match	
1	Drop
2	Drop
3	Drop
…	…
N	Drop

Every packet triggers 2 times a costly array resizing operation!

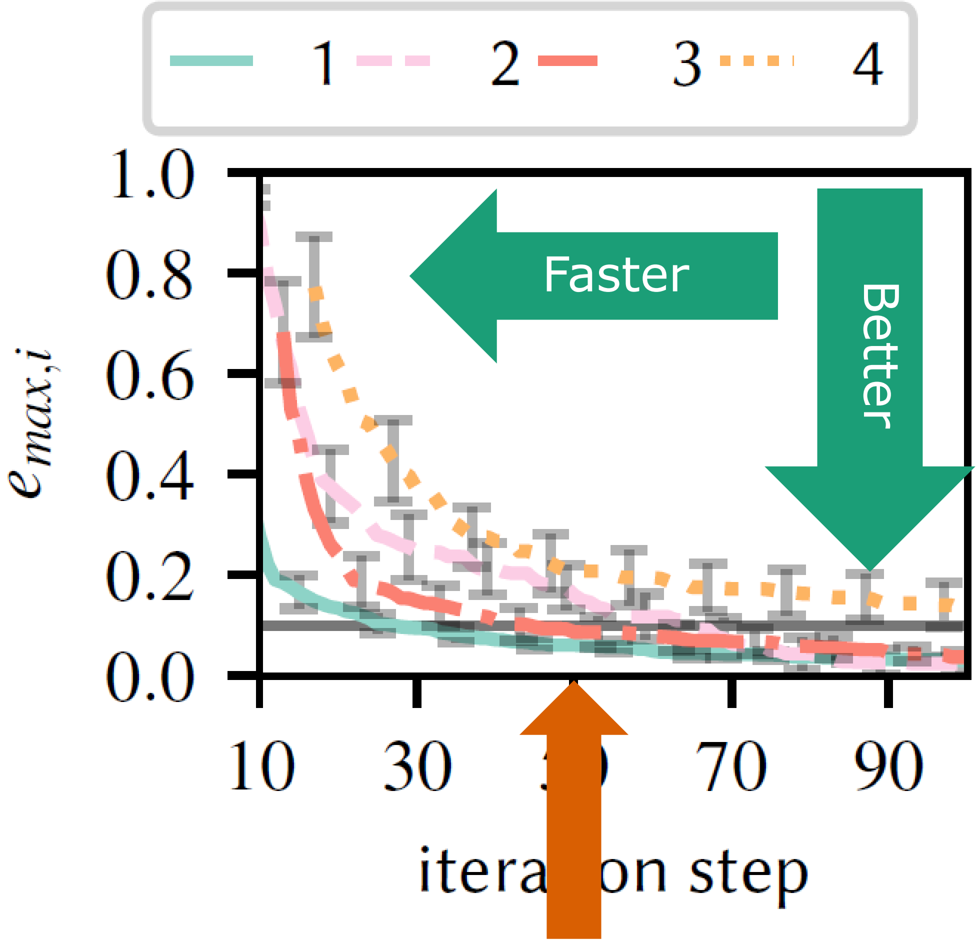
OvS rule timeout 10 seconds

- 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

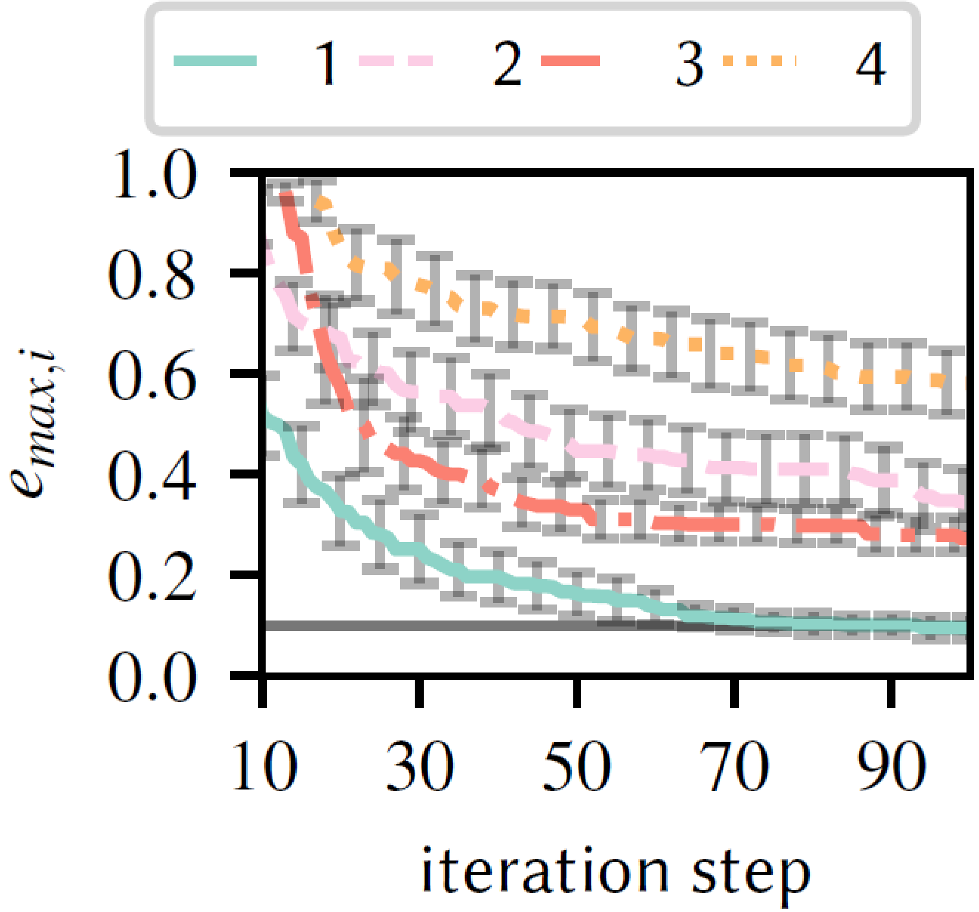
→ **Forcing OvS to continuously run through the array + resizing it**

NetBOA vs Random Search

NetBOA



Random Search



24 % higher CPU utilization

Conclusion

- Adversarial input generation to find weak spots, security holes … to make your systems bullet-proof? → **Use concepts like NetBOA to receive continuous feedback about your solutions/implementations**

- Use case: NetBOA is a Bayesian Optimization-based data-driven approach to generate network traffic configurations for benchmarking network function implementations

→NetBOA can efficiently find challenging network traffic configurations (maximize CPU/Latency)

→NetBOA can also be used to minimize, e.g., CPU or Latency

- Open questions and problems:
  - Does beating the machine means it generalizes?
  - Does it scale?
  - Alternatives?
  - Bayesian Optimization needs also tuning!