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

Adversarial Network Benchmarking

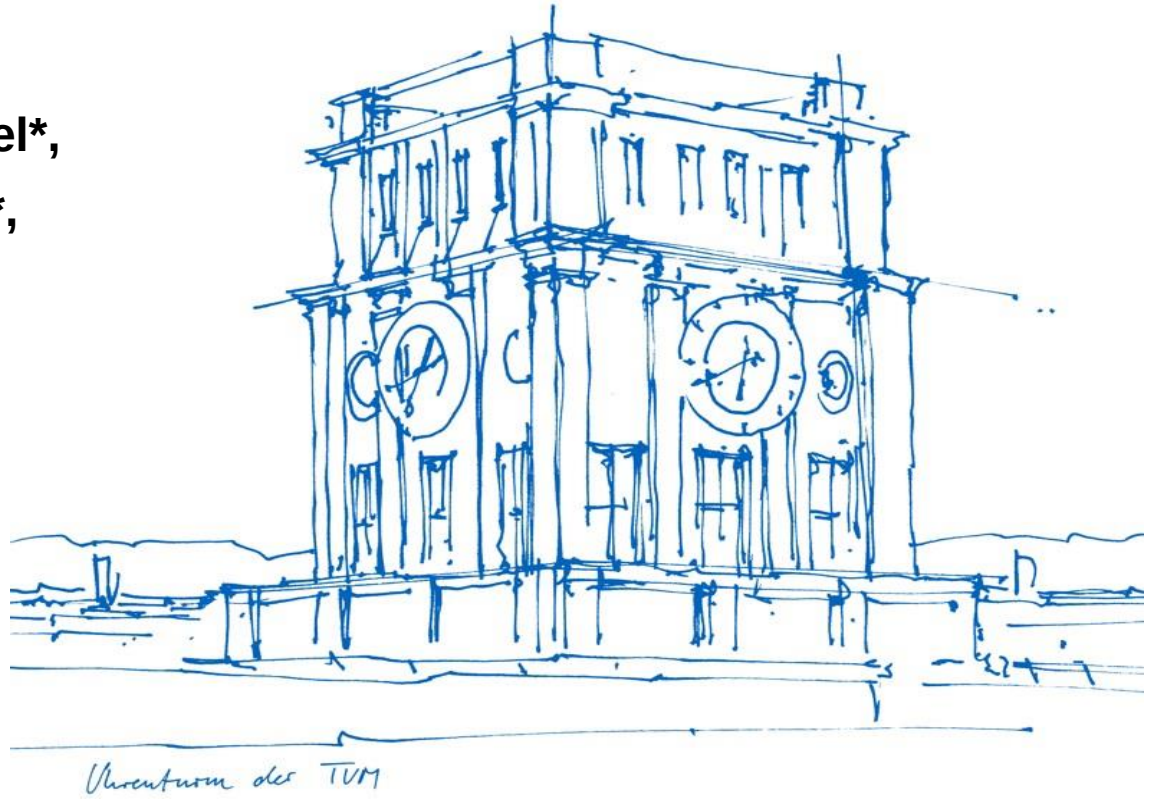
Andreas Blenk*

Joint work with:

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

Andreas Blenk*

**Technical University of Munich, Germany*
^Budapest University of Technology and Economics, Hungary
°Faculty of Computer Science, University of Vienna, Austria



IRTF Agenda IETF106: nmrg: Fri 12:20

IETF 106, Singapore

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

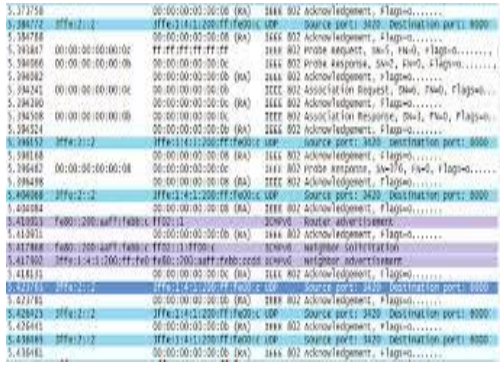
Self-Optimizing → Solution

Self-Benchmarking → Performance Evaluation

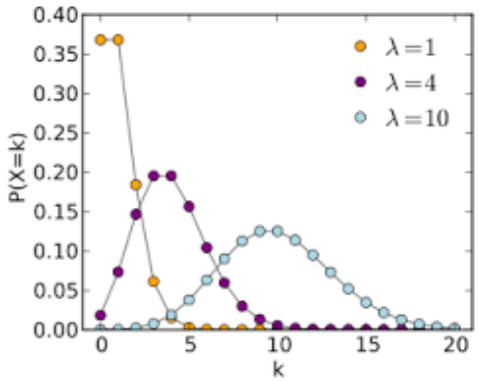
NetBOA (NetAI'19)

Benchmarking Network Algorithms, Architectures etc…

The Traditional Way …



Traces



Models

Human‘s Best Guesses

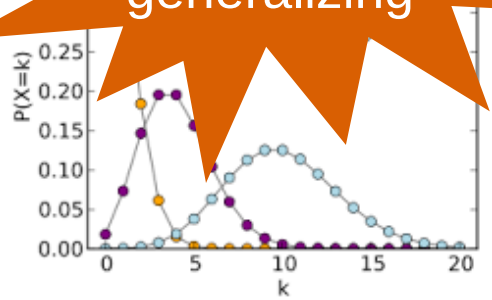
Data-Driven

Benchmarking Network Algorithms, Architectures etc…

The Traditional Way …

Not always available

Traces



Not generalizing

Models

Hmm… Biased?

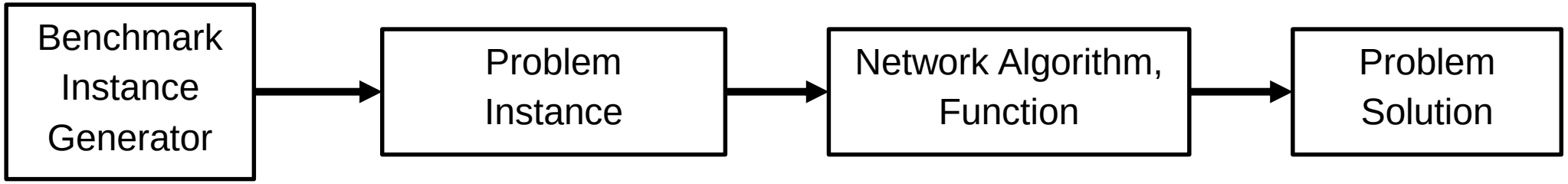
Human‘s Best Guesses

Alternative opponent?

Data-Driven

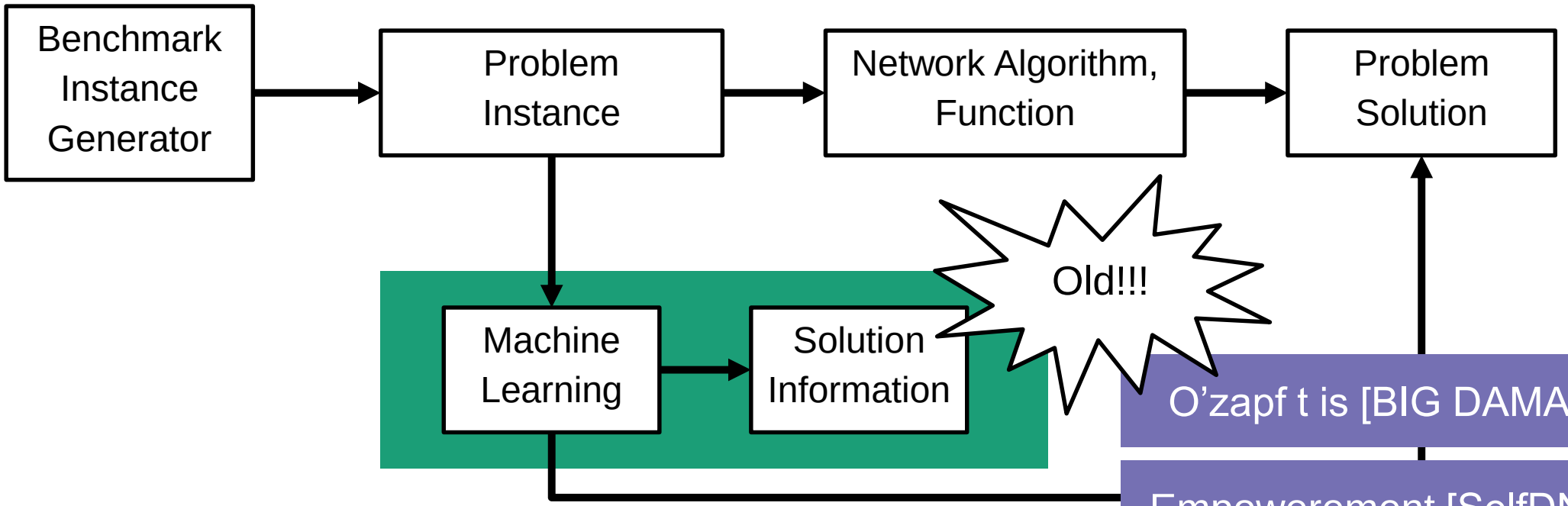
This Talk: Use Machine Learning to Benchmark Networks

Towards Automated Network Optimization and Design



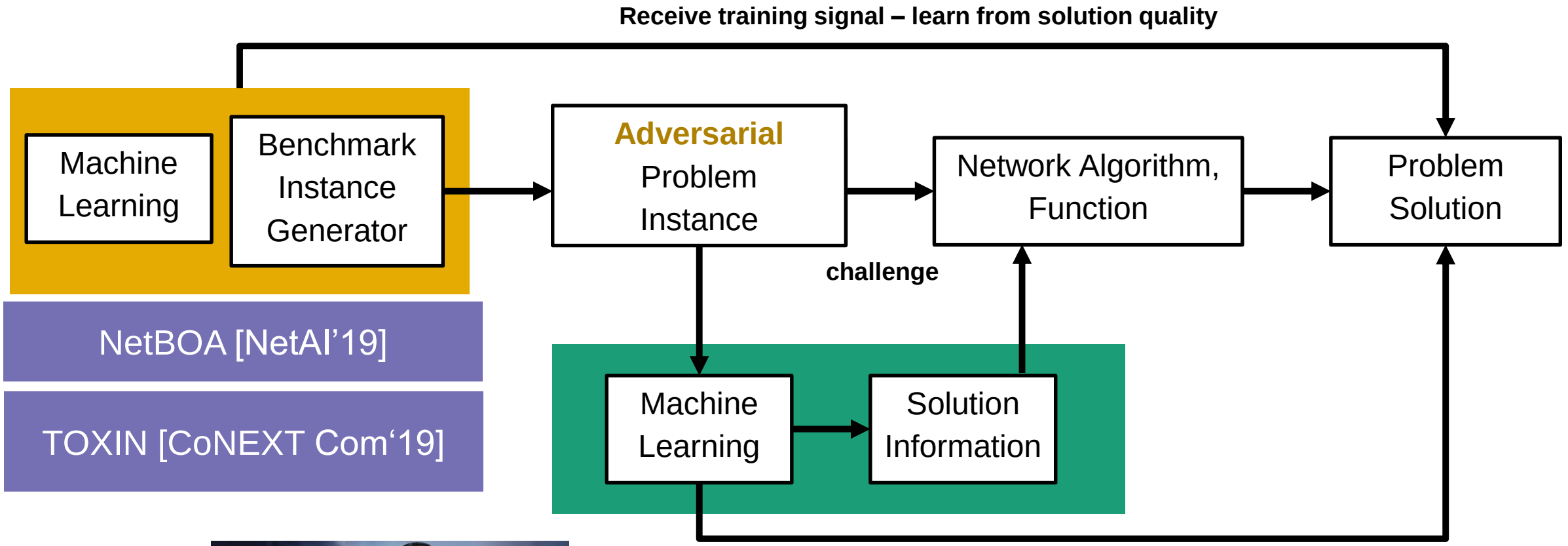
The Traditional Way!

Towards Automated Network Optimization and Design



Our ML/AI Way!

Towards Automated Network Optimization and Design



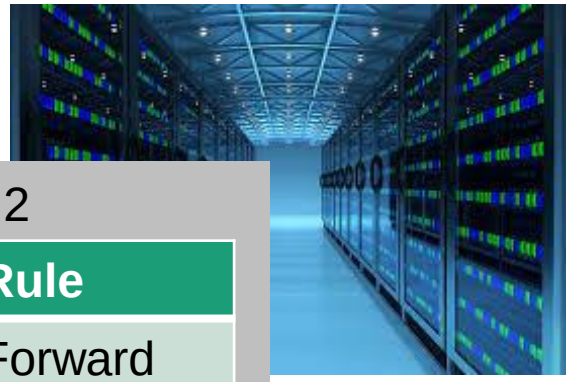
NetBOA [NetAI'19]

TOXIN [CoNEXT Com'19]

Our ML/AI Way!

ML/AI vs ML/AI

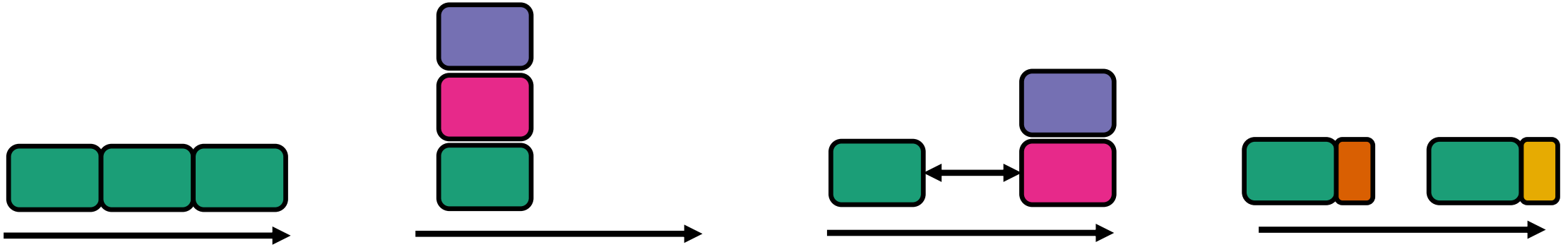
Example: Benchmark Open vSwitch



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]

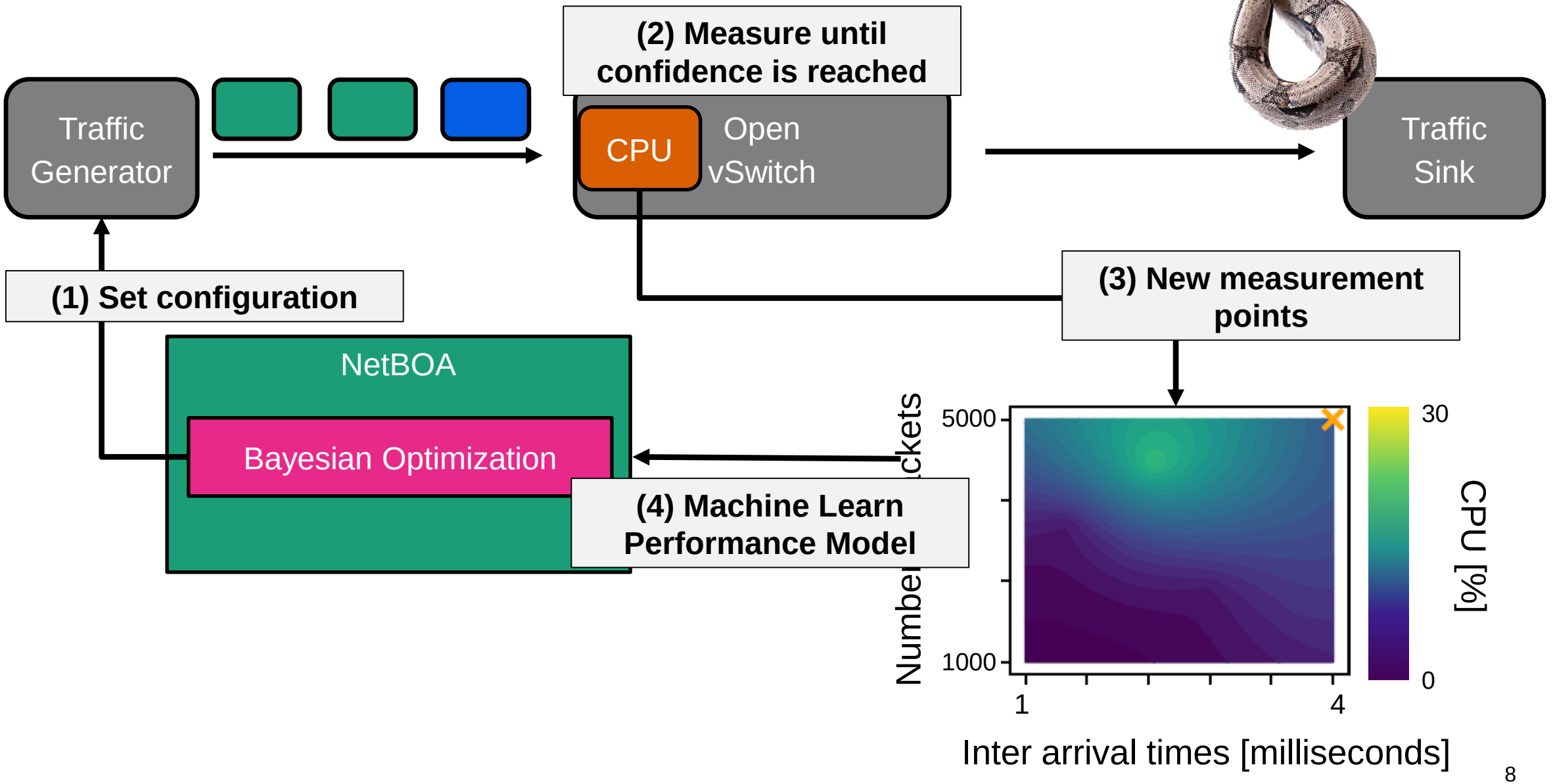
Batch Size [1-5]

Packet Inter Arrival Time [1ms – 13ms]

VLANs [1-5]

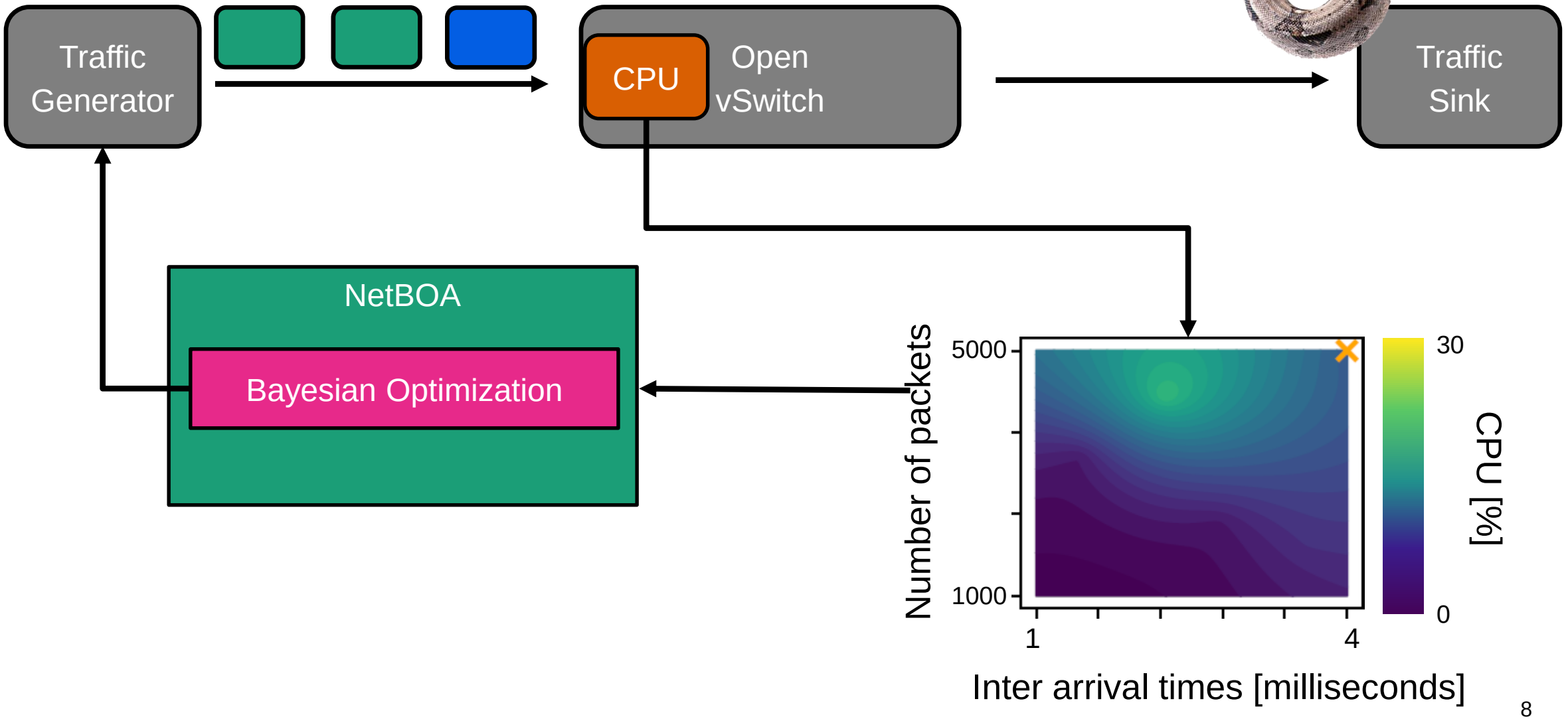
Source: https://de.kisspng.com/png-pc3d06/preview.html

NetBOA: A Bayesian Optimization-based Approach

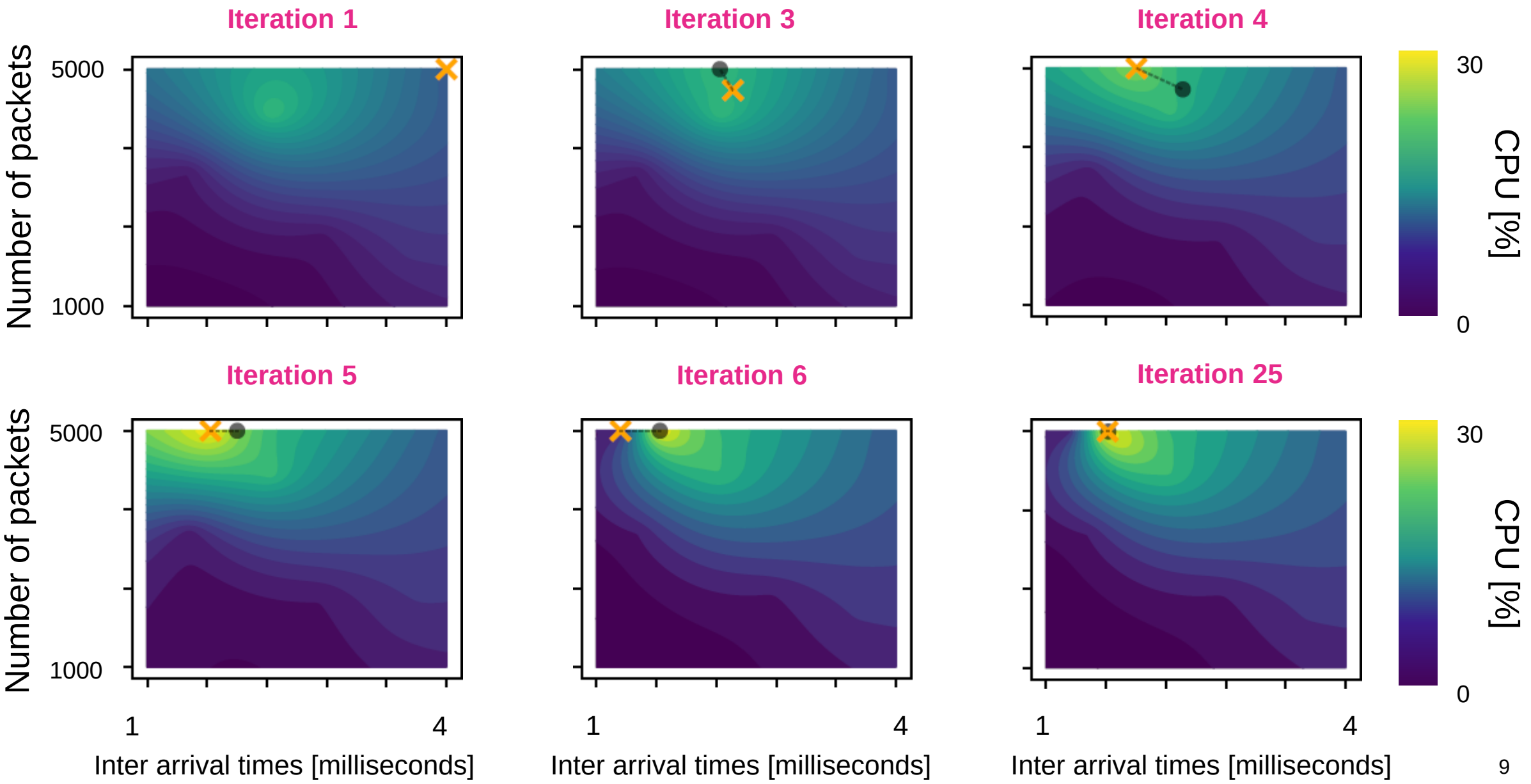


Source: https://de.kisspng.com/png-pc3d06/preview.html

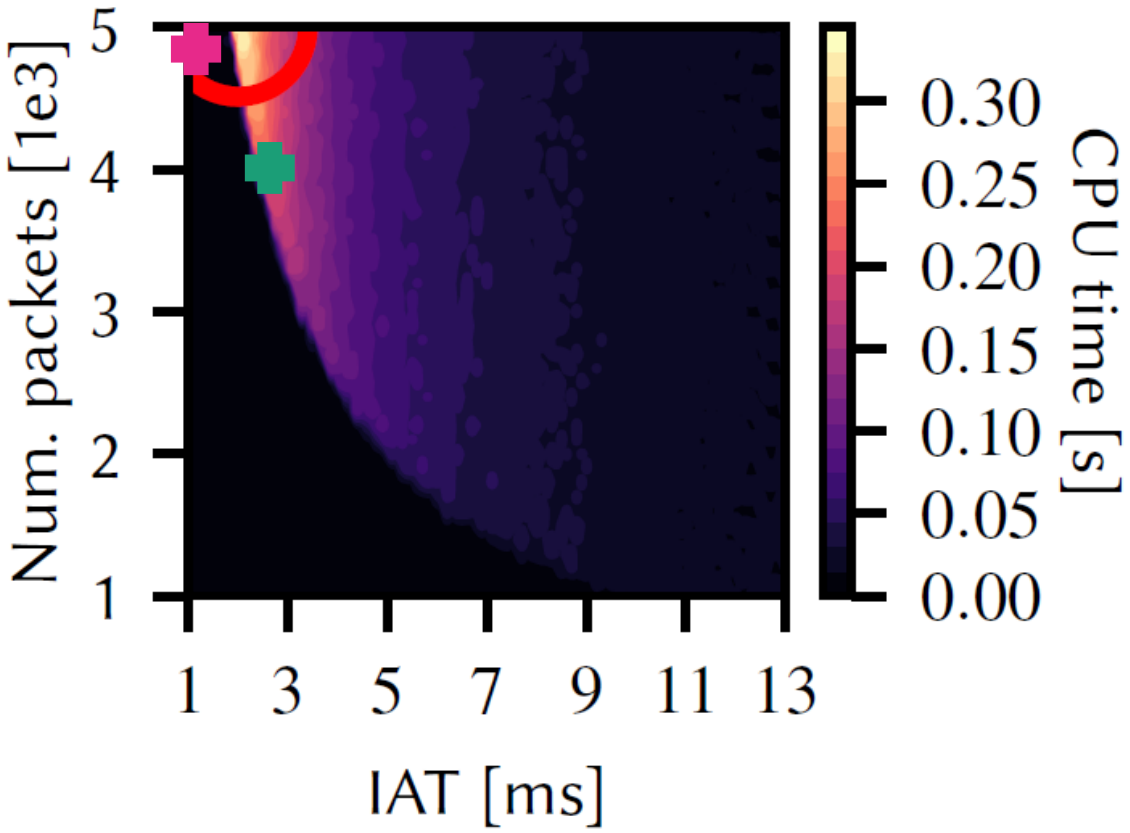
NetBOA: A Bayesian Optimization-based Approach



How NetBOA Explores the Performance Model



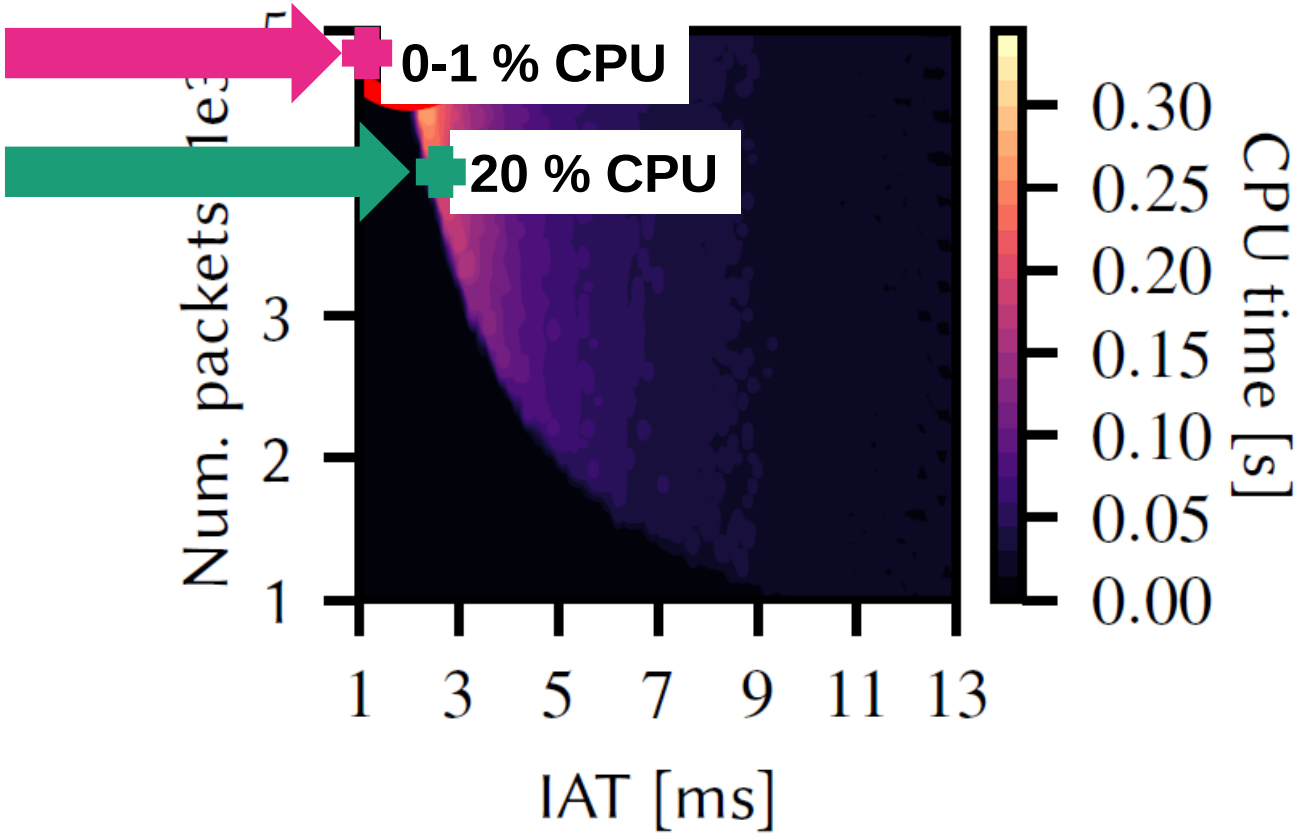
Grid Search for Two Parameters (Num. Packets and Inter Arrival Time)



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

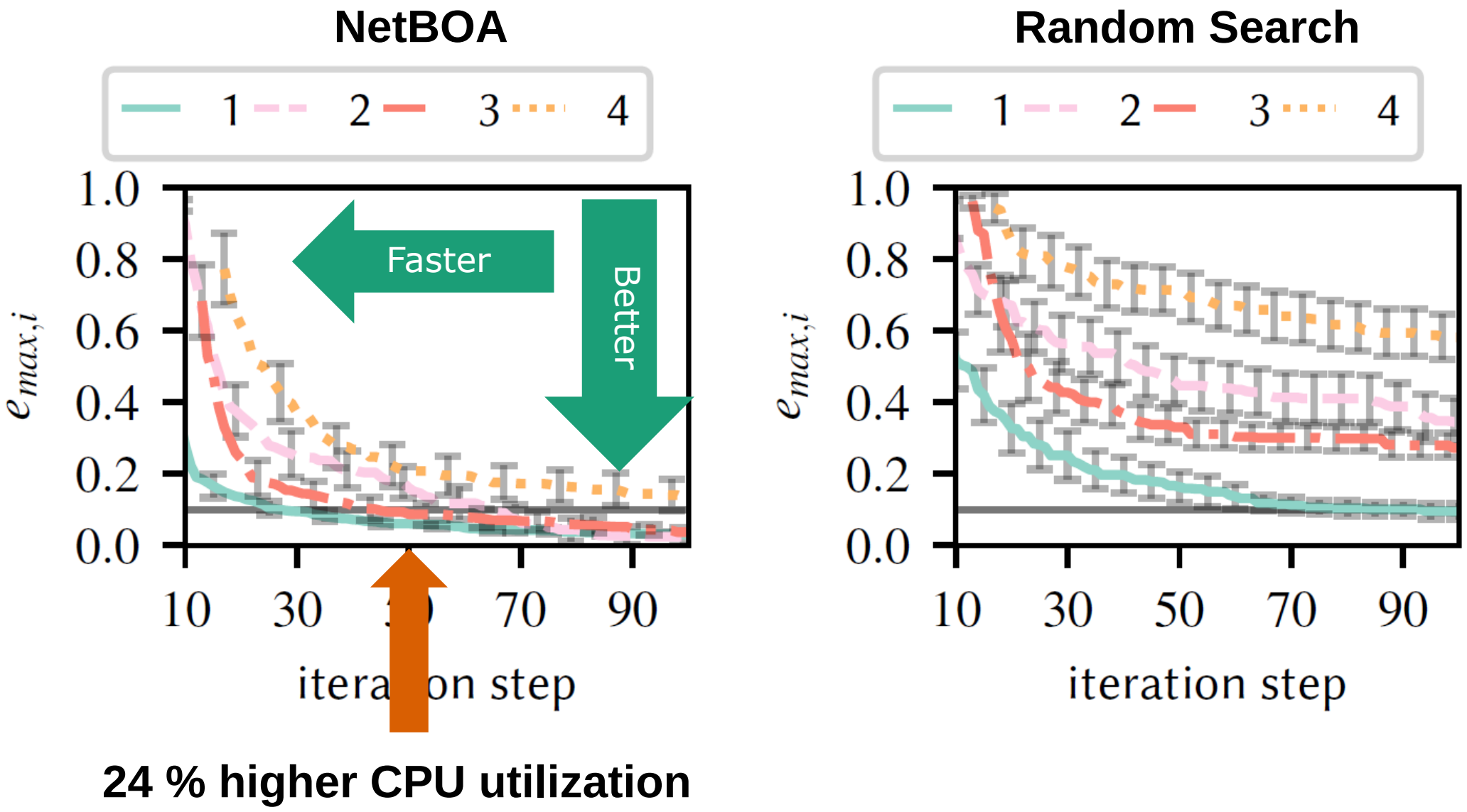
Every packet triggers 2 times a costly array resizing operation!

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



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!

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 Com‘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?

What Could be Seen as Related

- Algorithmic complexity attacks (software domain):
  - SlowFuzz
  - PerfFuzz
- *Automated Synthesis of Adversarial Workloads for Network Functions*, ACM Sigcomm 2018
- **Policy Injection: A Cloud Dataplane DoS Attack**, ACM Sigcomm DEMO 2018

Why Important?

Implementation aspects can harm performance

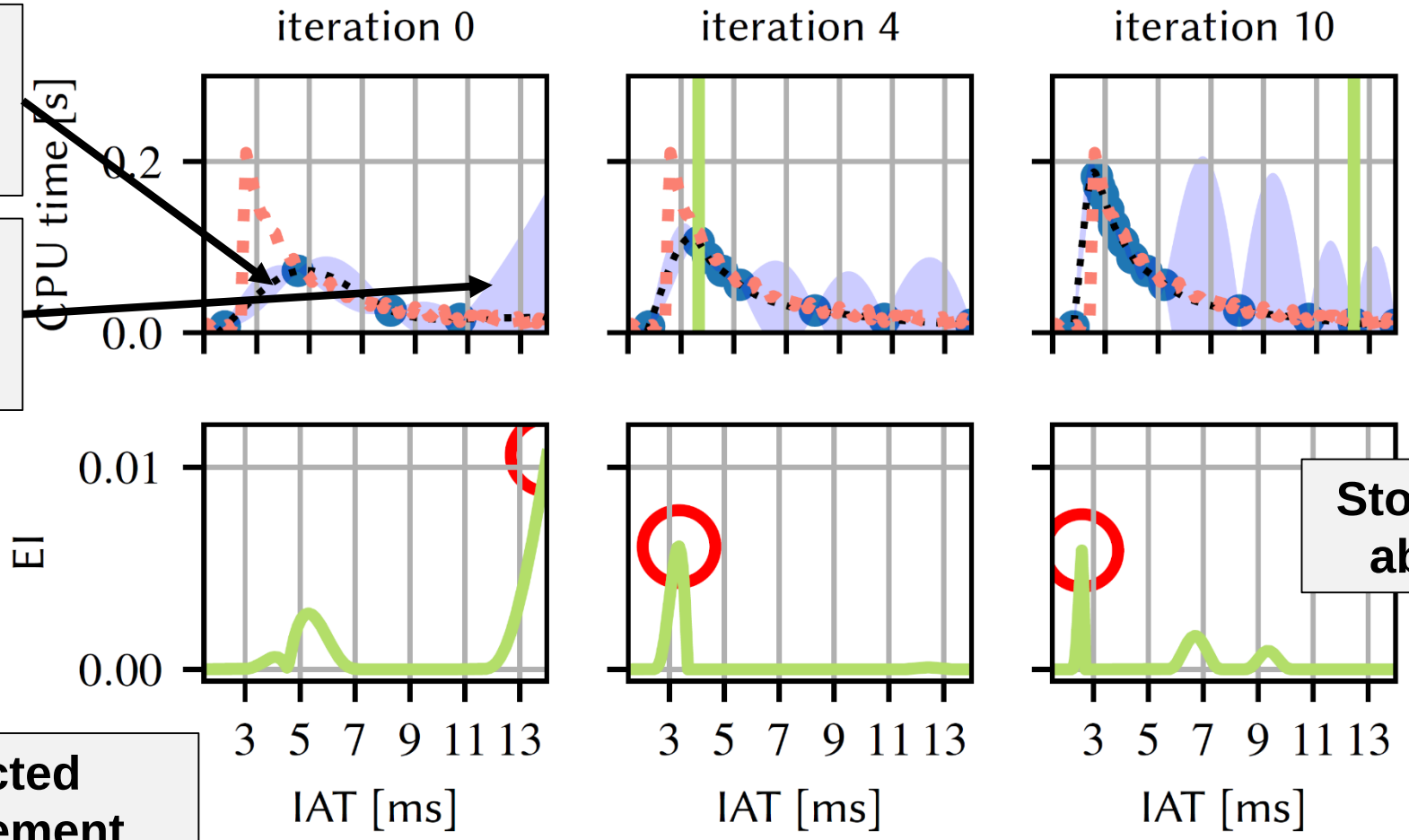
Could even be used to attack your systems!

We propose NetBOA to automatically create network traffic input

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

Update Gaussian Process at runtime

Sampling from Gaussian Process gives confidence



Stopping criteria aborts search

Expected Improvement guides search