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



Adversarial Network Benchmarking

Andreas Blenk*

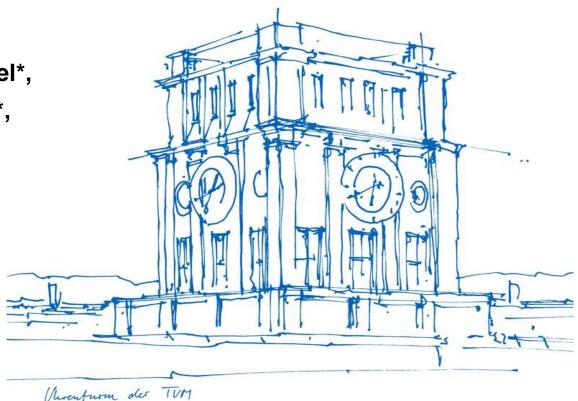
Joint work with:

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

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

Communication Technologies Group, Faculty of Computer Science, University of Vienna

©2019 Technical University of Munich



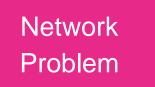






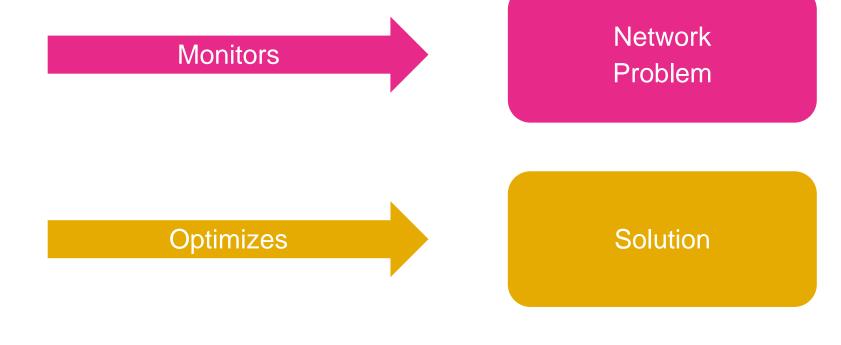




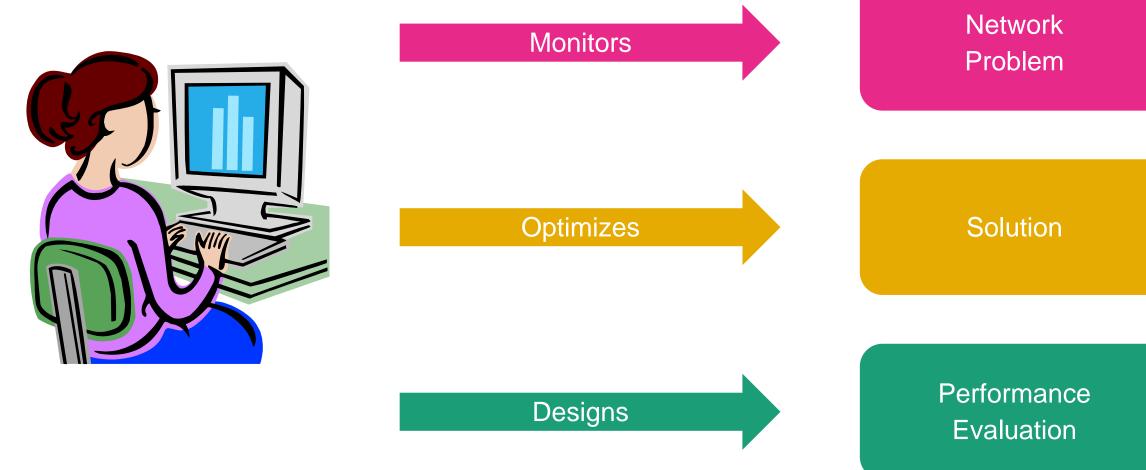




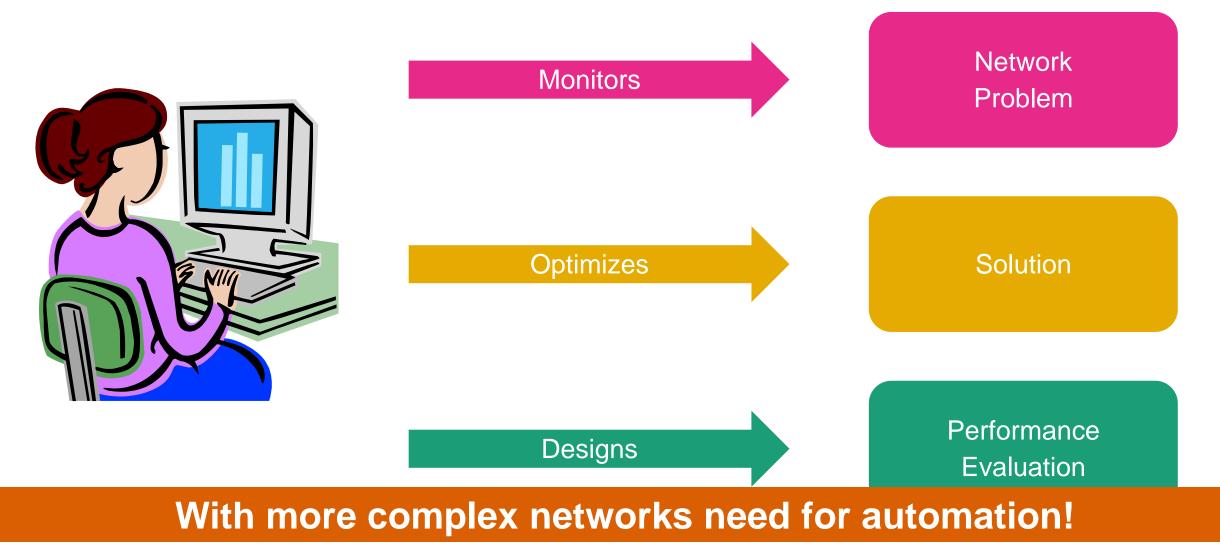




ТШ



















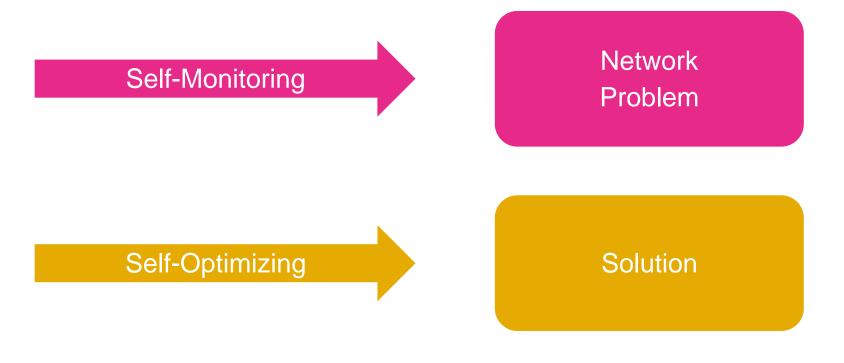
Source: https://www.pinterest.at/pin/318137161149129652/

Self-Monitoring

Network Problem

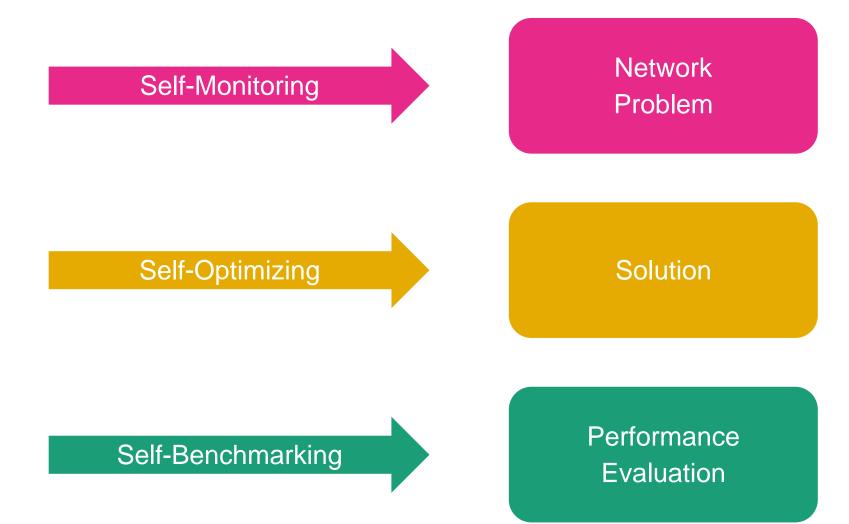
ТШП





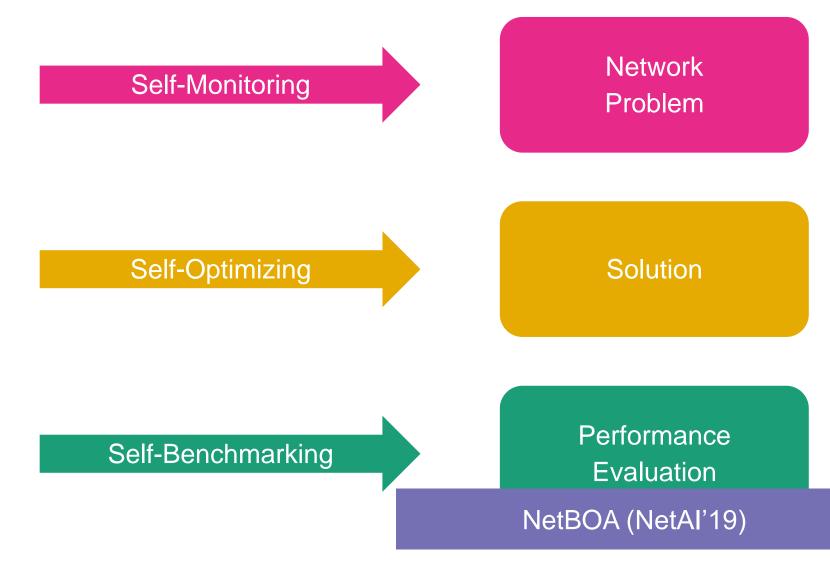
ТШП





ТШП



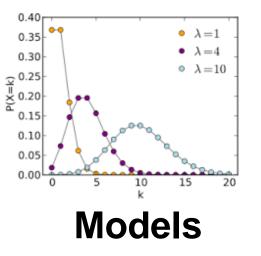


5.373758	1999 V 1	00:00:00:00:00:00 (RA)	2648 902	Acknowledgement, \$lags-c
3,384772	offering	dffei314(1)200(ff)(fe00)		pource porti 3020 Destination porti 80001
5,334784	2-11-11-11-1-1-1-1-1-1-1-1-1-1-1-1-1-1-	00:00:00:00:00:00:08 (RA)		Acknow ladgement . # lags-ti
5.292847	00:00:00:00:00:00	tf.ff.ff.ff.ff.ff	TEFE 802	Probe sequent, 18-5, FN-0, #lags-0
5.534066	40:00:00:00:00:00	90:00:00:00:00:0c		Probe Response, SNR2, ENRO, Flagmon,
5.294542		00:00:00:00:00:00 (MA)		Acknowledgement, Flagtwo
3.94241	30:00:00:00:00:00	60.00.00.00.00.00		Association Request, 50w0, 70w0, rlagsw0
5,234200		00:00:00:00:00:00 (RA)		Acknowledgement, Flags-o
5,394508	00100100100100100	00:00:00:00:00:00		Association Response, 5%-1, FN=0, Flags=0.
5.594524	the second second second	00:00:00:00:00:00 (RA)		Acknowledgement, Flags-G
50295152	3494:21:2	offective 1200 off cheory		maunce port: 3120 pentination port: 6000
5,594168	Contraction of the	00:00:00:00:00:08 (RA)		Advinov ledgement , # Lags=0,
5,296462	00:00:00:00:00:00	00:00:00:00:00:00		Probe ampoints, 34-176, Fa-0, Flags-c
5,396498		00-00-00-00-00-08 (AA)		Adviowledgement, flags=0
5.404068	Mfe:2::2	Hfu:1:4:1:200:Ff:fe00:	CUP	Source port: 3420 Destination port: 6000
5.404084		00:00:00:00:00:08 (MA)		Adonowledgement, #lags=0,
5.410903	Testination affirmation			Router advertigeners
5.410911		00:00:00:00:00:00 (KA)		Acknowledgement, Flagt-0
STAT7BER 1	rfesori 200 i april rfebbre			Netgebor solfcitation
5.417002	Ittesio4sis200sttste0	fello: :200:autt:fxbb;ccc	tel acepyd-	Neighbor advertisement
3,418135		00:00:00:00:00:00 (8A)		Addrowledgement, flags=0
5.421761	Manual Contraction	CLOSE RESIDENCE LA COM	C 110	Source port: 3420 Deptimation port: 6000
5.423785	CNT AND A	(AN) 40:00:00:00:00:00	THE 802	Adotovledgenent, Flagsso
5,426423	2014-21/2	Jffe:1:4(1)200(ff)/fe00)		Source port: 3420 Destination port: 60001
5.426441		00:00:00:00:00:00 (RA)	1118 (0)2	Actional edgement, #lagu-0
5.438183	3ff#121/2	Jffei3(4)11200(ff(fe00)	ENDERSON	source port: 3420 Destination port: 6000:
5.43640	1.1	00:00:00:00:00:00 (0b (0kA)	3666.002	Acknowledgement, +lags=q



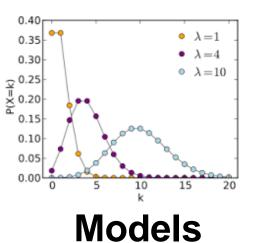
5.373758	1889 V 3 6	00:00:00:00:00:00 (RA)	1118 0.	2 Acknowledgement, Flags-c
51384772	attering	3ffei1(4)1200(ff)(fe00)(LOP	pource porti 3470 Destination porti 8000
5,334784	e alle i de la companya de la	00:00:00:00:00:00:08 (RA)	1666.00	2 Acknow ledgement . # Jags-0
5.393847	00:00:00:00:00:00	ff.ff.iff.ff.ff.ff	THE &	2 Probe Request, 18-5, FN-0, #lags-c
5.534566	40:00:00:00:00:00	90:00:00:00:00:00	1666 80	2 Probe Response, SNI2, FNID, Flagsvo
5.294542		00:00:00:00:00:00 (ma)		2 Acknowledgement, slagtwo
3.394241	30:00:00:00:00:00	60.00.00.00.00.00	3111 8	2 Association Request, SNuo, FNuo, Flagsun.
5.234200		00:00:00:00:00:00 (RA)	1666 80	2 Acknowledgement, Flags+0
5,394508	00:00:00:00:00:00	00:00:00:00:00:00		2 Association Response, DN-1, FN-0, Flags-0.
5.594524		00:00:00:00:00:00 (RA)		2 AddrewTedgement, Flags-0
1.295152	3174:21:2	Iffect140102000ffcfe0000	LEP-	indunce port: 3420 pentination port: 6000
5,594168	Sugar Astronom	00:00100100:00:08 (RA)	1666 80	2 AdvnowTedgement , + Tags=0,
5.296482	00:00:00:00:00:00	00:00:00:00:00:00		2 Probe Assponse, SA-176, FA-0, Flags-c
5,206498		(AA) 80100100100108 (AA)		2 Advovledgerent, flags=0
5.404068	Mfet2112	Hfu:1:4:1:200:ff:fe00:c	UCP	Source port: 3420 Destination port: 6000
5.404084		00:00:00:00:00:08 (MA)	THEE OF	2 Adonowledgement, #lags=0,
3.410923	Te801:200.saffrfebb;c	HWHITHERE	SCHEVE.	Router advertisement
5.410911		00:00:00:00:00:00 (RA)	2666.00	2 Acknowledgement, Flagt-0
STAT7BERT	16801200120120711680000	ff02112)ff0016	(SCHOOL)	Nationed Solfcitation
5.417902	Ittesio4sis200stfste0	fe60::200:astf:febb;ccdd	DOMENTO-	weighbor advertisement
3 418135		00:00:00:00:00:00 (8A)	1111 80	2 Addrowledgement, flags=0
57421763	CHICANNES -	LICORRESPONDENCES INCOME	110	Source port: 3420 Deptimation port: 6000
5.423785	CNT AND A	00:00:00:00:00:00.00.00.00.	1118 8.	2 AdonowTedgenent; Flags-0
5,426423	3164:21/2	JFFe:1:4(1)200/FFife00:0	UDP	Source ports 3420 Destination ports 6000:
5.426441	8-22294	00:00:00:00:00:00 (RA)	THE R	2 Acknowledgement, #lags-0
5.438183	31Te325/2	3ffei3(4)11200/ffife0010	ODP.15	source port: 3420 Destination port: 6000:
5.436481	State 1998	00:00:00:00:00:00 (ka)	2666.00	2 Acknowledgement + lags=0

Traces



пп

5.373758	1000 V/3 0	00:00:00:00:00:06 (RA)	1668-00	2 Acknowledgement, Flags-d
51384772	atterne .	3ffe(1)4(1)200(ff)(fe00)(c	LOP	bounce ports 3070 Destination ports 8000
5,334784	Color Carlos	00:00:00:00:00:00 (RA)	1666.00	2 Acknowledgement, Flags-0
5.393847		ff.ff.ff.ff.ff.ff.ff		2 Prote sequett, IN-5, FN-0, Flags-0
5.534566	00:00:00:00:00:00	90:00:00:00:00:0c		2 Probe Response, SNR2, ENRO, Flagson,
5.294542		00:00:00:00:00:00:00		2 Acknowledgement, slagtwo
3.94241	30:00:00:00:00:00	60.00.00.00.00.00		2 Association Request, 5846, 7840, Flags-0
5.234200		00:00:00:00:00:00 (RA)		2 Acknowledgement, flags+0
5,334508		00:00:00:00:00:00		2 Association Response, DN=3, FN=0, Flags=0
5.594524		00:00:00:00:00:00 (RA)		2 Acknowledgement, Flags-0
5.296152	3174:21:2	offection:200:ff:fe00ct	LEP	hource port: 3420 pentination port: 6000
5,5941.68		00:00100:00:00:08 (RA)	1666 80	2 AdvisorTedgement, + Tags=0,
5,296462		06:00:00:00:00:00		2 Probe senporne, 34-176, Fa-0, Flags-c
5, 206498		00:00:00:00:00:08 (AA)	ttfr st	2 Admowledgement, flags=0
5.404066	Mfet2112	UFFect:4:1:200:FF:Te00:c	LEP	Source port: 3420 Destination port: 6000
5.404084		00:00:00:00:00:08 (MA)	1688 (0)	2 Adonavledgement, #lags=0
5.410933	Te801200.saff(febb)C			Router adventisement
5,410911		00:00:00:00:00:00 (KA)		2 Acknowledgement, Flagt-0
STAT7BER	100012001207100000		(BORING)	
5.417902	Ittesi sisi s200sttate0	fell0::200:astf:febb;ccdd	ackeyd-	
5,418135		00:00:00:00:00:00 (8A)	ILLE 80	z Adonavlędgesent, tlags=0
574217151	CHIANNIS C	LITCOLICE DISTORT OF A DESCRIPTION OF A		Source port: 3420 Destination port: 6000
5.423/85		00:00:00:00:00:00:00 (MA)		2 Adonavledgenent, Flagsso
5,426423	2014:21/2	101e-114(1)/200(00)/000(0	LOP	Source ports 3420 Destination ports 00001
5.426441		00:00:00:00:00:00 (RA)		2 Advoarledgement, #lags-0
5.438183	bffe121/2	STEELS WAT 200 TE (TeODIE	SEP-1	source port: 3420 Destination port: 6000
5.436461		00:00:00:00:00:00 (KA)	1666.00	2 Acknowledgement, +lags+0
10.00		and the second second	1118.99	



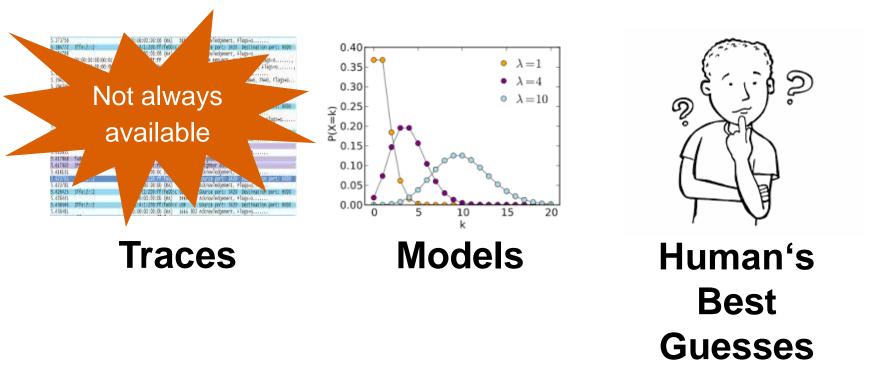




Ŷ

S

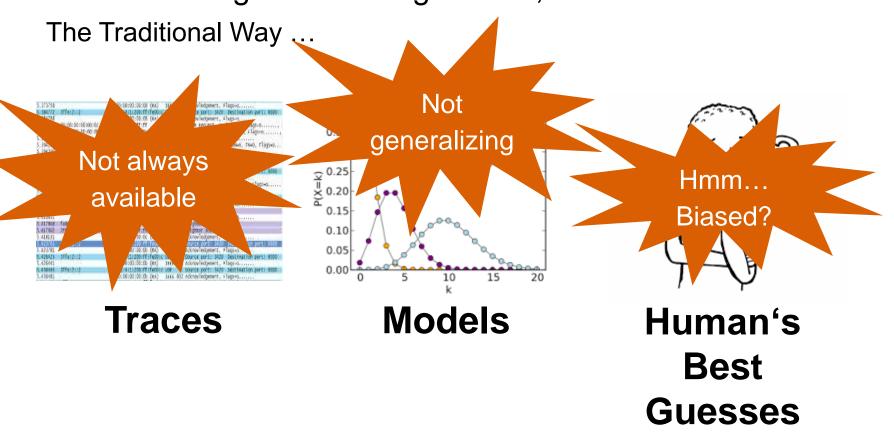
пп



The Traditional Way ... Not 5.373758 5.984772 #ff#s2ss :00:00:06 (R generalizing \mathcal{S} Not always Ż (¥ 0.25 = 0.20 × 0.20 available 0.15 5.417868 feb 5.417800 1* 5.418153 0.10 0.05 UPCE DOTT O MAN 0.00L 436163 MT#1210 SOURCE FOCET 3420 DES 20 0 5 10 15 **Models** Traces Human's **Best** Guesses

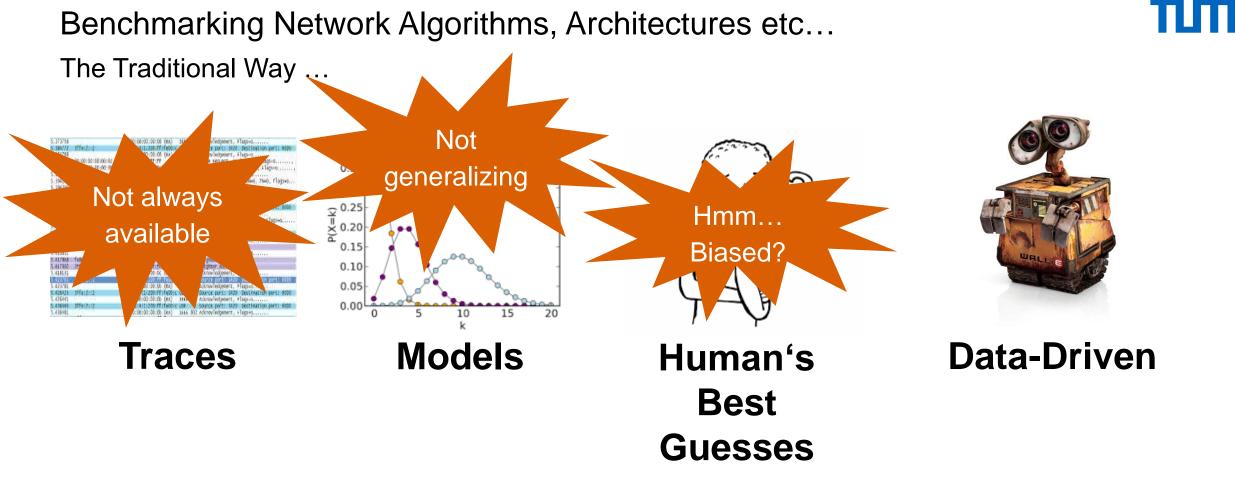
Benchmarking Network Algorithms, Architectures etc...

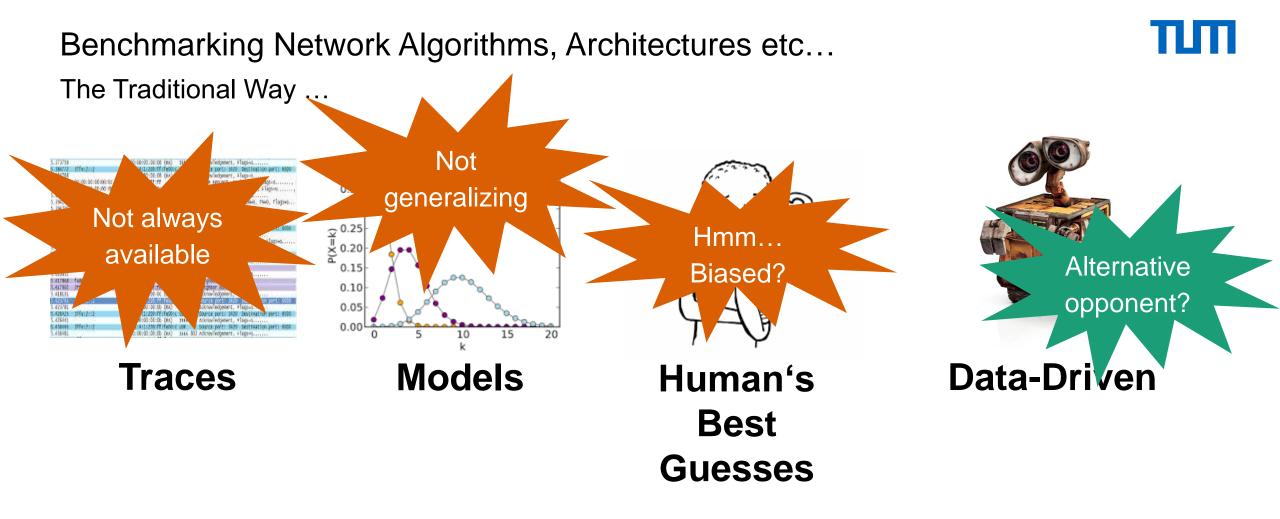
ПЛ

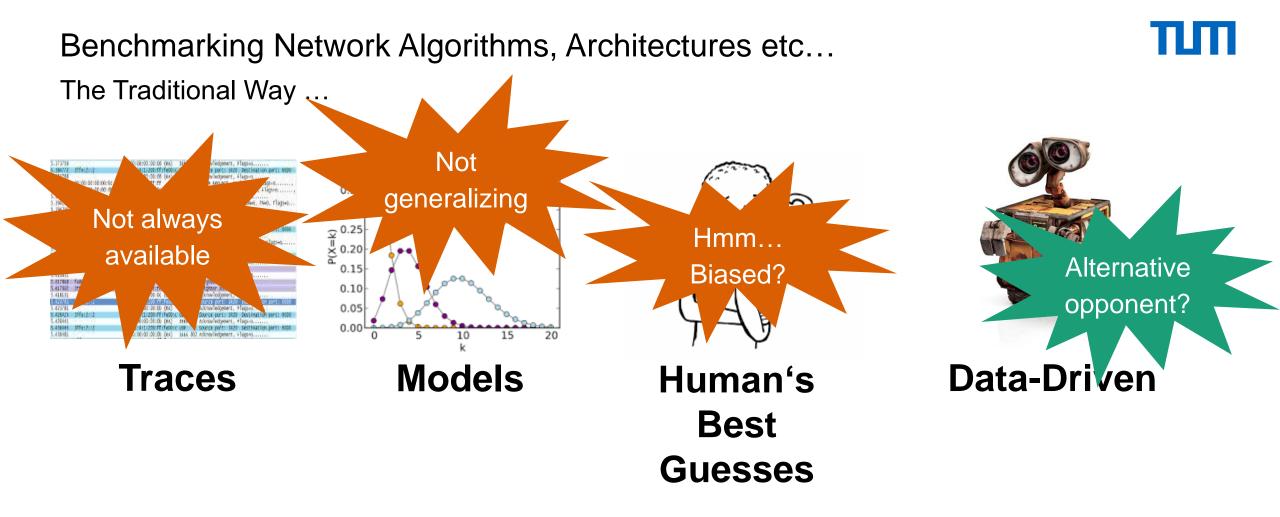


Benchmarking Network Algorithms, Architectures etc...

ПЛ





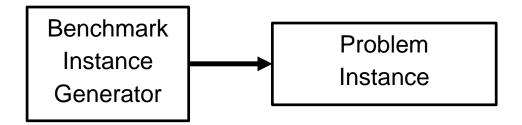


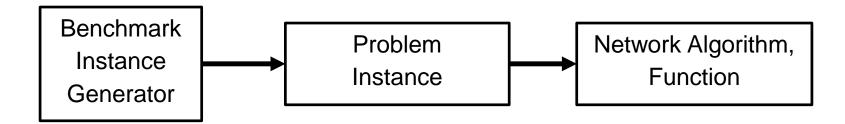
This Talk: Use Machine Learning to Benchmark Networks



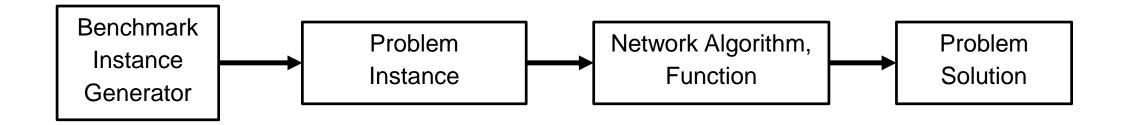
Benchmark Instance Generator

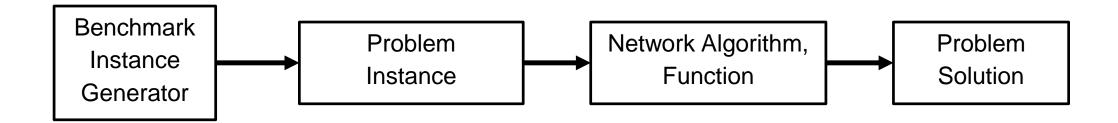






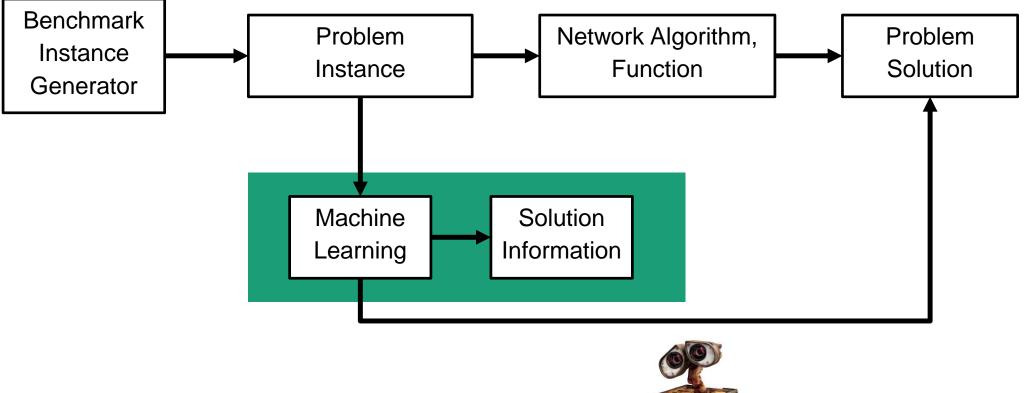






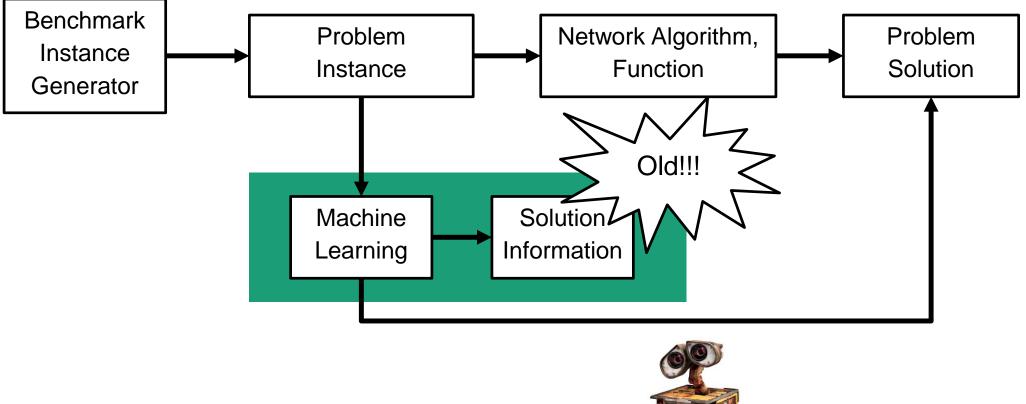






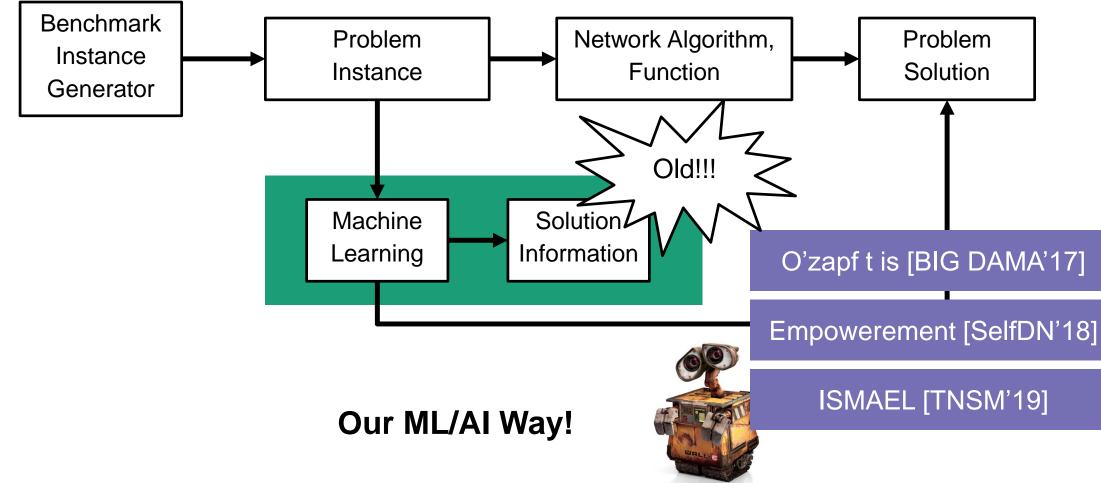




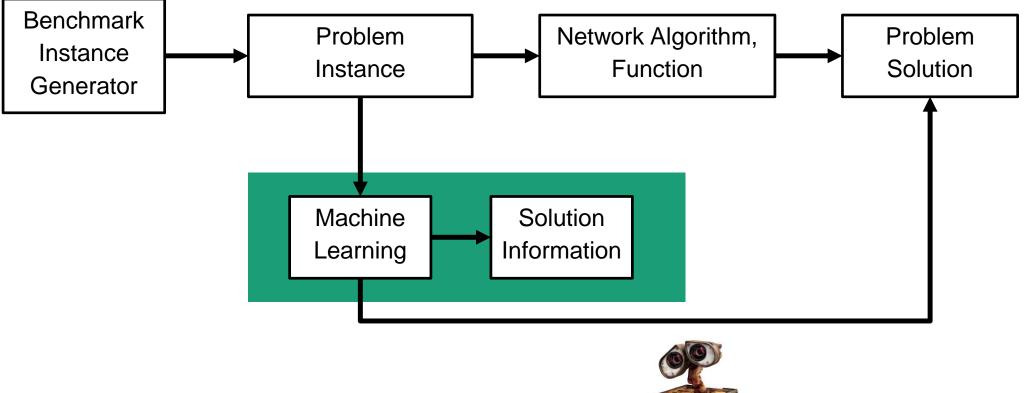




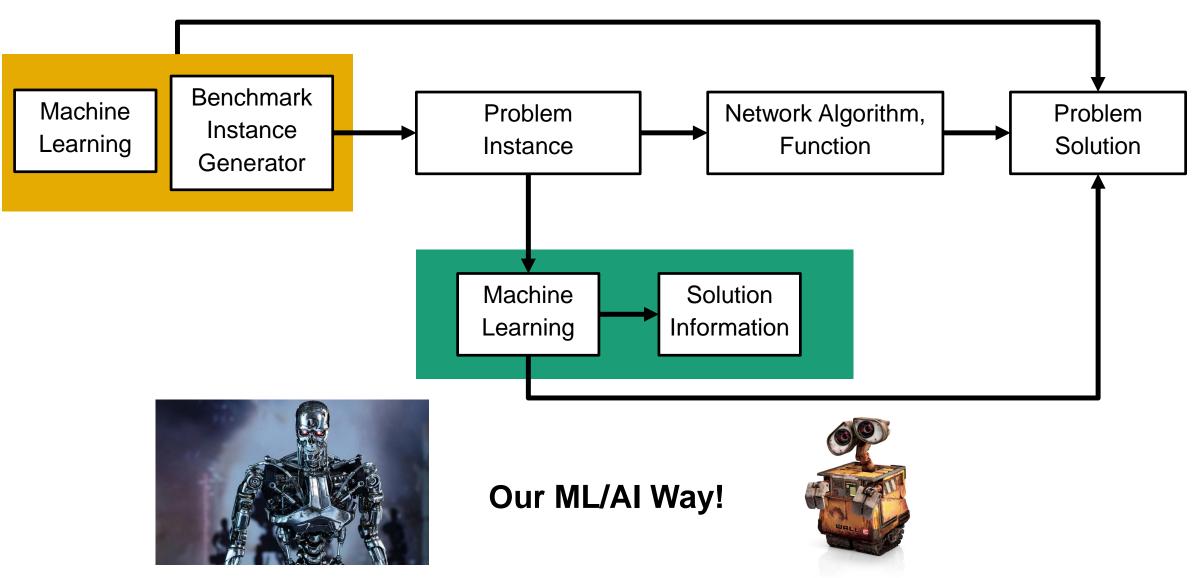








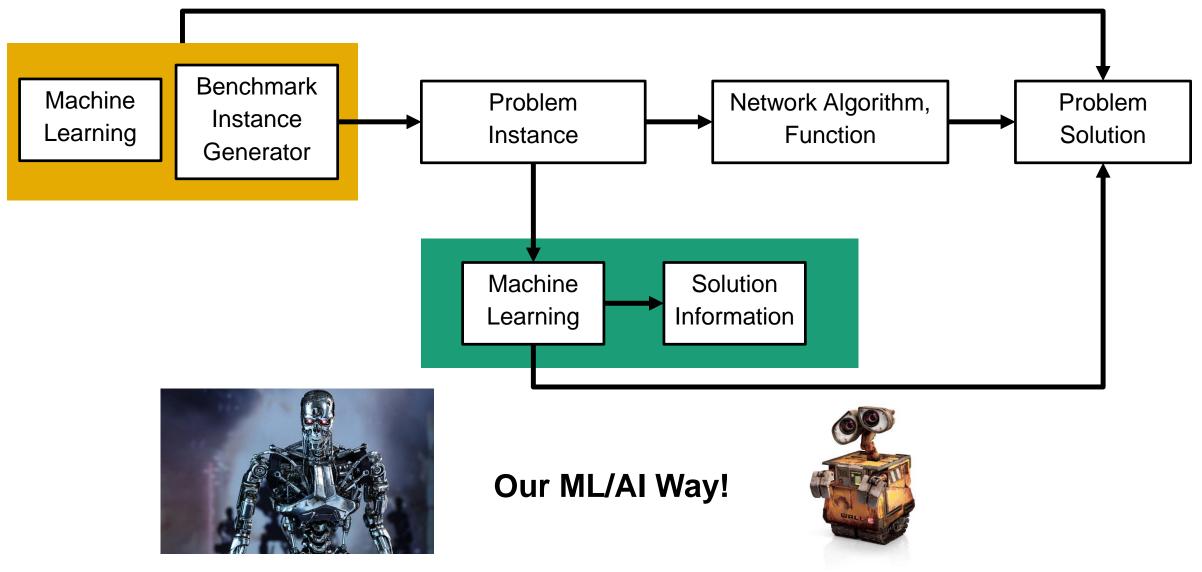




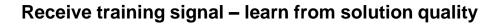


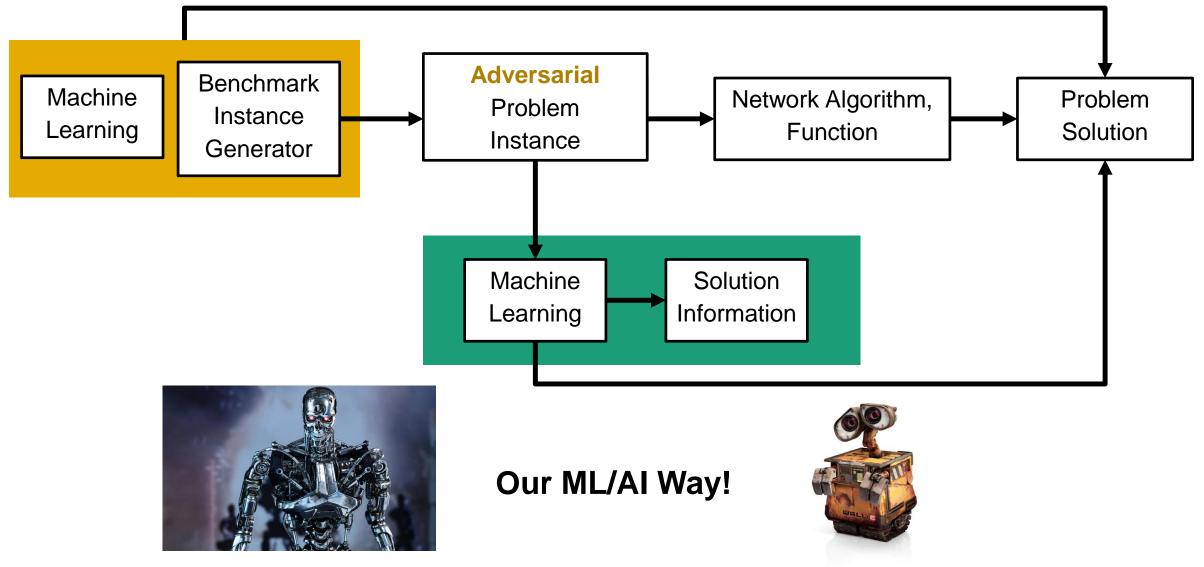
5

Receive training signal – learn from solution quality

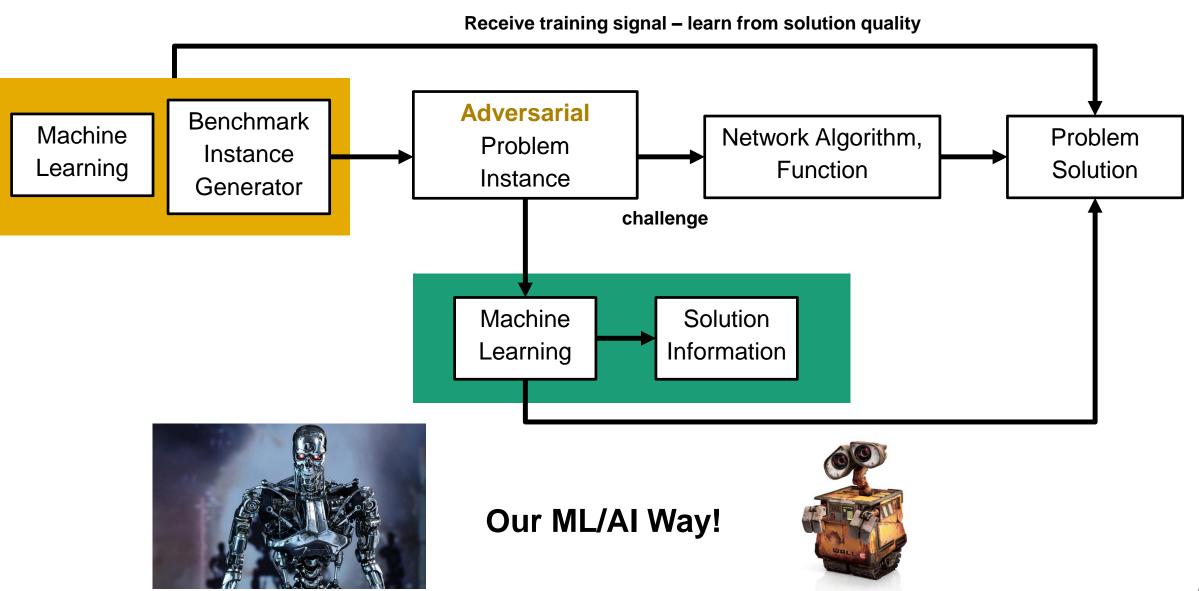




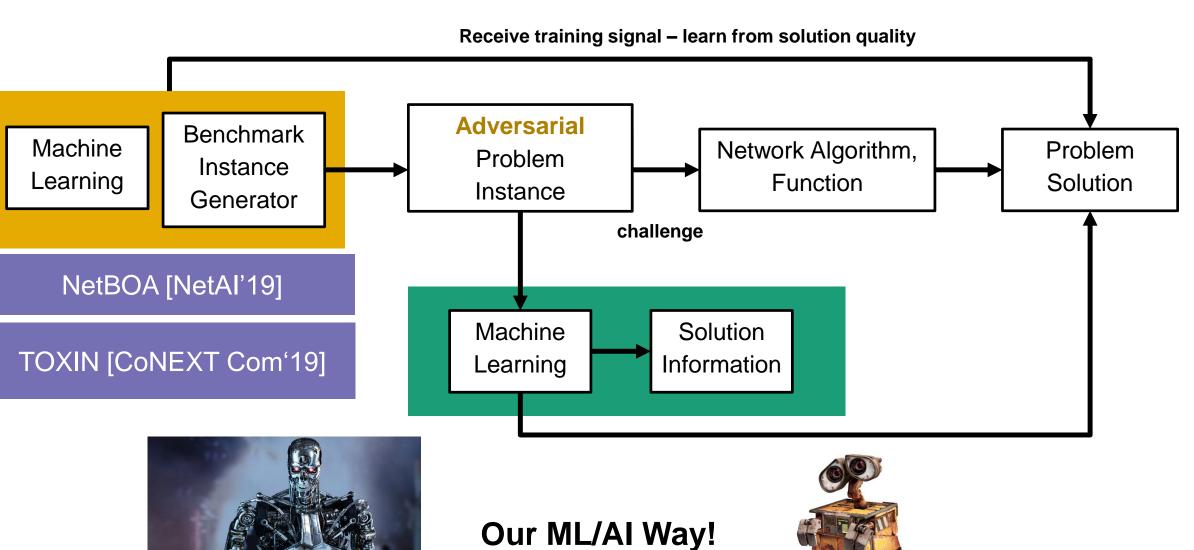












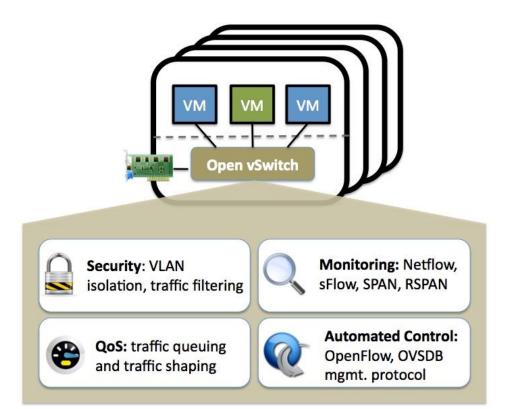
ML/AI vs ML/AI

5



Adversarial Network Algorithm Benchmarking: Use Cases

(1) Benchmarking Open vSwitch



(2) Benchmarking Data Center Traffic Scheduling Algorithms



TUT



(1) Benchmarking Open vSwitch



- Algorithmic complexity attacks (software domain):
 - SlowFuzz
 - PerfFuzz

- Algorithmic complexity attacks (software domain):
 - SlowFuzz
 - PerfFuzz
- Automated Synthesis of Adversarial Workloads for Network Functions, ACM Sigcomm 2018

ТЛП

- 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

- 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?

ТШ

- 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!

ТШП

- 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

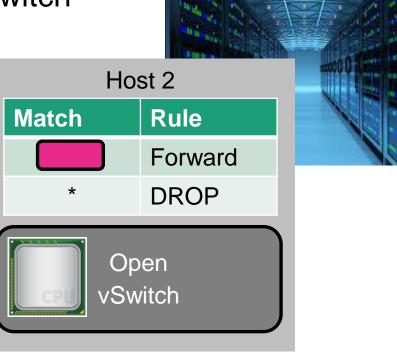
Why Important?

Implementation aspects can harm performance

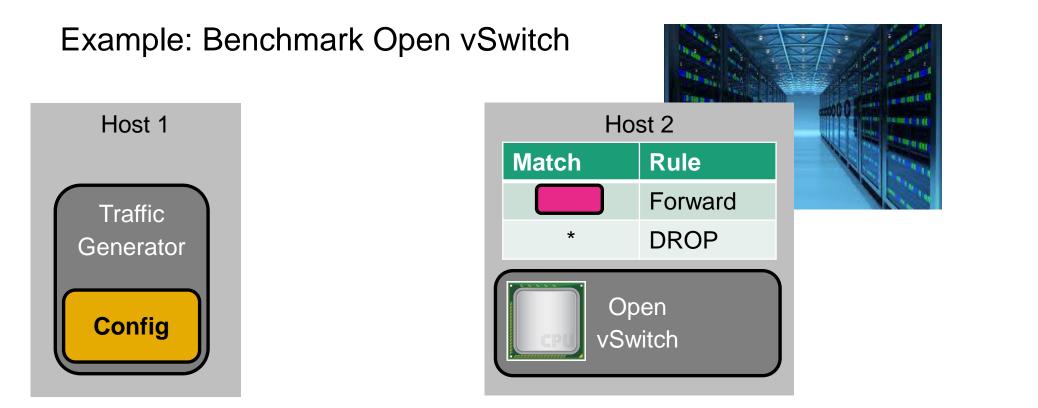
Could even be used to attack your systems!

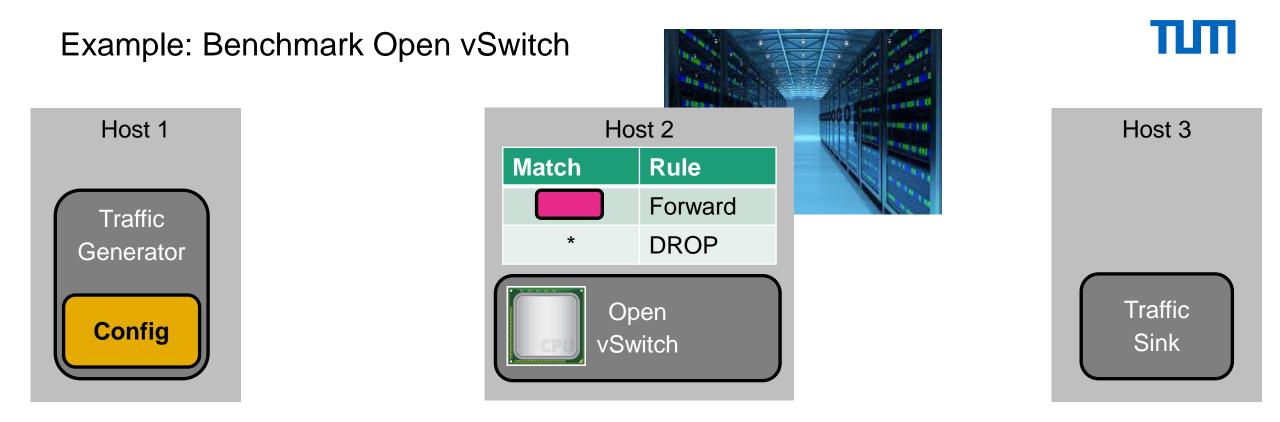
We propose NetBOA to automatically create network traffic input

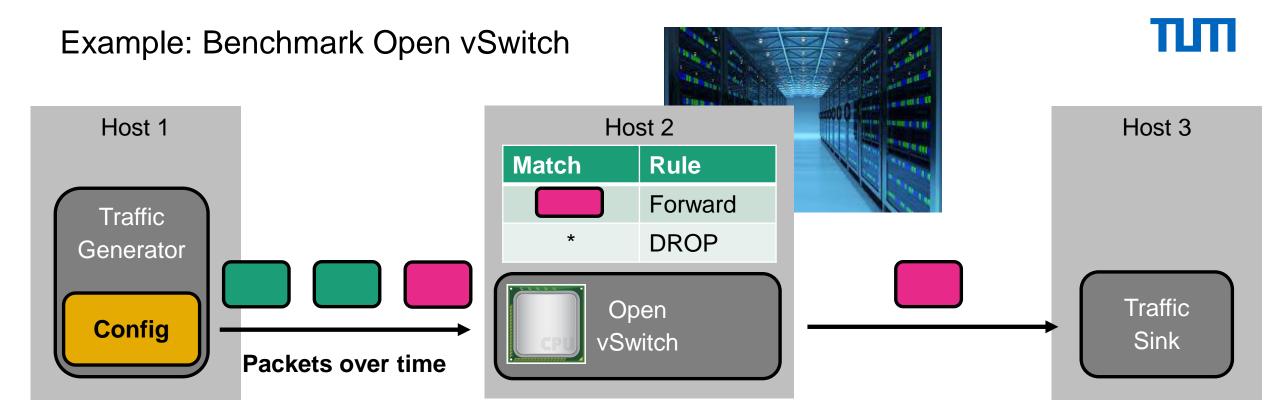
Example: Benchmark Open vSwitch

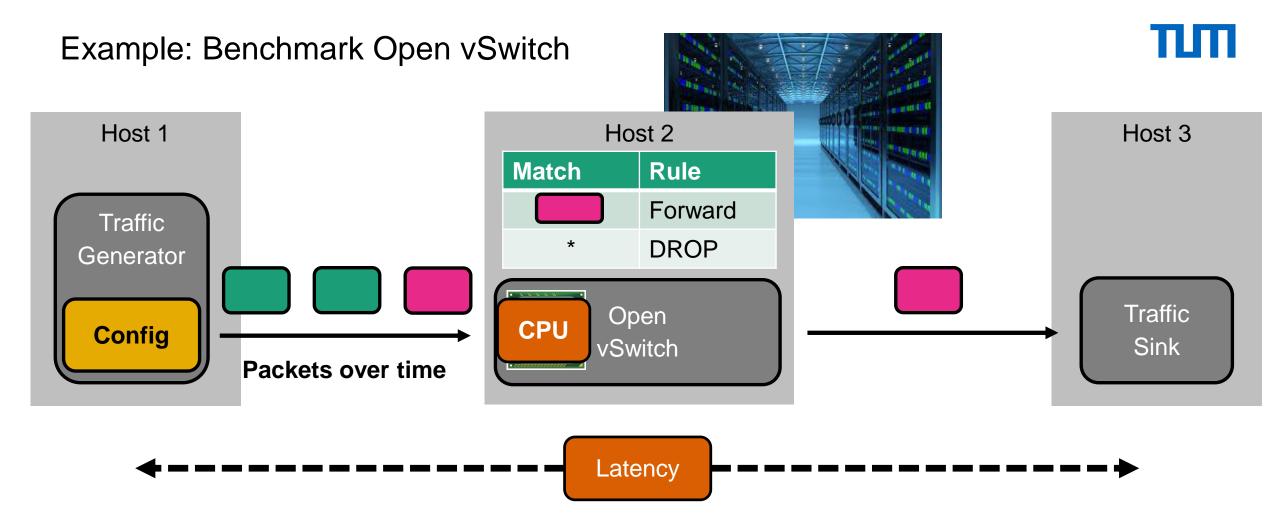


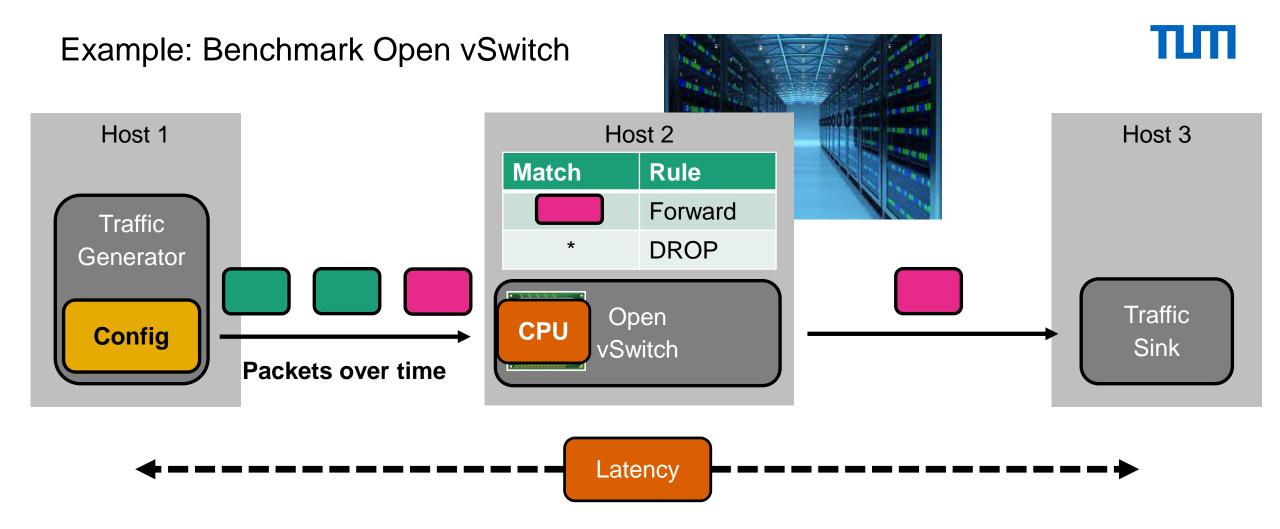
πт









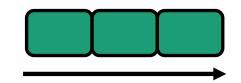


Goal: Find Network Traffic Configuration that Maximizes CPU/Latency



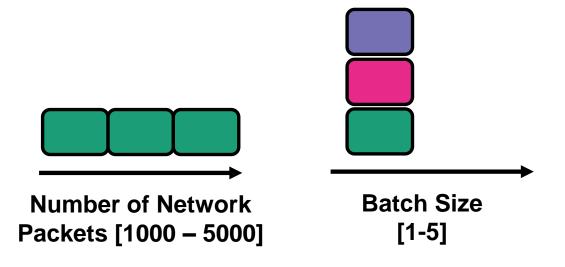


How many packets to send? How should headers look like? What protocol to use? When to send packets? Etc.

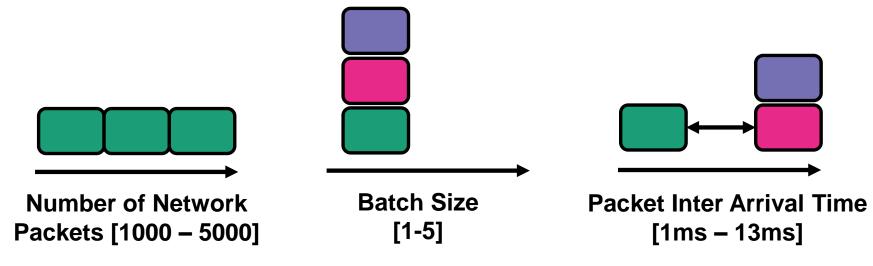


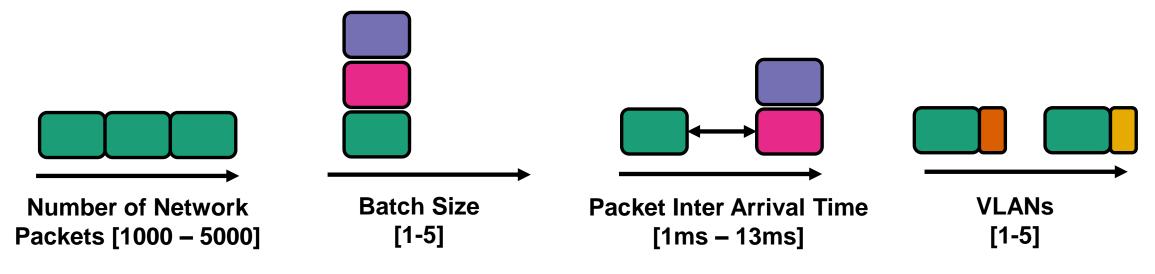
Number of Network Packets [1000 – 5000]

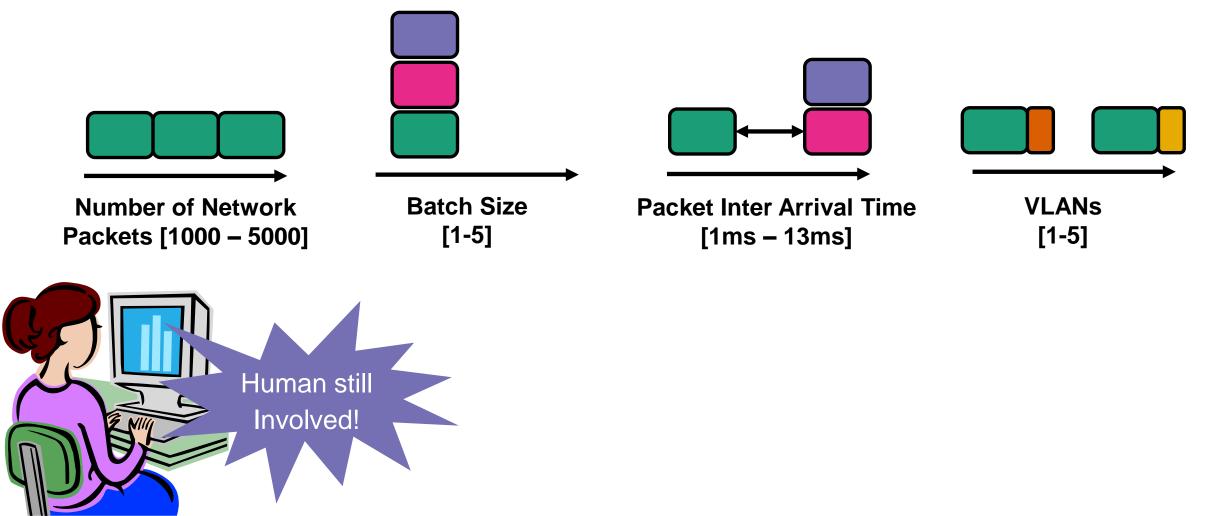


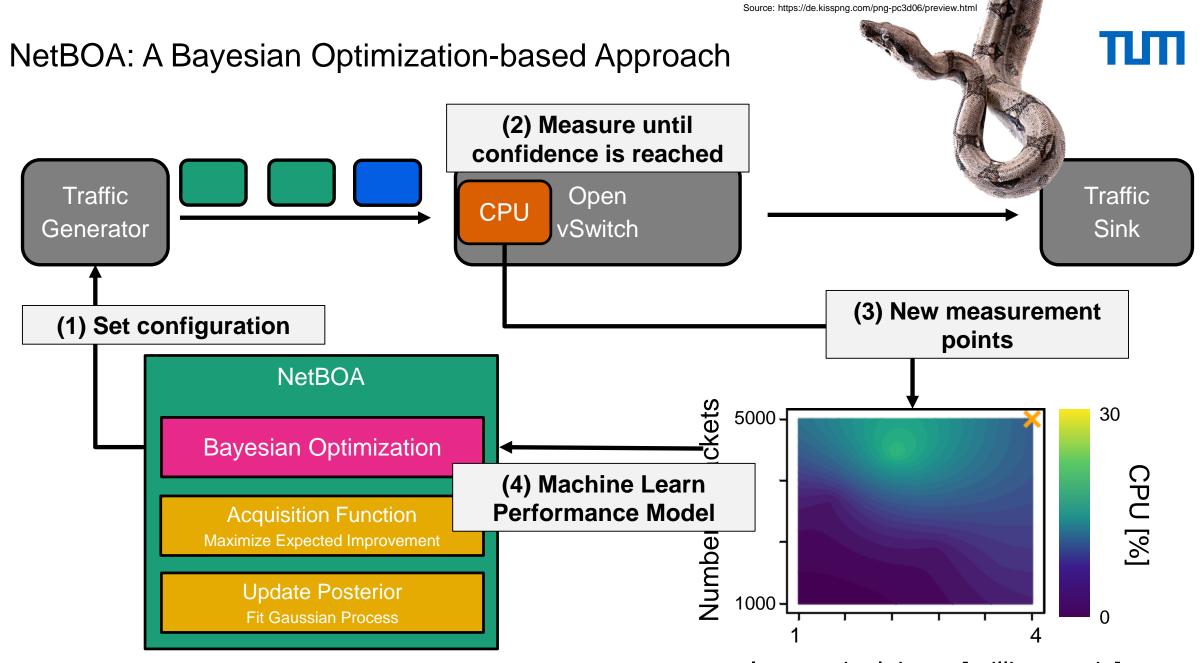


ТШ

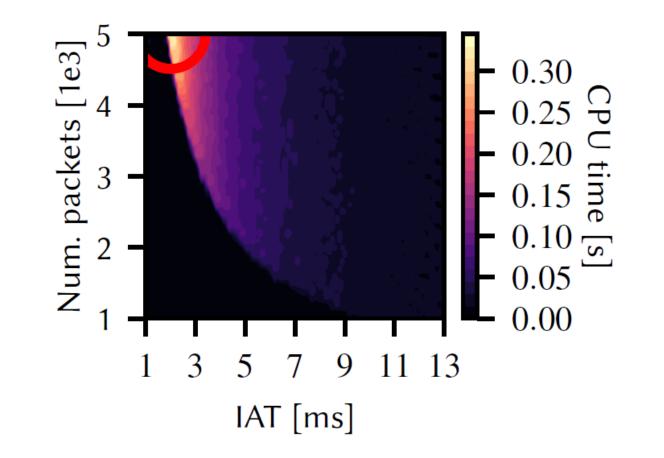


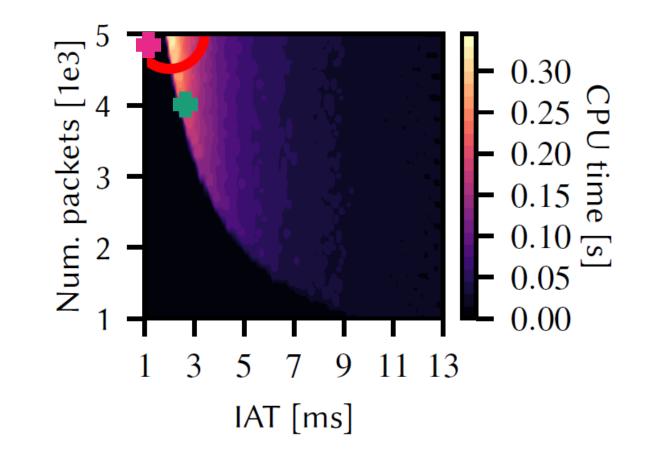






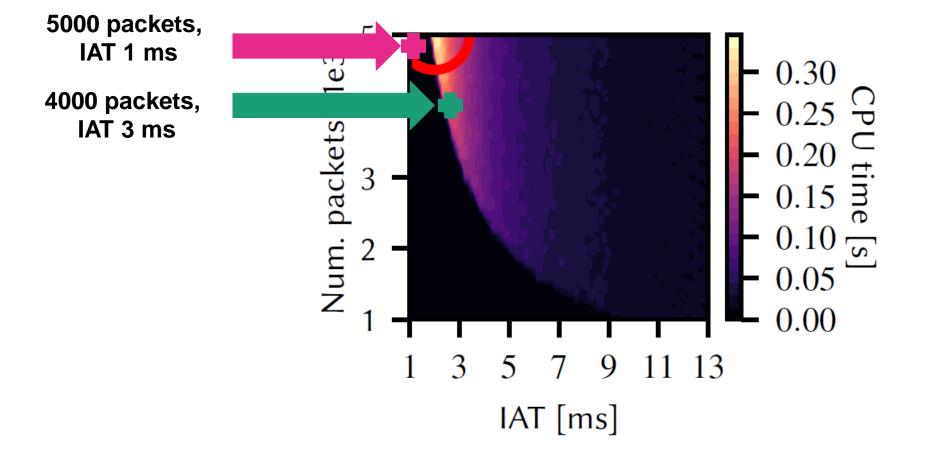
Inter arrival times [milliseconds]



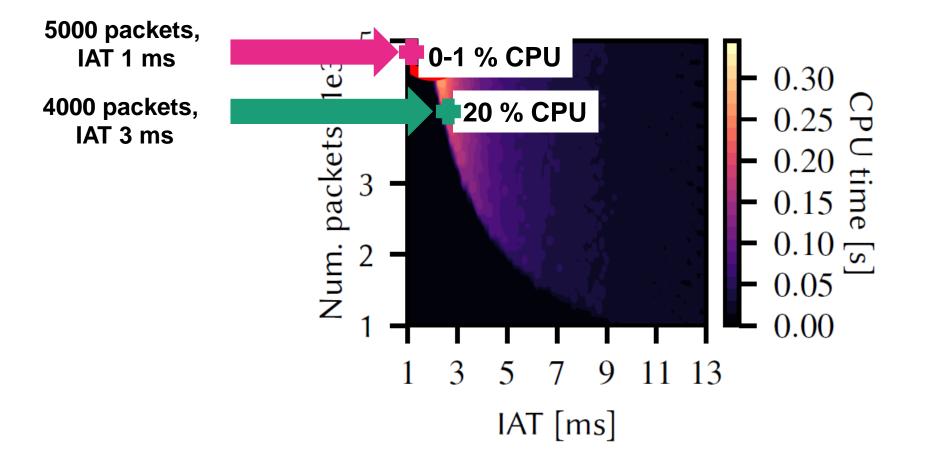


Grid Search for Two Parameters (Num. Packets and Inter Arrival Time)

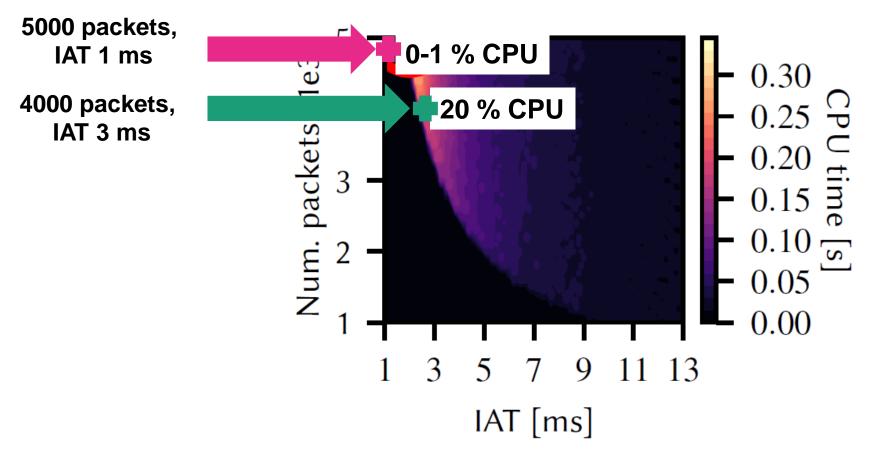




пп



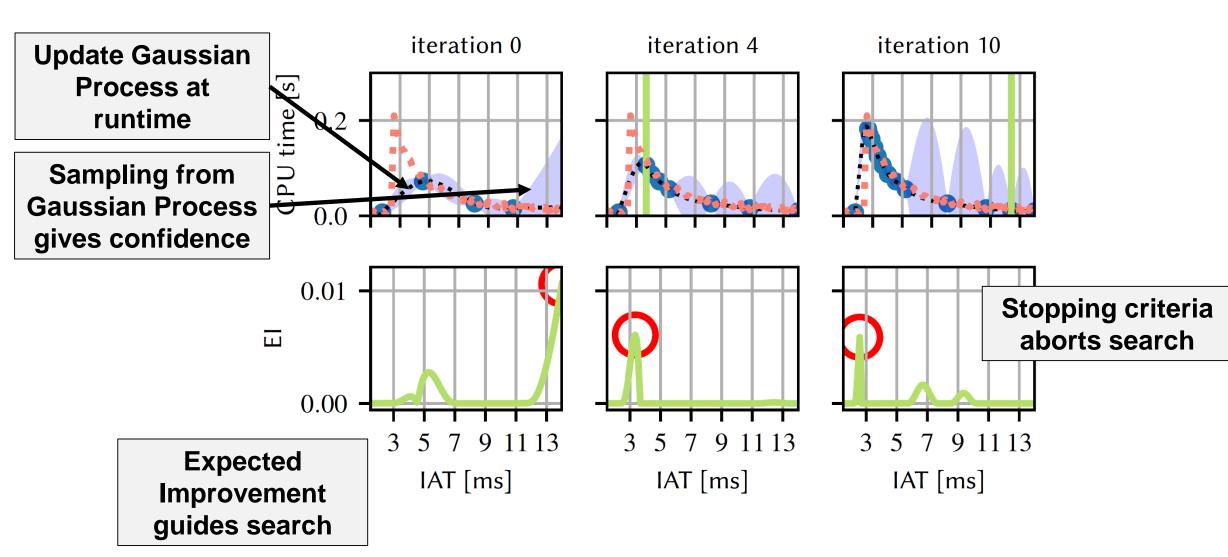


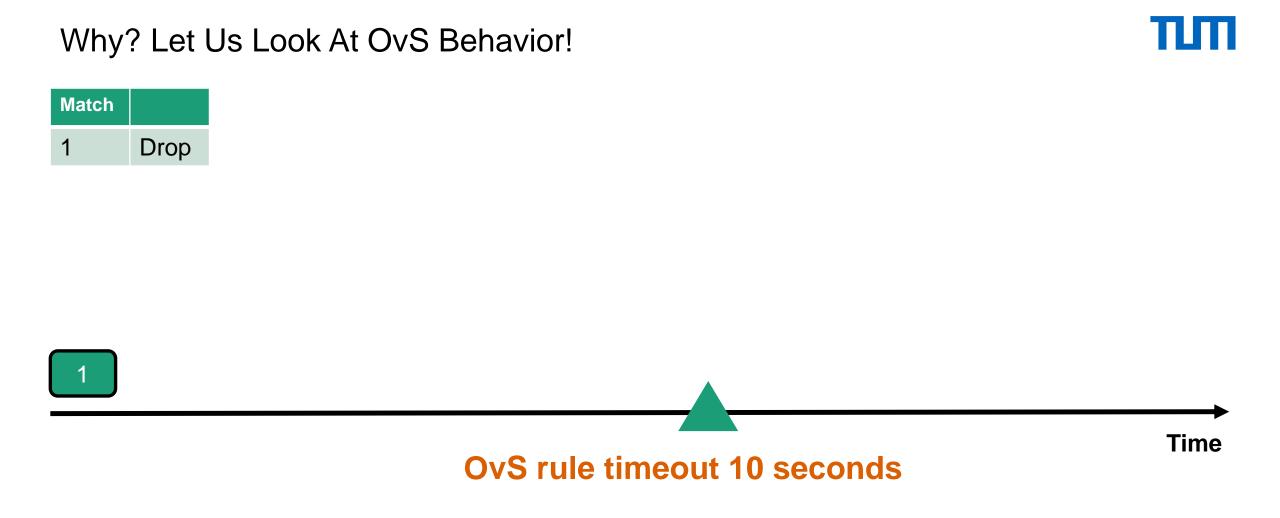


- Performance models are non-trivial
- Surprising: Sending less network packets over time can lead to significantly higher CPU
- But: Can we find such weak-spots automatically?

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



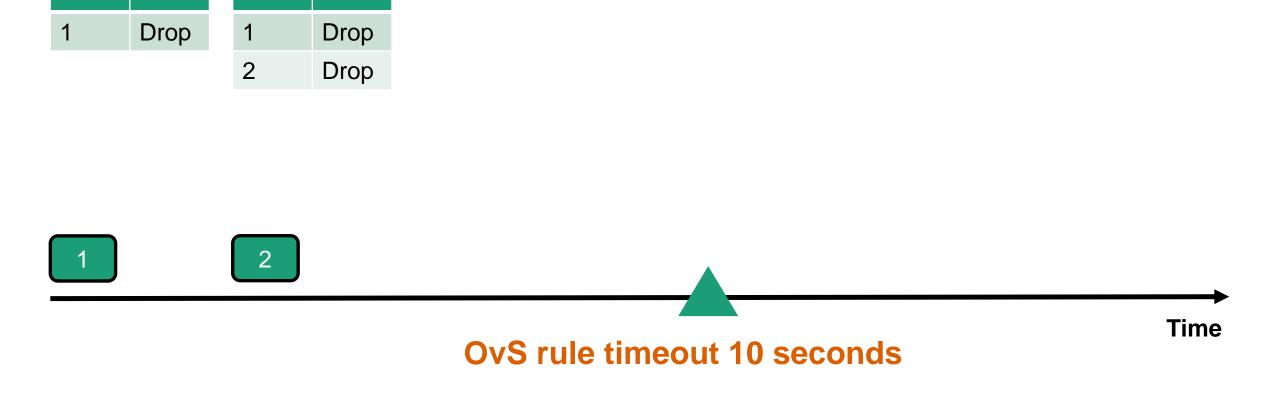




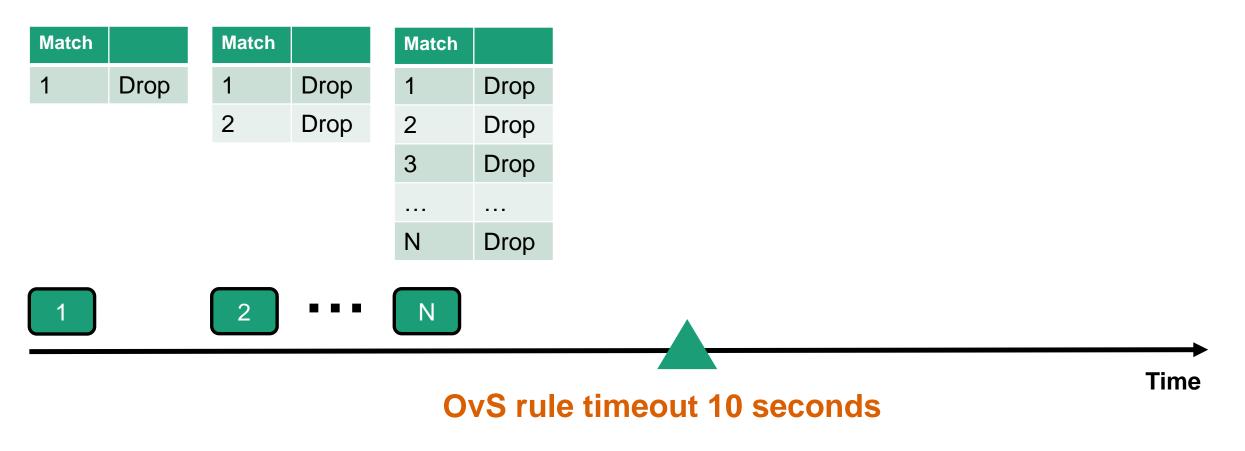
Match

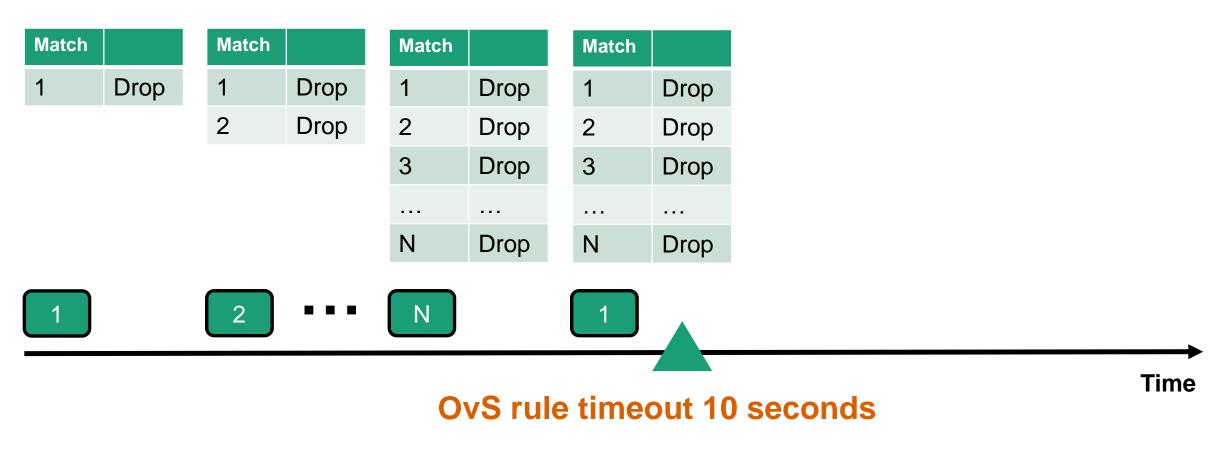
Match



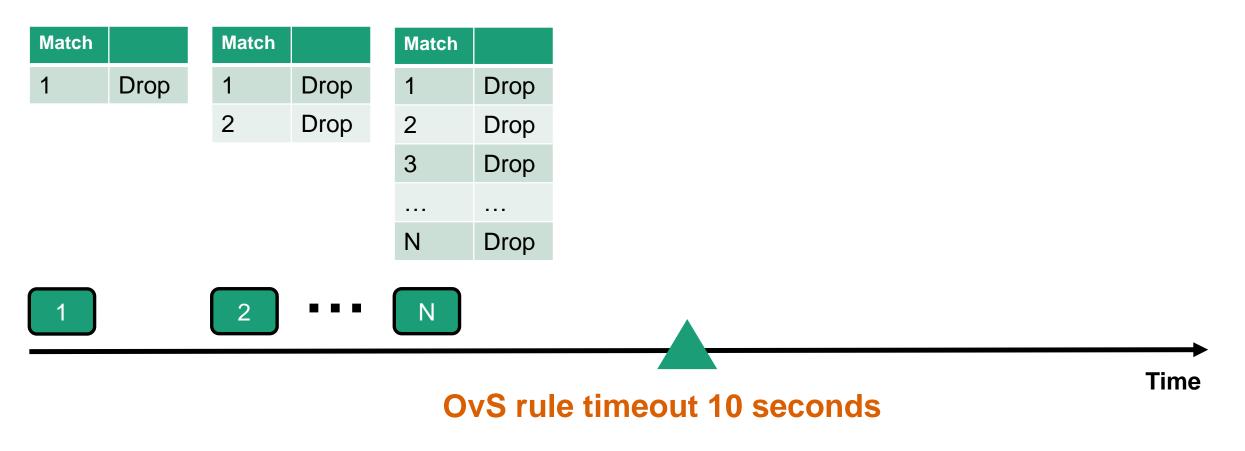


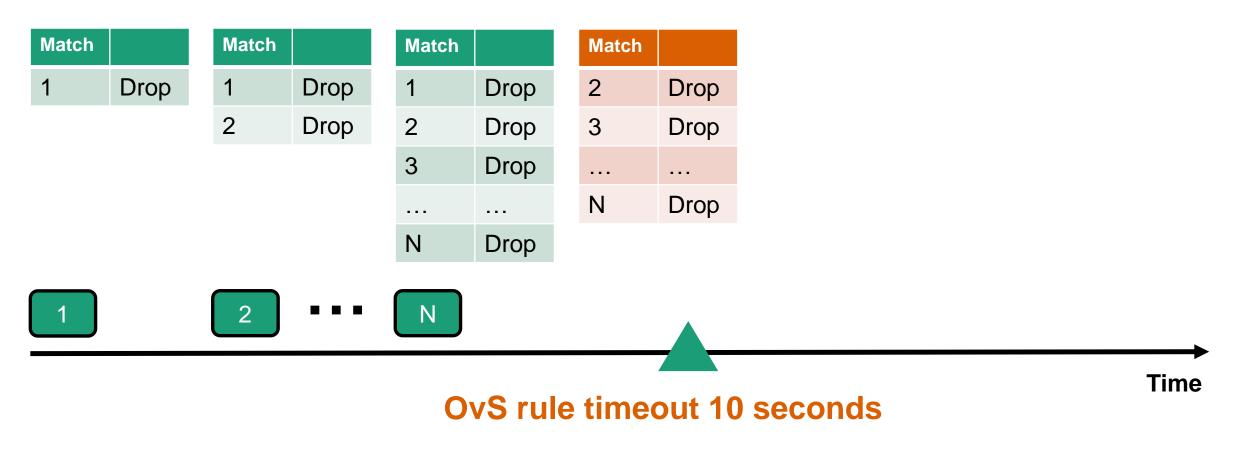


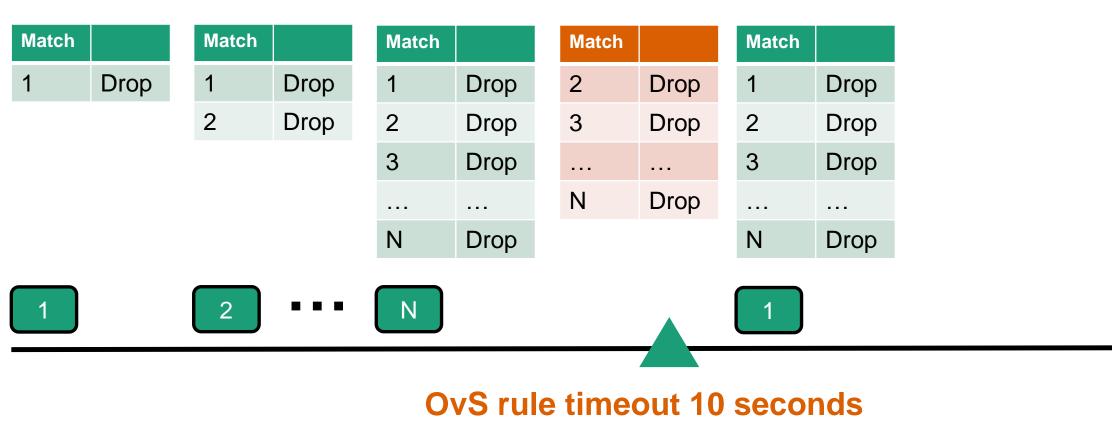








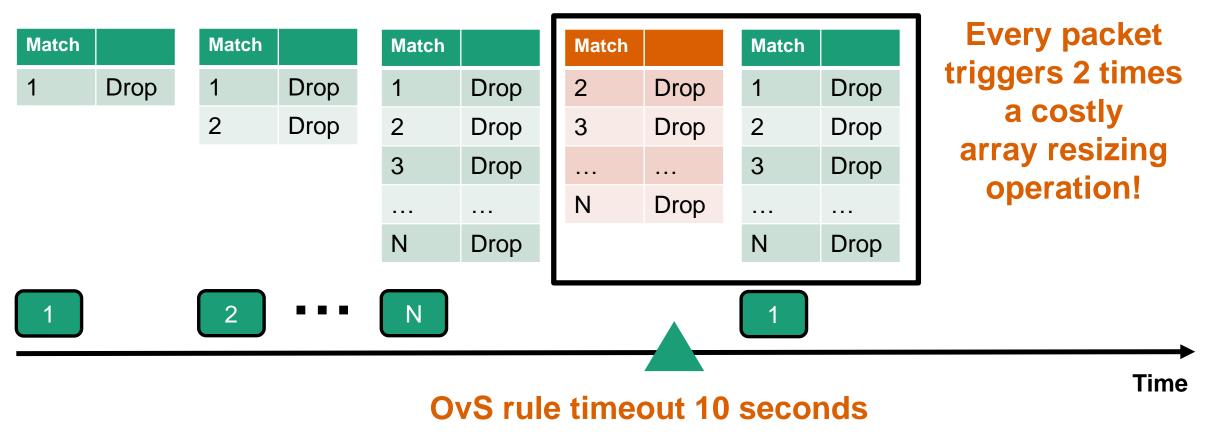




• We are using the OvS switch with the Megaflow Cache enabled

Time

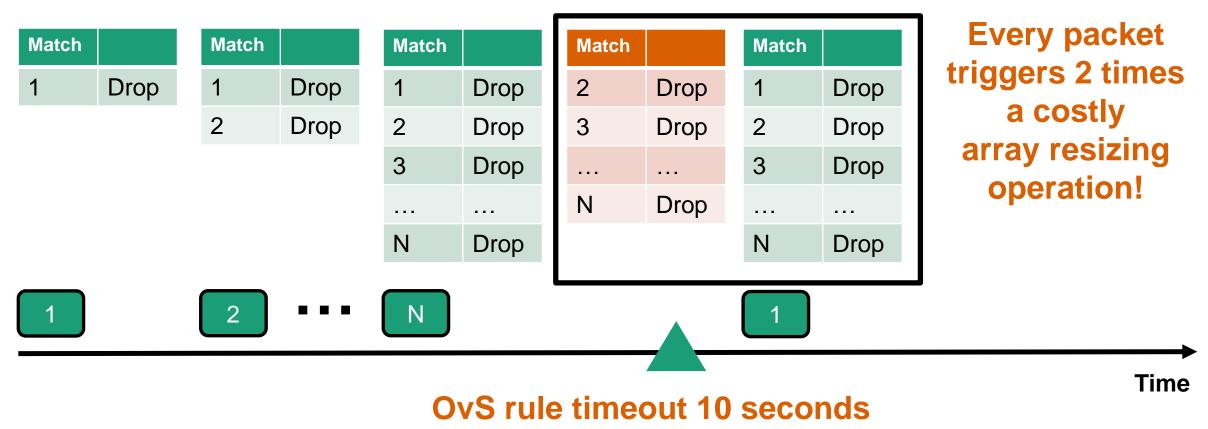
пп



[•] We are using the OvS switch with the **Megaflow Cache enabled**

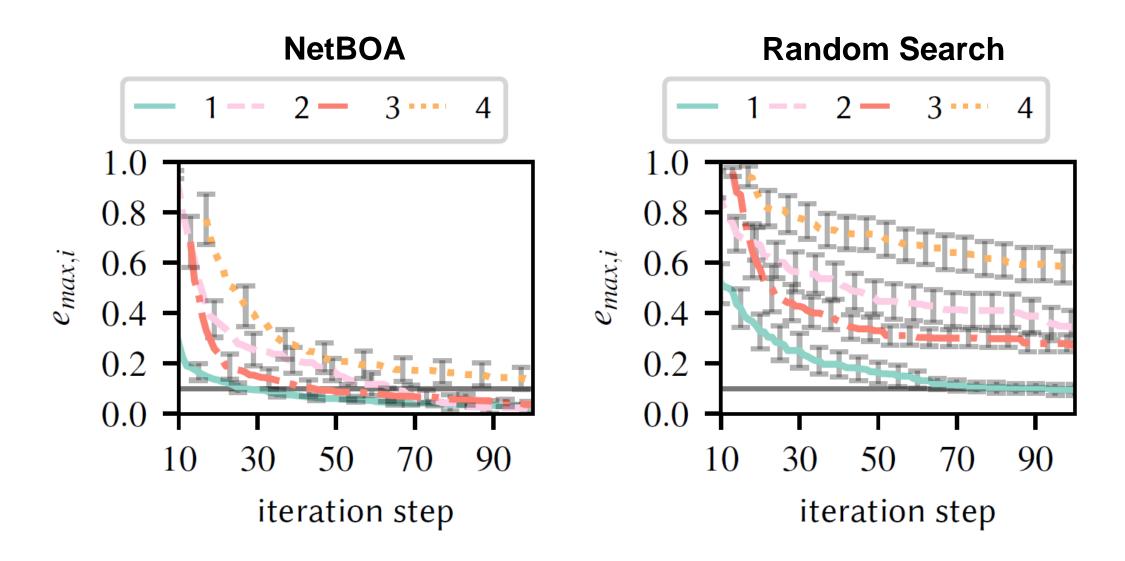
Why? Let Us Look At OvS Behavior!

ПП

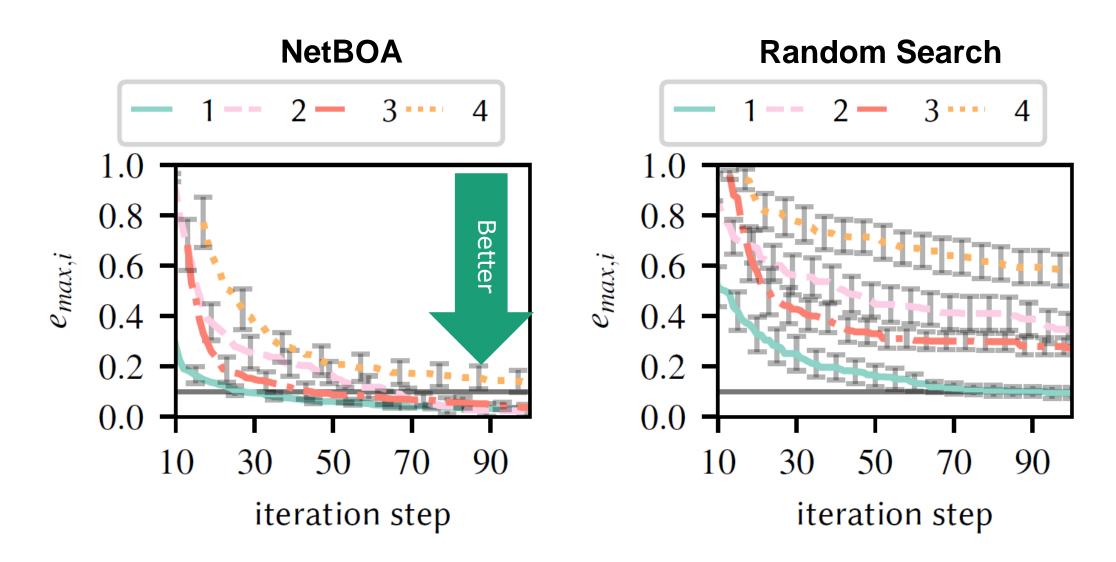


- We are using the OvS switch with the **Megaflow Cache enabled**
- For instance for 5000 packets: We trigger roughly every >2 ms a flow insertion + removal
- \rightarrow Forcing OvS to continuously run through the array + resizing it



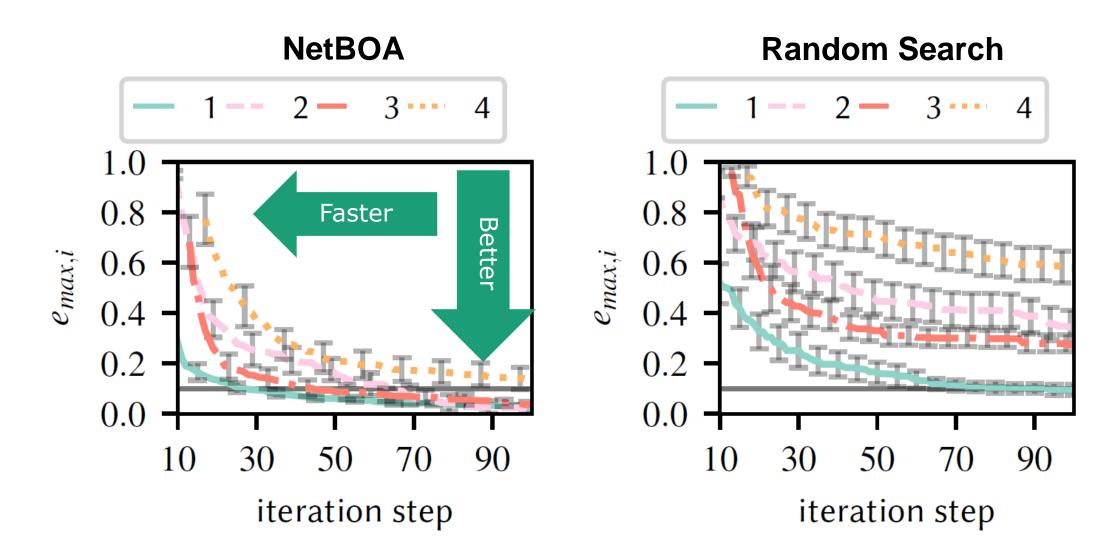




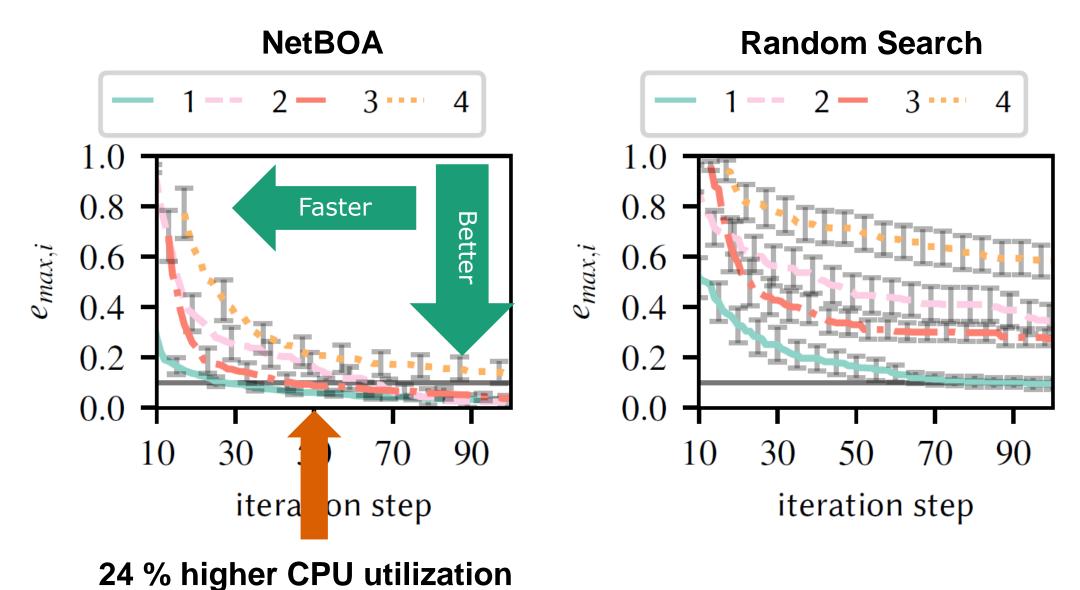


NetBOA vs Random Search









Adversarial input generation to find weak spots, security holes ... to make your systems bullet-proof?

- Adversarial input generation to find weak spots, security holes ... to make your systems bullet-proof?
- Use case: NetBOA is a Bayesian Optimization-based data-driven approach to generate network traffic configurations for benchmarking network function implementations

- Adversarial input generation to find weak spots, security holes ... to make your systems bullet-proof?
- 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

- Adversarial input generation to find weak spots, security holes ... to make your systems bullet-proof?
- 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!



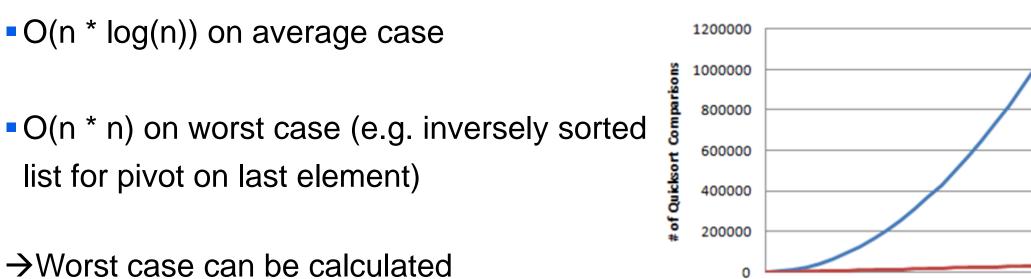
(2) Benchmarking Data Center Traffic Scheduling Algorithms

Motivation: Automation Helps Finding Weak-Spots



Special Input

Random Input



Quick-Sort

0

500

1000

Input Sequence Length

1500

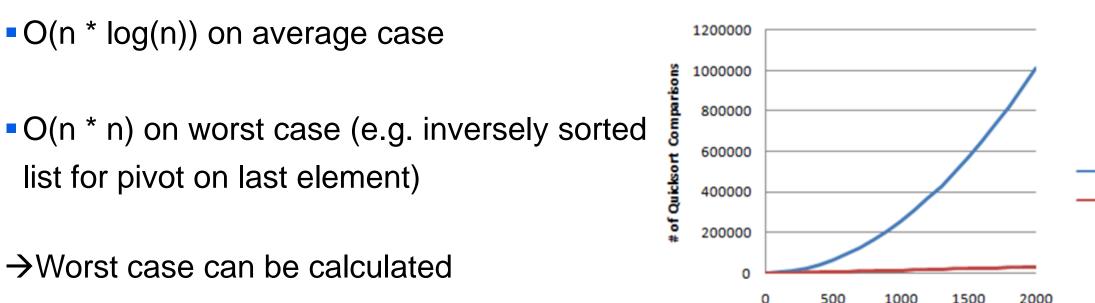
2000

Motivation: Automation Helps Finding Weak-Spots



Special Input

Random Input



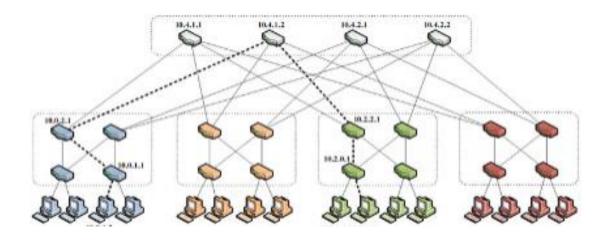
Quick-Sort

Input Sequence Length

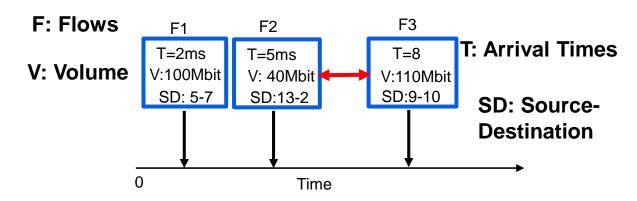
Question: How to apply automation to data center traffic?

Data Center Scenario





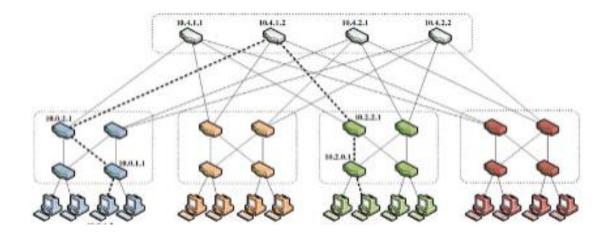
What can be changed?



- Small flows are scheduled first
- Shortest-Path-Routing
- K=4 Fattree

Data Center Scenario



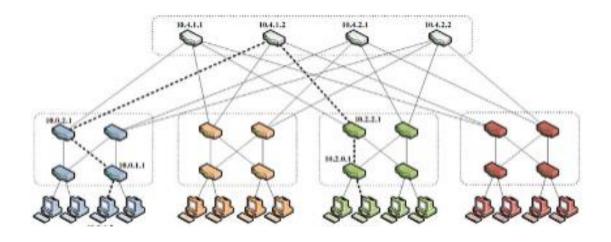


- Small flows are scheduled first
- Shortest-Path-Routing
- K=4 Fattree

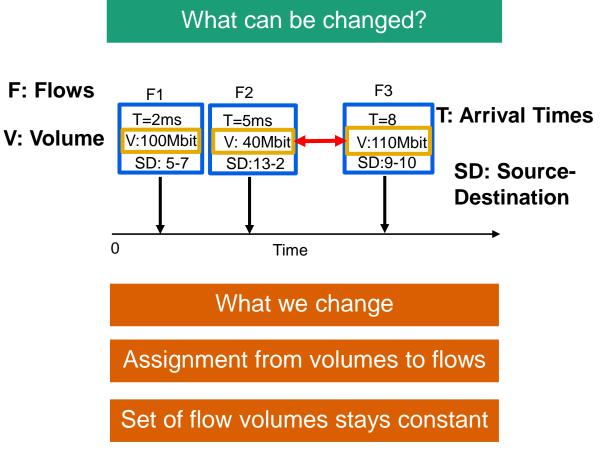
What can be changed? F: Flows F3 F2 F1 T: Arrival Times T=2ms T=5ms T=8 V: Volume V:100Mbit V: 40Mbit V:110Mbit SD: 5-7 SD:13-2 SD:9-10 SD: Source-**Destination** 0 Time What we change Assignment from volumes to flows Set of flow volumes stays constant

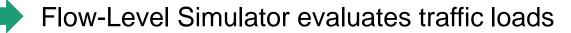
Data Center Scenario





- Small flows are scheduled first
- Shortest-Path-Routing
- K=4 Fattree





Problem Definition

ТШ

Given Set of Flows:

	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

Find the order of Volumes such that:

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

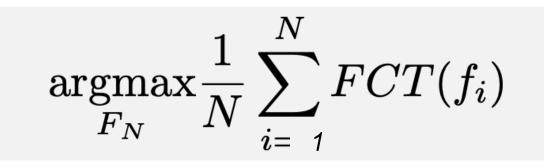
Problem Definition

ПШ

Given Set of Flows:

	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

Find the order of Volumes such that:



FCT: Flow Completion Time



ТШП

12ms	14ms	17ms	18ms	21ms	24ms
3	4	13	2	3	12
14	12	7	7	1	6
10Mbit	400Mbit	90Mbit	200MBit	9Mbit	110Mbit



Pseudo Code

1. Sample N flows

12ms	14ms	17ms	18ms	21ms	24ms
3	4	13	2	3	12
14	12	7	7	1	6
10Mbit	400Mbit	90Mbit	200MBit	9Mbit	110Mbit



Pseudo Code

1. Sample N flows

х	x	x	x	x	x
x	x	х	x	x	x
x	x	x	x	x	x
10Mbit	400Mbit	90Mbit	200MBit	9Mbit	110Mbit



- 1. Sample N flows
- 2. Extract the sequence of volumes V

x	x	х	х	х	х
x	x	x	x	x	x
х	x	x	x	x	x
10Mbit	400Mbit	90Mbit	200MBit	9Mbit	110Mbit



- 1. Sample N flows
- 2. Extract the sequence of volumes V
- 3. Generate permutations of V by changing is order (=initial population)

x	x	х	х	х	х
x	x	x	x	x	x
x	x	x	x	x	x
10Mbit	400Mbit	90Mbit	200MBit	9Mbit	110Mbit

тлп

- 1. Sample N flows
- 2. Extract the sequence of volumes V
- 3. Generate permutations of V by changing is order (=initial population)
- 4. Repeat (until convergence)

x	x	х	х	х	х
x	x	x	x	x	x
x	x	x	x	x	x
10Mbit	400Mbit	90Mbit	200MBit	9Mbit	110Mbit

Pseudo Code

- 1. Sample N flows
- 2. Extract the sequence of volumes V
- 3. Generate permutations of V by changing is order (=initial population)
- 4. Repeat (until convergence)

Crossover

x	х	x	х	х	х
x	x	x	x	x	x
x	x	x	x	x	x
10Mbit	400Mbit	90Mbit	200MBit	9Mbit	110Mbit

Pseudo Code

- 1. Sample N flows
- 2. Extract the sequence of volumes V
- 3. Generate permutations of V by changing is order (=initial population)
- 4. Repeat (until convergence)

Crossover

Mutation

x	х	x	х	х	х
x	x	x	x	x	x
x	x	x	x	x	x
10Mbit	400Mbit	90Mbit	200MBit	9Mbit	110Mbit

Pseudo Code

- 1. Sample N flows
- 2. Extract the sequence of volumes V
- 3. Generate permutations of V by changing is order (=initial population)
- 4. Repeat (until convergence)

Crossover

Mutation

Fitness

x	x	x	x	x	x
x	x	x	x	x	x
x	x	x	x	x	x
10Mbit	400Mbit	90Mbit	200MBit	9Mbit	110Mbit

Pseudo Code

- 1. Sample N flows
- 2. Extract the sequence of volumes V
- 3. Generate permutations of V by changing is order (=initial population)
- 4. Repeat (until convergence)

Crossover

Mutation

Fitness

Selection

x	x	x	x	x	x
x	x	x	x	x	x
x	x	x	x	x	x
10Mbit	400Mbit	90Mbit	200MBit	9Mbit	110Mbit

Fitness Function



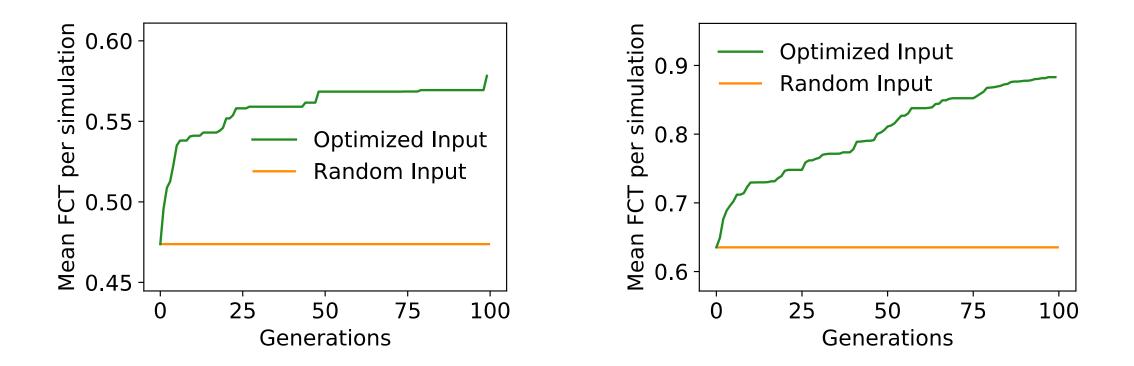
Data Center Simulator

Simulator:

- 500 Mbit Links
- Uniform Volume between 1 and 500 Mbit
- Poisson Arrival Times (mean 0.7 sec)
- Uniform Src-Dst pairs

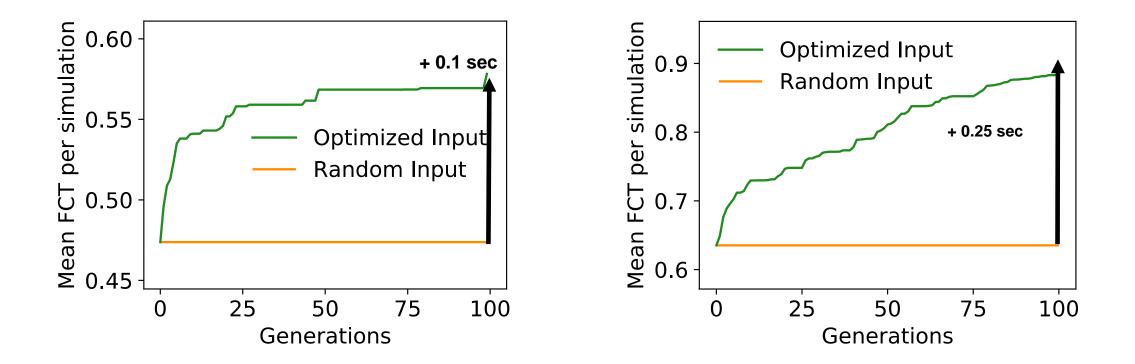
Population with N = 10 flows

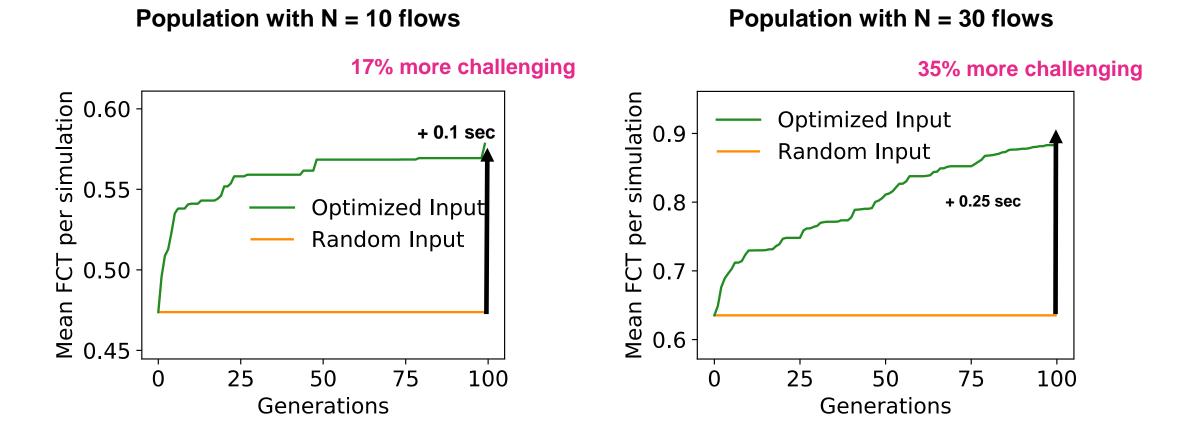
Population with N = 30 flows

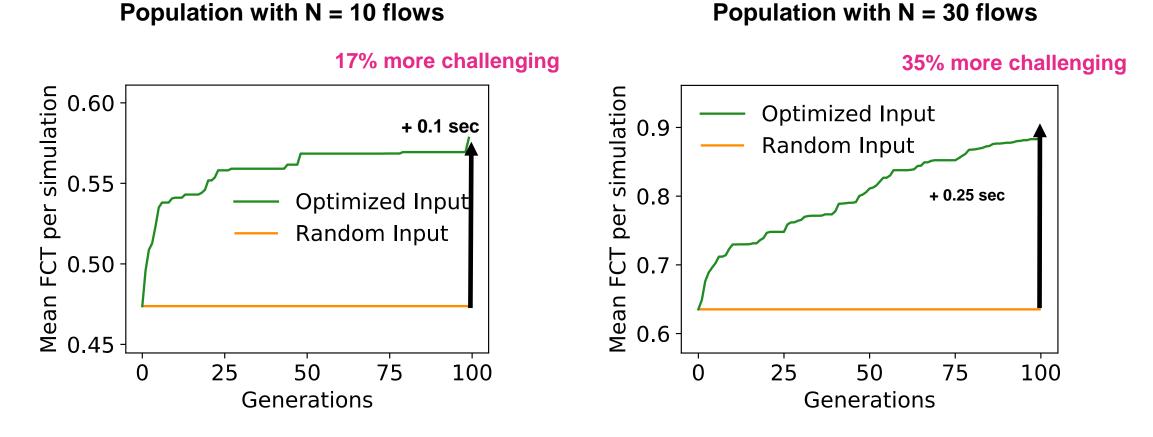


Population with N = 10 flows

Population with N = 30 flows

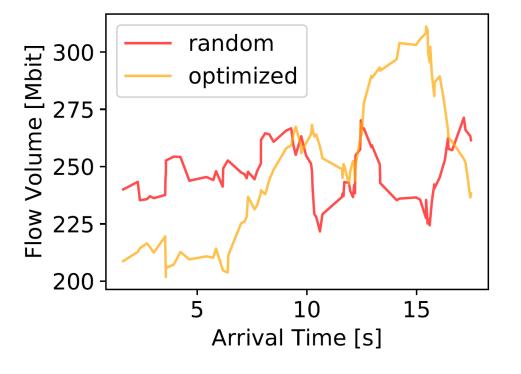


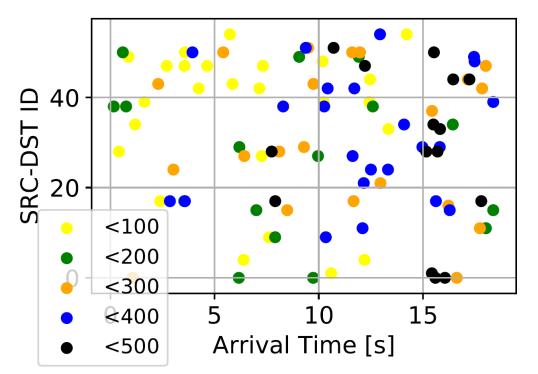




More Flows Higher margin of optimization

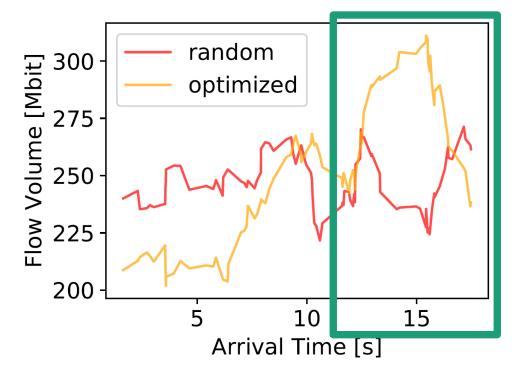


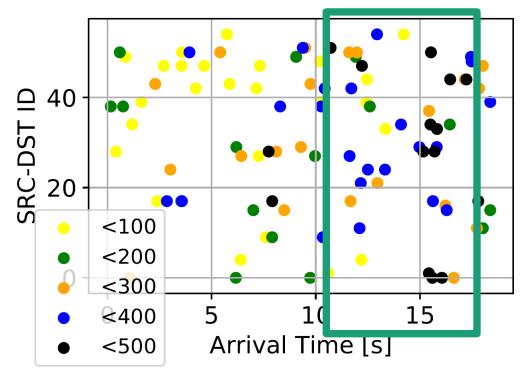




- Concentrate larger flows together
- Place large flows on the same link for close arrivals

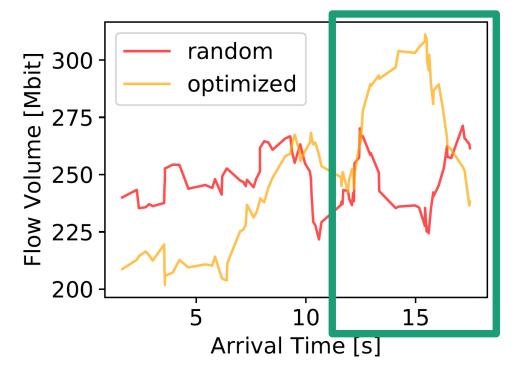


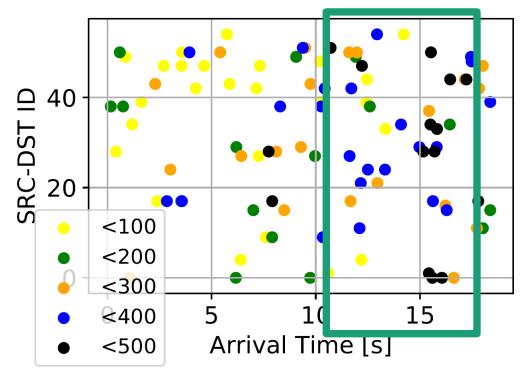




- Concentrate larger flows together
- Place large flows on the same link for close arrivals



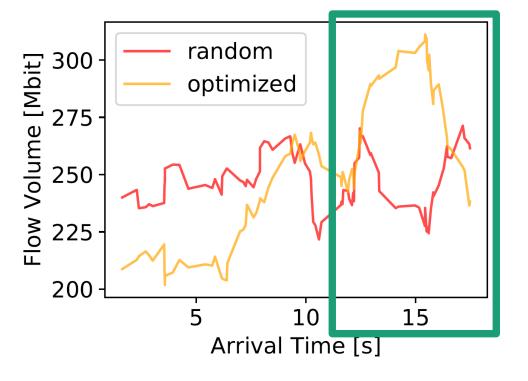


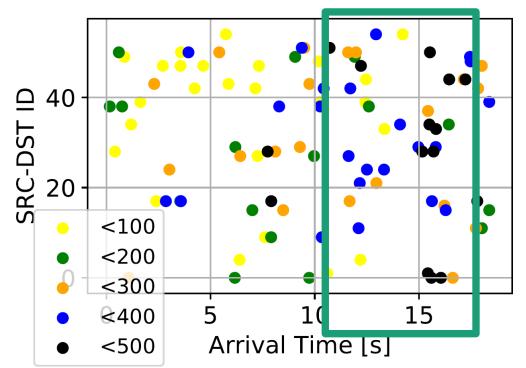


- Concentrate larger flows together
- Place large flows on the same link for close arrivals

BUT: Simulations consume a lot of time!







- Concentrate larger flows together
- Place large flows on the same link for close arrivals

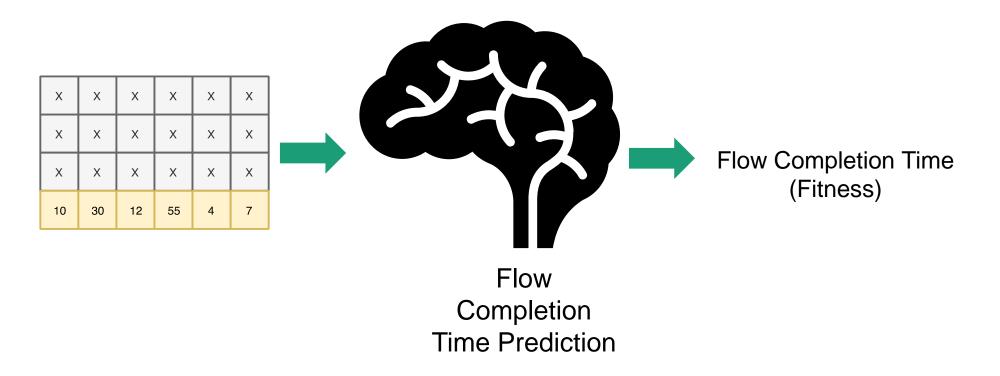
BUT: Simulations consume a lot of time!

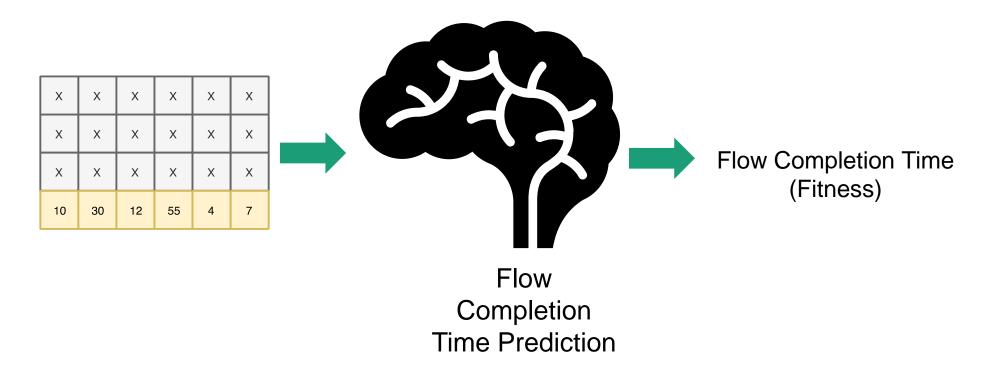
Idea: Use Machine Learning in Genetic Algorithm [Bha13]



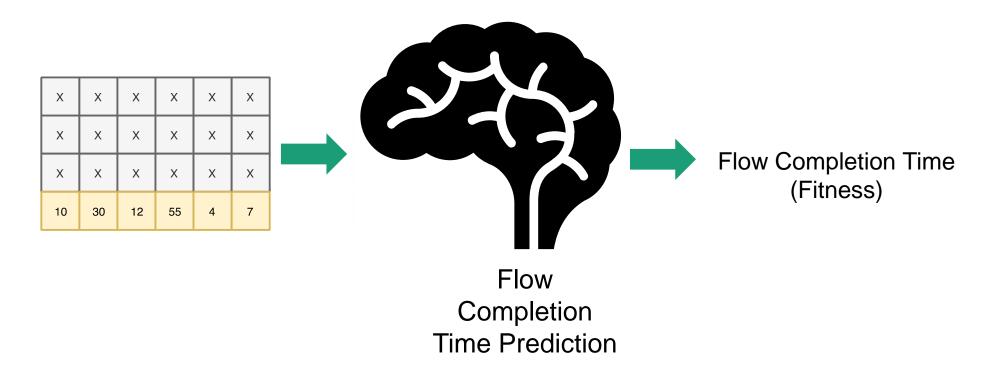


Data Center Simulation





- \rightarrow Needs to be evaluated very frequently
- \rightarrow slow, does not scale

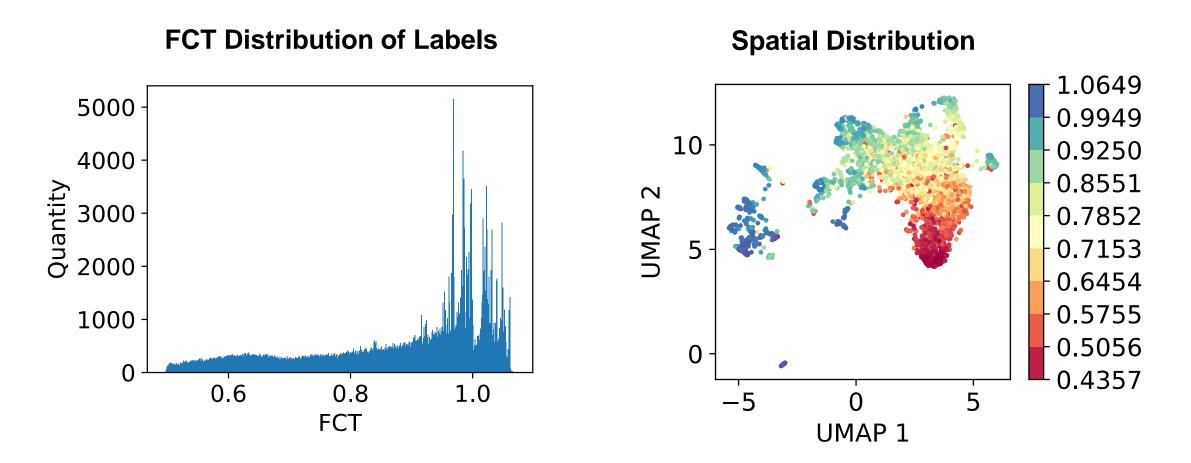


- \rightarrow Needs to be evaluated very frequently
- \rightarrow slow, does not scale

Approximate Fitness Function with Deep Neural Network

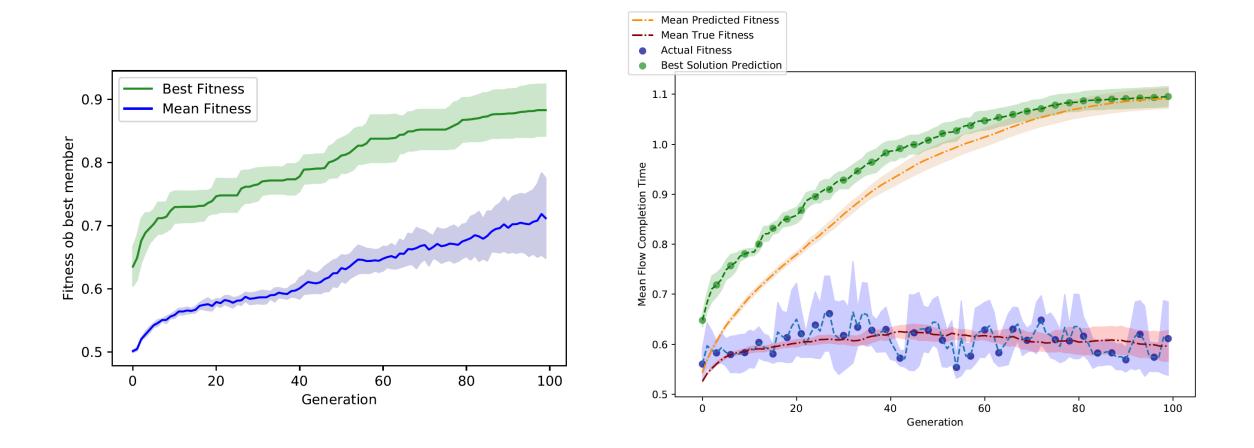
The Training Data





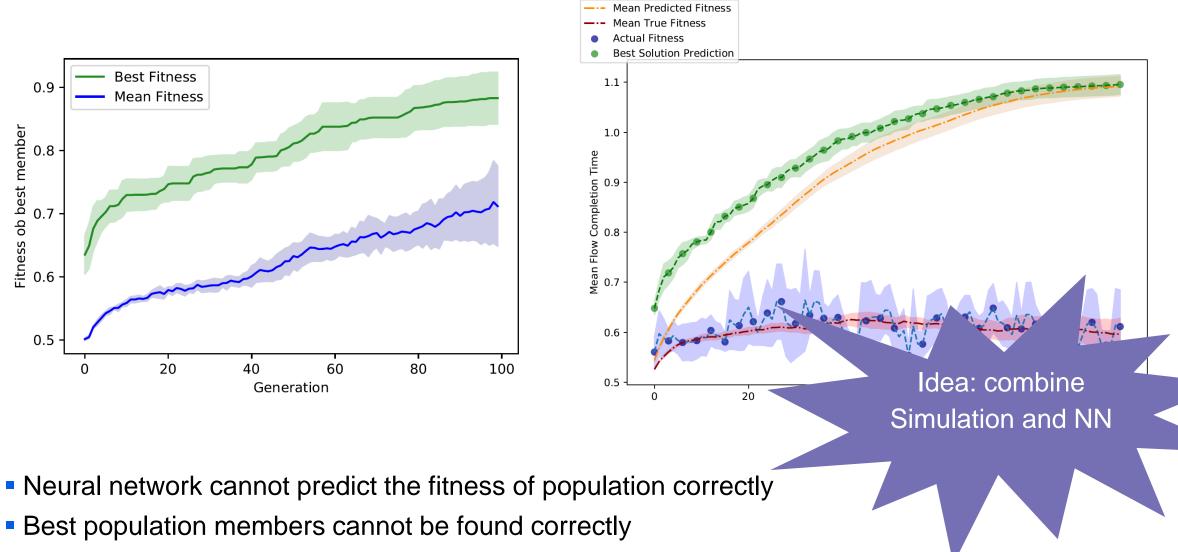
Test Set Score: 87% of the samples achieved a relative error of less then 5%

Comparison of Simulation vs Neural Network

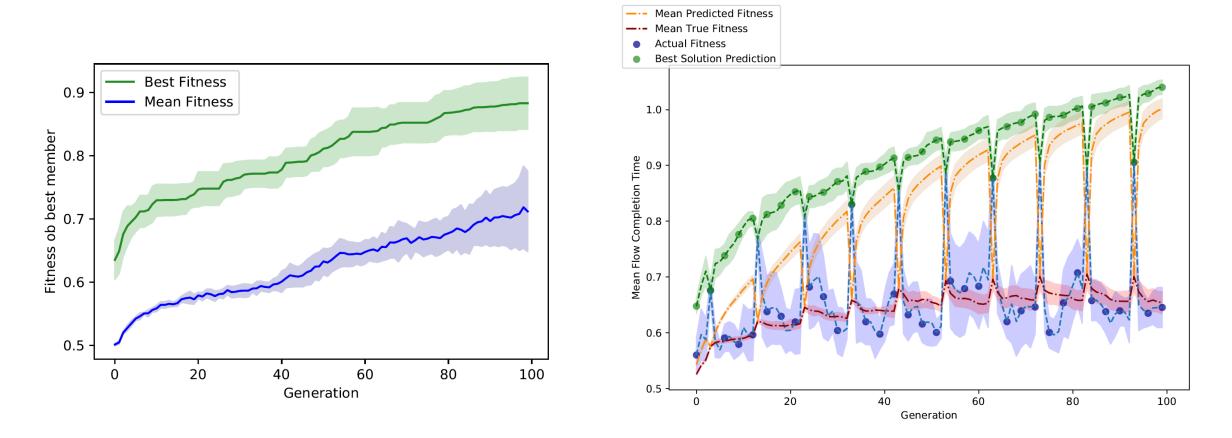


- Neural network cannot predict the fitness of population correctly
- Best population members cannot be found correctly

Comparison of Simulation vs Neural Network



Comparison of Simulation vs Simulation-enhanced Neural Network Approach



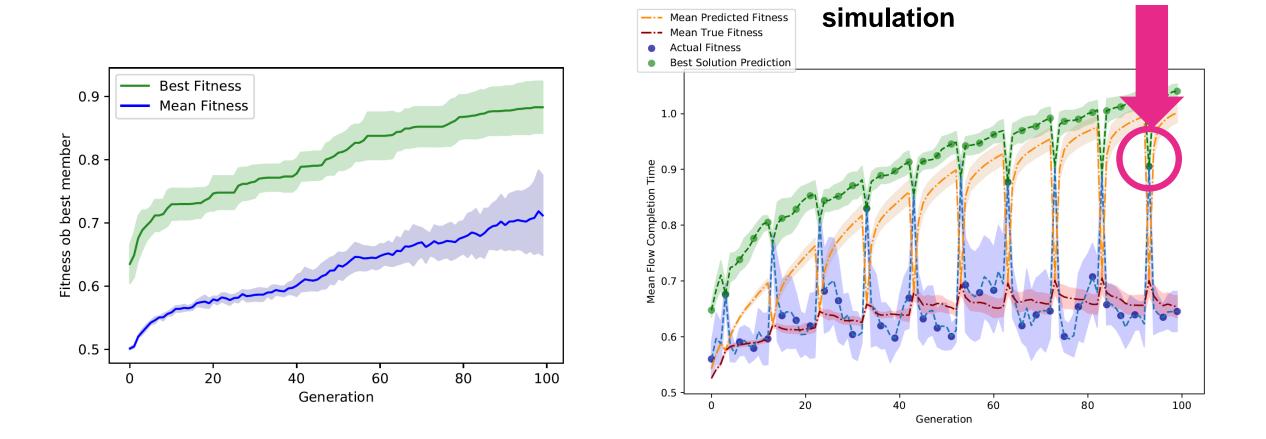
Simulations can be used to determine current best simulation members

More than one simulation needed to improve population

Comparison of Simulation vs Simulation-enhanced Neural Network

ПШ

Performance even better than



Simulations can be used to determine current best simulation members

More than one simulation needed to improve population

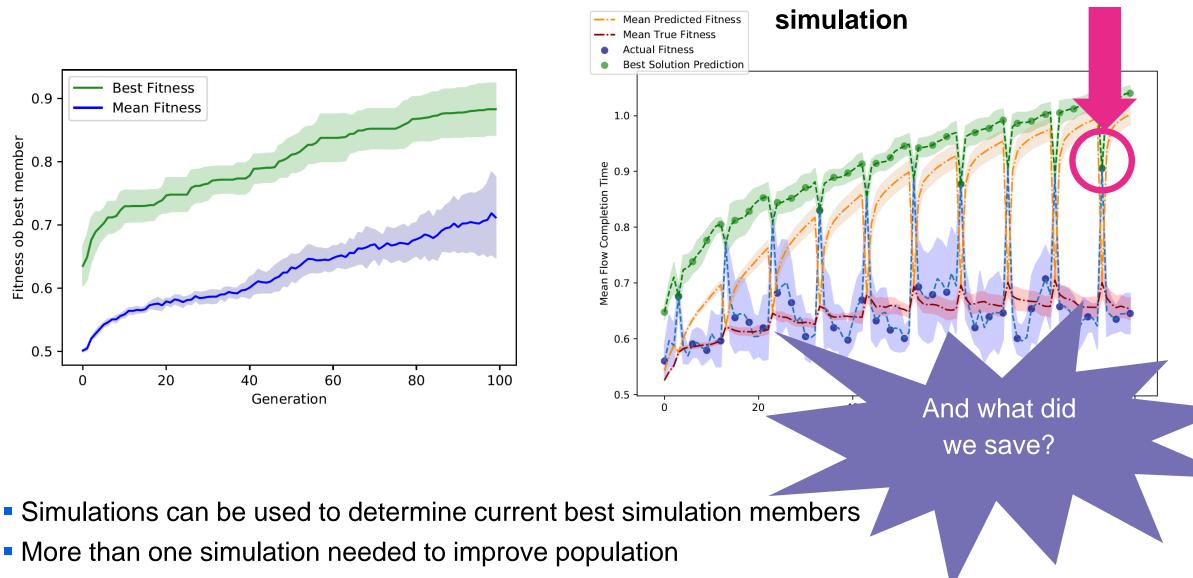
Approach

Comparison of Simulation vs Simulation-enhanced Neural Network

Approach

ПШ

Performance even better than





Mixed 4 times slower than NeuralNet

Population	10	30	50	100
Simulation	106 (7.1)	369 (7.7)	638 (24.2)	1187 (17.9)
NeuralNet	23 (1.8)	29 (1.5)	32 (1.6)	43 (1.1)
Mixed	32 (0.9)	73 (1.0)	118 (2.6)	210 (3.1)

... but Mixed 4 times faster than Simulation

-

Neural Net < Mixed < Simulation

Part 2: Conclusion

- Genetic Algorithm can automate adversary Traffic Generation
- →Automated Benchmarking
- Neural Network can significantly accelerate Genetic Algorithms

→Scalability

Limitations:

- Long training time of GA
- Accelerator trades-off solution quality and compute time
- Accelerator needs to be re-trained when fitness function changes

Potentials and Future Work:

- Utilize current network state (e.g., demand matrix)
- Make a prediction for the next arrival(s) e.g., investigate existing network traces



Thank you!

Questions?

[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

[NetAl'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 (NetAl '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

[Bha13] Maumita Bhattacharya. 2013. Evolutionary Approaches to Expensive Optimisation. Arxiv - Computers & Society 2, 3 (2013), 53–59. DOI:http://dx.doi.org/10. 14569/IJARAI.2013.020308