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

150 Jahre culture of excellence

Adaptation and Flexibility in Softwarized Networks

Wolfgang Kellerer

Technical University of Munich (TUM), Germany

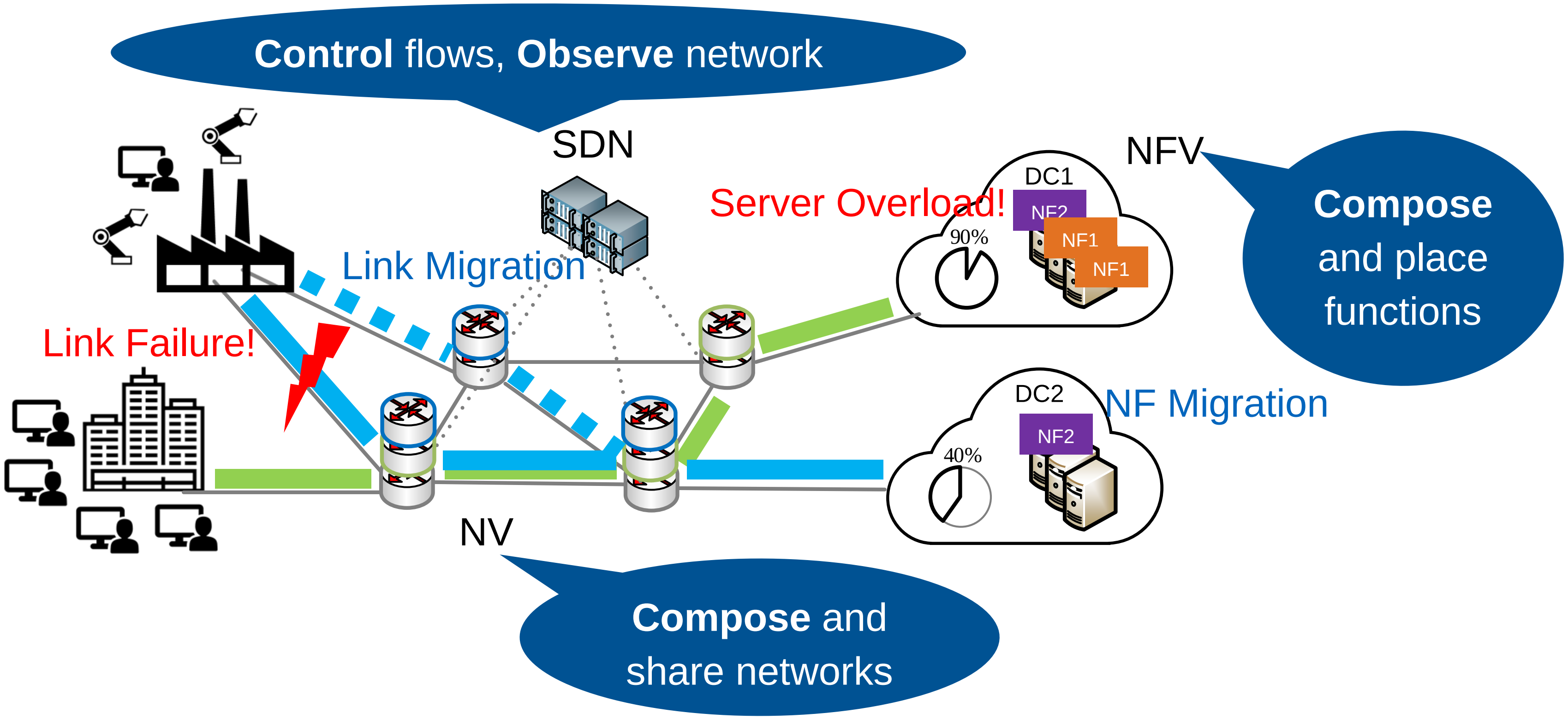
with Dr. Peter Babarzci, Dr. Andreas Blenk, Dr. Arsany Basta, Mu He, Patrick Kalmbach, Dr. Markus Klügel, Alberto Martinez Alba, Johannes Zerwas

ETHZ, Zurich, April 16, 2019

www.networkflexibility.org

This work is part of a project that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program grant agreement No 647158 – **FlexNets (2015 – 2020)**.

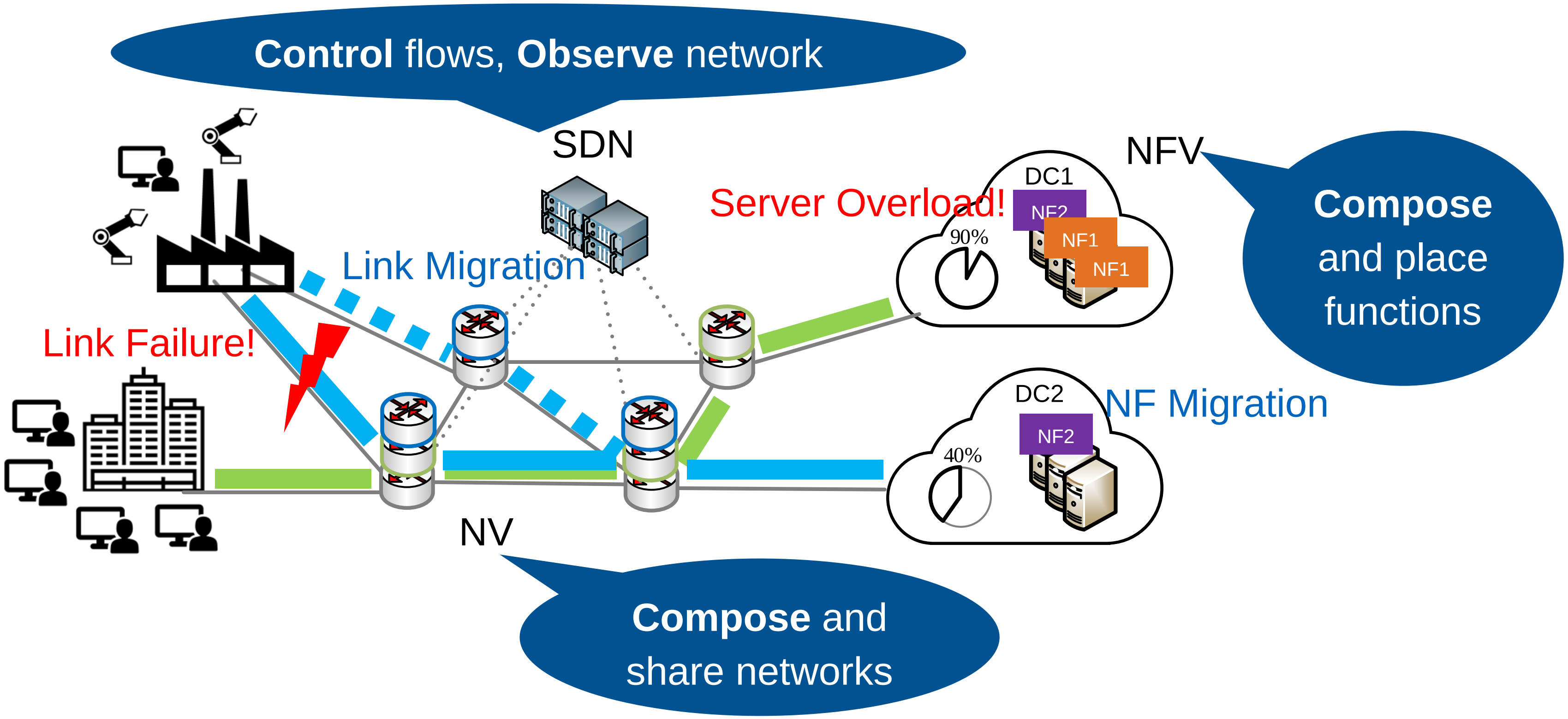
Softwarized Networks: flexible network adaptation



Network Virtualization (NV)
Network Function Virtualization (NFV)
Software Defined Networking (SDN)

} enablers for **flexible adaptation**

Softwarized Networks: flexible network adaptation



Adaptation: (1) Detection – (2) Decision – (3) Execution

Network Observation (SDN, NFV)

Composition (NFV, NV)

Control (SDN, NV)

W. Kellerer, et al. Adaptable and Data-Driven Softwarized Networks: Review, Opportunities and Challenges. Proc. of the IEEE, 2019.

Why is flexibility so important?

- Evolution tells us that the more flexible species can better survive

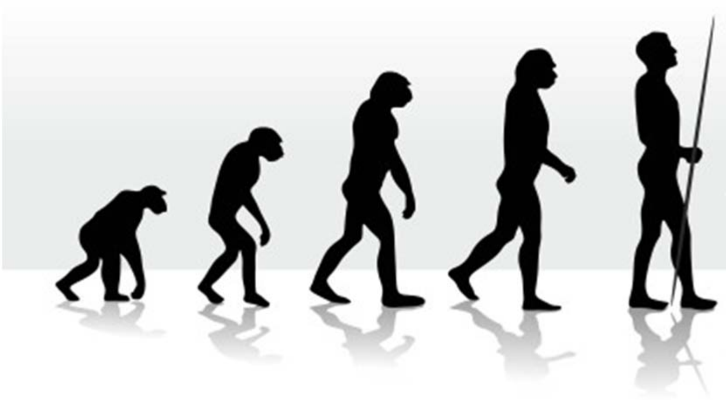
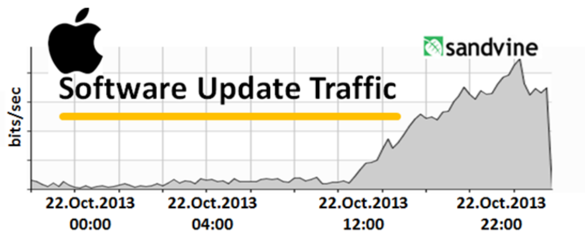


Image source: http://www.paleoplan.com

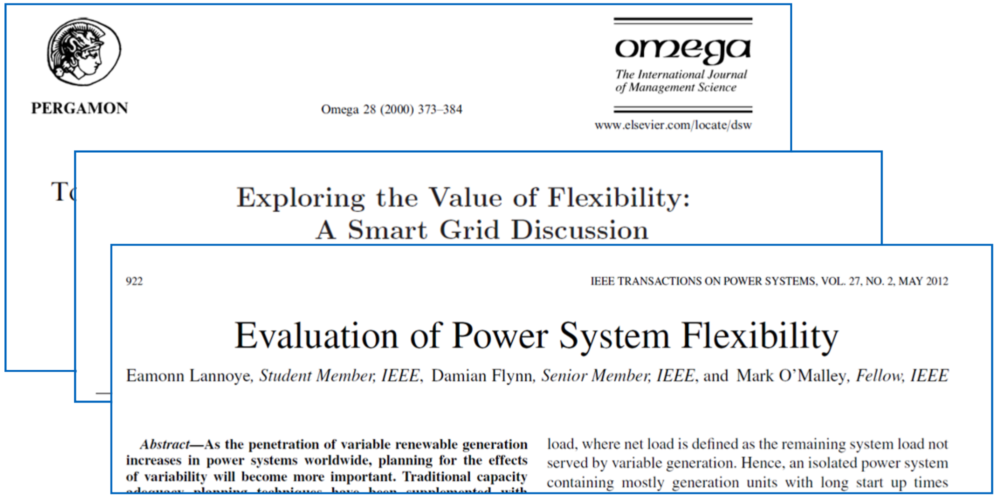
- *What about networks? Will they survive?*
- Less <u>explicitly</u> addressed: ***flexibility*** *to* ***adapt*** *to future demands*
- **Considering the Future** is <u>very</u> important for survival
  - enables operators to cover the future
  - key decision factor between network designs
  - optimize networks for flexibility



Are we there already?

- Are we <u>100% flexible</u> already (e.g. with Softwarized Networks)?
- How <u>far</u> can we go? What is the <u>optimal network design</u> for flexibility?

- What is ***network flexibility*** ?



We need

- a **fundamental understanding** of how to provide flexibility
- a **quantitative measure** for flexibility pro and contra certain designs

M. He, A. Martinez Alba, A. Basta, A. Blenk, W. Kellerer. *Flexibility in Softwarized Networks: Classifications and Research Challenges*. **IEEE Communication Surveys & Tutorials**, 2019.

An exercise on measuring flexibility

vs.

Source: Magazin.com

Fixed-set tool **Re-configurable tool box**

- Which tool is more flexible?
  - re-configuration shows more potential to be **more flexible**

- When can both exihbit the same flexibility?
  - **no need to change** (no challenge) → probability of requests make a difference
  - both cannot satsify my requests → **infeasible**

- When can the re-configurable tool be less flexible?
  - **adaptation time** → re-configurable object might not be handy
  - **cost** → inefficient

Screwdriver

Our approach for *Network Flexibility*

Network **flexibility** = ability to support *adaptation requests (challenges)* (e.g., new requirements or traffic patterns) in a *timely* and *efficient* manner

W. Kellerer, *et al.*, “How to measure network flexibility? A proposal for evaluating softwarized networks,” *IEEE Communications Magazine*, 2018.

www.networkflexibility.org

We provide

- a **definition** and a **quantitative measure of network flexibility**
- **Optimization** to ease decision making for flexible adaptation
- **Empower** networks for flexibility to cover the future

This work is part of a project that has received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 program grant agreement No 647158 – **FlexNets (2015 – 2020)**.

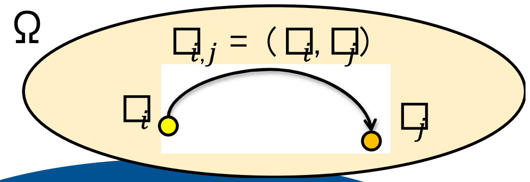
FlexNets
2015 - 2020

European Research Council

Measuring Network Flexibility (our proposal)

(comparing network designs)

adaptation time threshold (T) and cost budget (C)



Input: Constraints T, C

Input: Request set $\boldsymbol{D} = \{d_{i,j} \ldots\}$ with $d_{i,j} \in \Omega \times \Omega$

challenges: request sequence

1. Initialize $\Sigma \coloneqq 0$
2. FOR k = 1:K
   a. Challenge state change $d_{i,j}(t'): d(t'-1) = d_i \rightarrow d(t') = d_j,\ d_i \neq d_j$
   b. Observe τ_k and c_k
   c. If $\tau_k \leq T$ and $c_k \leq C$: $\Sigma \coloneqq \Sigma + 1$
3. END
4. $\varphi(T, C) \coloneqq \Sigma / K$

check if system can adapt and record time and cost

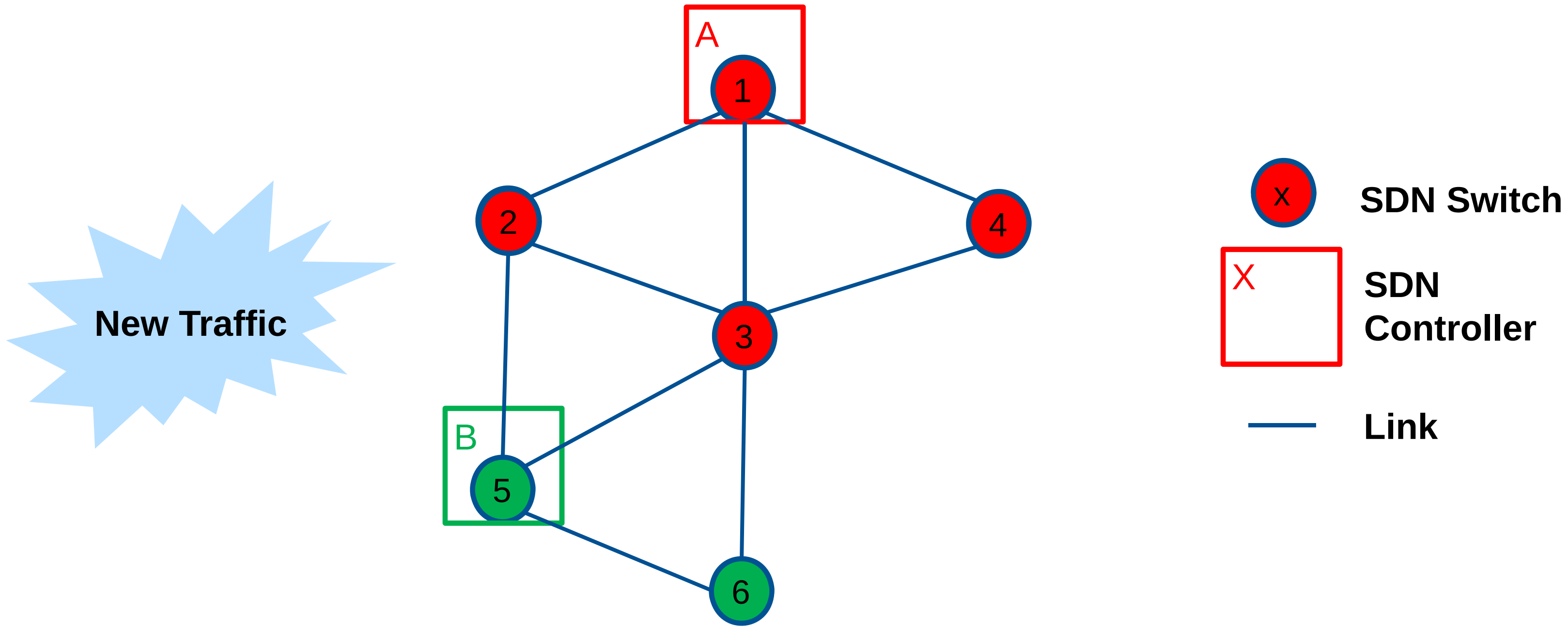
count successes

Flexibility

$$\varphi(T, C) = \frac{|\text{supported requests within constraints } (T, C)|}{|\text{Number of requests}|}$$

M. Klügel, M. He, W. Kellerer, P. Babarczi:
A Mathematical Measure for Flexibility in Communication Networks. **IFIP NETWORKING 2019** (to appear).

Case study 1: Dynamic Controller Placement



- Traffic fluctuations require control plane to adapt in order to achieve better control performance → *Dynamic Control Plane*
  - SDN controller migration & SDN switch reassignment

Flexibility Aspect	New Request	Flexibility Measure	System Objective	Cost in focus
function placement	new flow arrival (from distribution)	fraction of successful controller placements	control performance: (min. avg. flow setup time)	operation latency (OPEX): avg. flow setup time

Case study 1: Dynamic Controller Placement

max. adaptation time threshold
(will be varied)

Varying traffic flow profiles

$$\varphi_T(S) = \frac{|supported\ requests\ within\ T|}{|given\ new\ requests|}$$

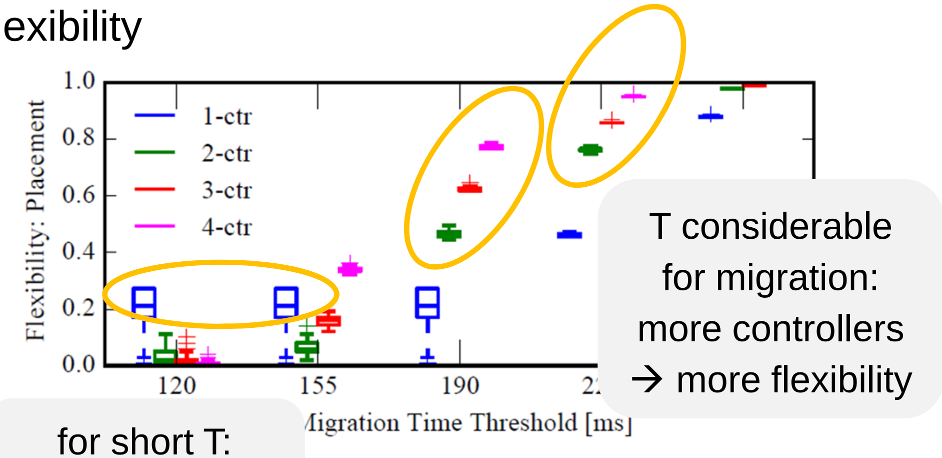
C -> ∞
recorded

SDN controller migration and switch reassignment can be done within T

- Flexibility → Migration Success Ratio
  - Calculate controller migration and switch reassignment time *T_migration*
  - If *T_migration* smaller than T → count as a supported request

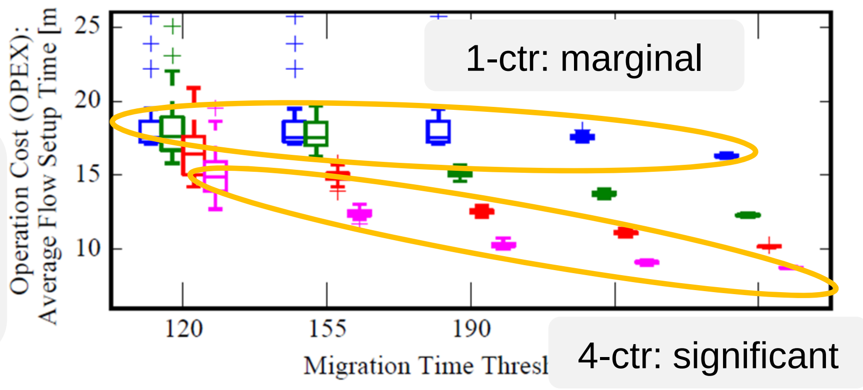
Case study 1: Dynamic Controller Placement

Flexibility



n terms of successful control plane migration.

Cost



(b) Operation cost (OPEX) in terms of the average flow setup time.

intuitive

- **More controllers** (larger migration time threshold) → higher flexibility
- **Single controller** case: more flexible for **tight time threshold** as probability that single controller stays in optimal location is high

unexpected!

- 1 controller → **marginal** performance improvement vs. adaptation T
- 4 controllers → **significant** performance improvement vs. adaptation T
- However, if we consider **all cost factors**, we can reach a trade-off!

M. He, A. Basta, A. Blenk, W. Kellerer, *How Flexible is Dynamic SDN Control Plane?,* IEEE INFOCOM Workshop, SWFAN‘17, Atlanta, USA, May 2017.

Case study 2: SDN Resilience

- Flexibility aspect of **flow configuration** for a **resilience** scenario in an SDN network under a given **recovery time** threshold T.
- Objective: system recovery
- Compare 3 systems: 1:1 protection vs 1+1 protection vs restoration
- Flexibility measure: fraction of recoverable failures
- New requests: all possible **single and dual link failures**

New Request	Flexibility Measure	System Objective	Cost in focus
all possible single and dual failures	fraction of recoverable failures	system recovery: (single and dual failures)	resources overhead (CAPEX): node and link reservation

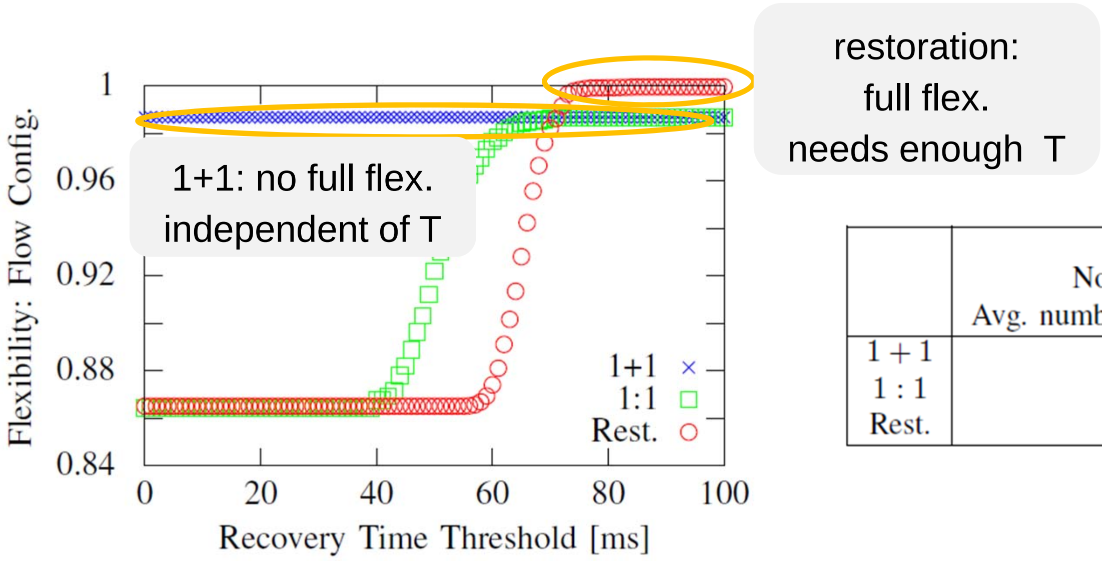
Case study 2: SDN Resilience

- **1:1 protection**
  - primary and backup paths pre-calculated
  - backup path is inactive
  - need switching time between primary and backup in case of a failure

- **1+1 protection**
  - primary and backup paths pre-calculated
  - primary and backup paths are both active
  - recovery time is almost instantaneous!

- **Restoration**
  - no backup path in advance
  - switch detect failure → controller informed → re-routes affected flows
  - recovery time is very critical

Case study 2: SDN Resilience



(a) Flexibility in terms of covered single and dual link failures.

	Resources Cost (CAPEX)	
	Node reservation: Avg. number of flow table entries	Link reservation: Number of required links
1 + 1	11.78	13038
1 : 1	11.78	13038
Rest.	5.05	5400

(b) System resources cost (CAPEX) in terms of nodes and links used for reservation.

intuitive

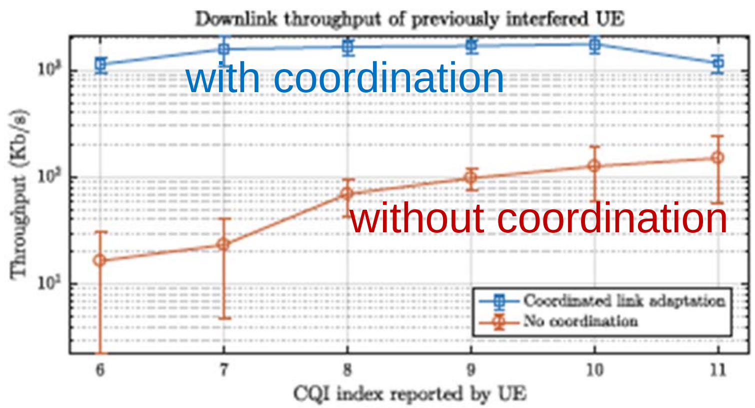
- 1+1 **can not** reach **full flexibility**
- However, 1+1 is obviously **independent** of **recovery time**
- Restoration can cover **all failures** if given enough recovery time

intuitive

- Protection imposes more than **2x capex overhead** than restoration
- Again, if we consider **all cost factors**, we can reach a trade-off!

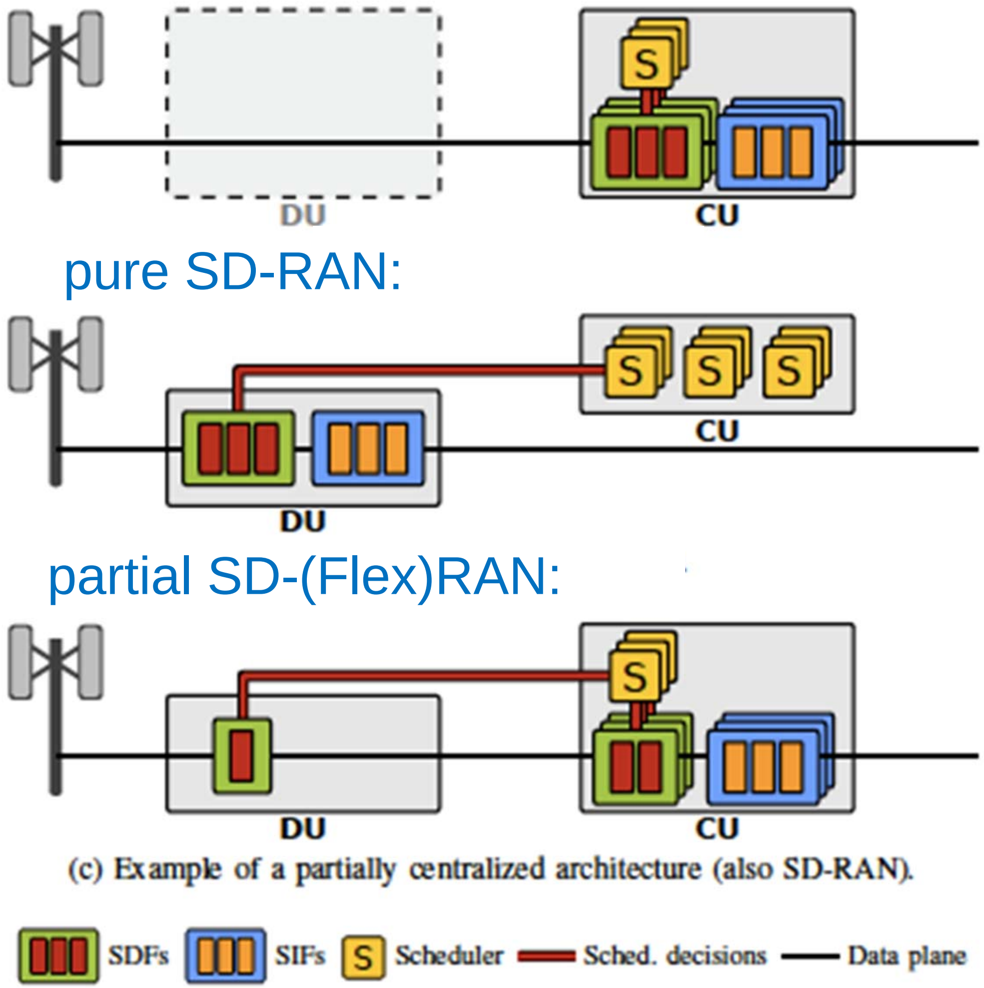
Case study 3: FlexRAN (ongoing work)

- Radio Access Network plus SDN/NFV
  → unexplored flexibility
- our use case: coordinated scheduling
- initial results: PoC



- next: quantify flexibility
  flexibility: ratio of successful handling of request

CloudRAN:



- We can **measure flexibility**
  so far relatively between multiple systems

- Results can be **less intuitive** than one might think

- Measure can be used to **design for flexibility**

Optimize for Flexibility φ

Measure Phase

- Optimize for performance metric (e.g. latency and throughput)
- quantify flexibility value (success ratio)

Design Phase

- Optimize for flexibility measure, **decide** system design parameters (e.g., bandwidth, # base stations, etc.)

Use Case example: **Dynamic Controller Placement Problem**

- Requests: traffic profiles with target average flow setup time
- Objective: max. flexibility (success: # accomodated traffic profiles)
- Design parameters: # data centers and their locations

Optimize for Flexibility

- **Design of methods to optimize for flexibility, i.e. speedup the decision on the optimal design**

  - offline optimization

  - **online** optimization
    - <u>adaptation time speedup through machine learning</u>
    - *empower* a network to optimize for the future

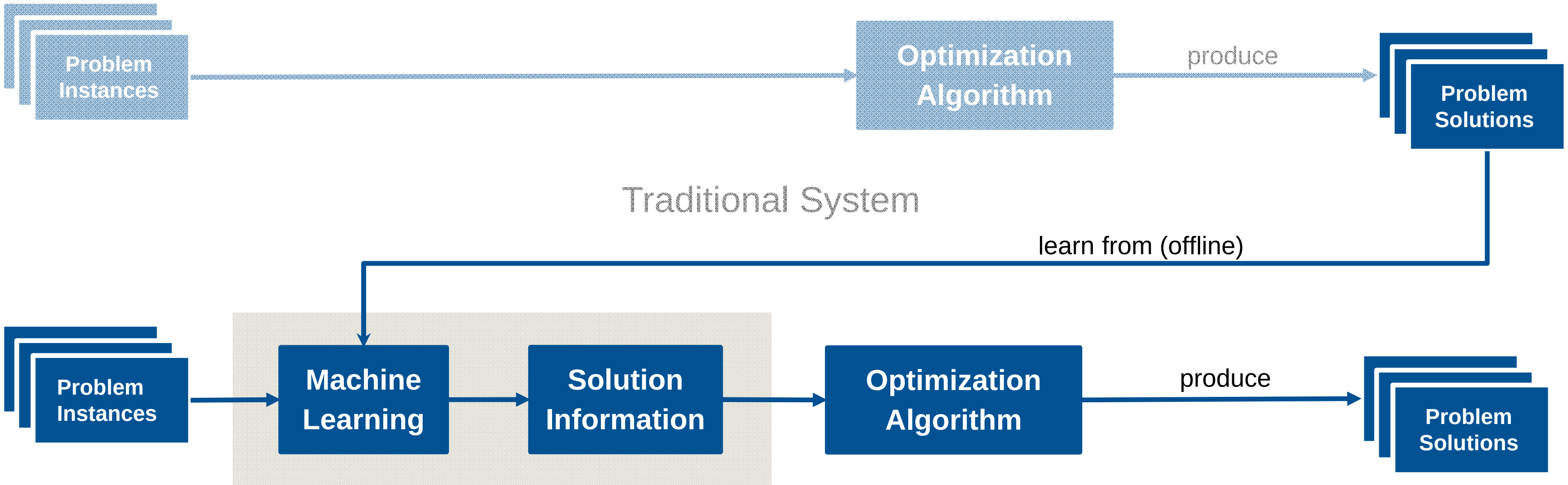
Speedup adaptation time

- **Adaptation time** is very important for flexible networks
- Examples:
  - Where to migrate a function? e.g., SDN controller
  - How to (re-)embed virtual networks/flows, e.g. for resilience
- How can we speedup?
- Optimization (= decision making) becomes an algorithmic problem with highly flexible systems (options increase)
- Yet another heuristic for a specific case study?

We propose:

- Keep your favourite optimization algorithms and
- ***Boost your network algorithm with ML preprocessing***

How can we boost the solving of the related optimization problems (leaving you algs. untouched)?



o'zapft is framework [5]

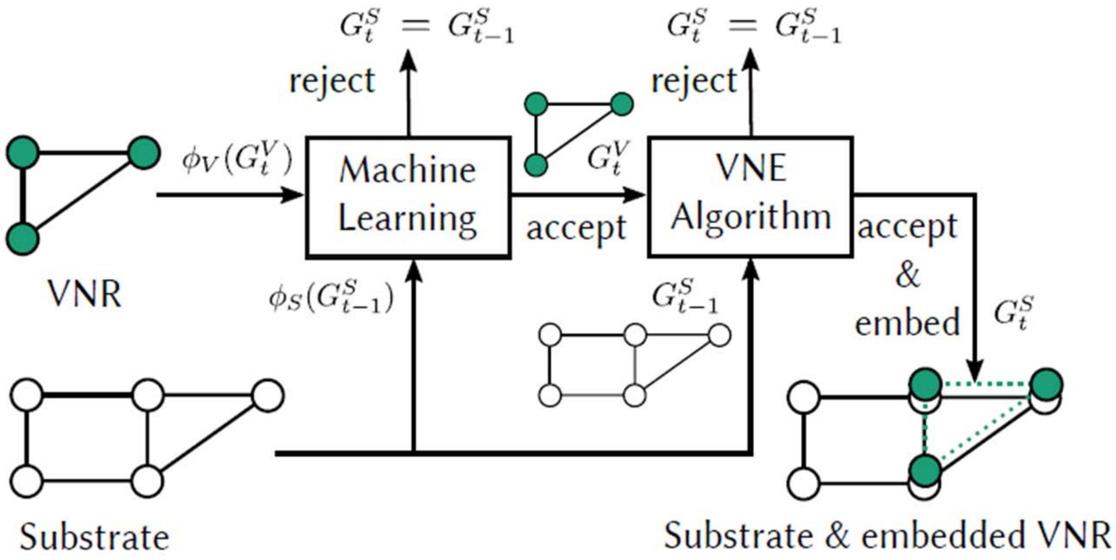
State-of-the-art: Neglects produced data!

Idea: Use problem/solution data generated by algorithms regularly solving problems

A. Blenk, P. Kalmbach, S. Schmid, W. Kellerer: ***o'zapft is: Tap Your Network Algorithm's Big Data!*** ACM SIGCOMM 2017 Wrksp. on Big Data Analytics and Machine Learning for Data Communication Networks (Big-DAMA), 2017.

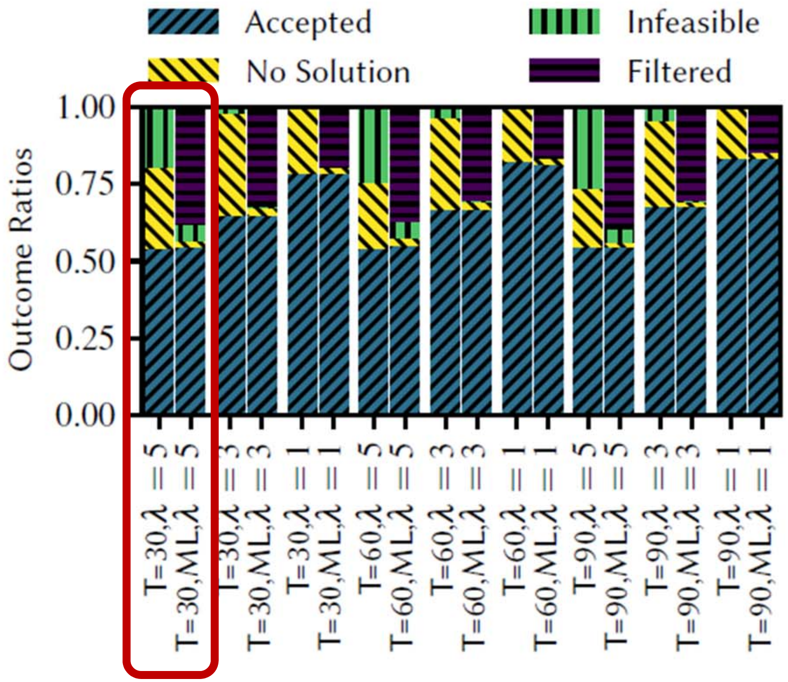
Data Available: P. Kalmbach, J. Zerwas, M. Manhart, A. Blenk, S. Schmid, W. Kellerer. Data on "o'zapft is Tap Your Network Algorithm's Big Data!",2017 https://doi.org/10.14459/2017md1361589

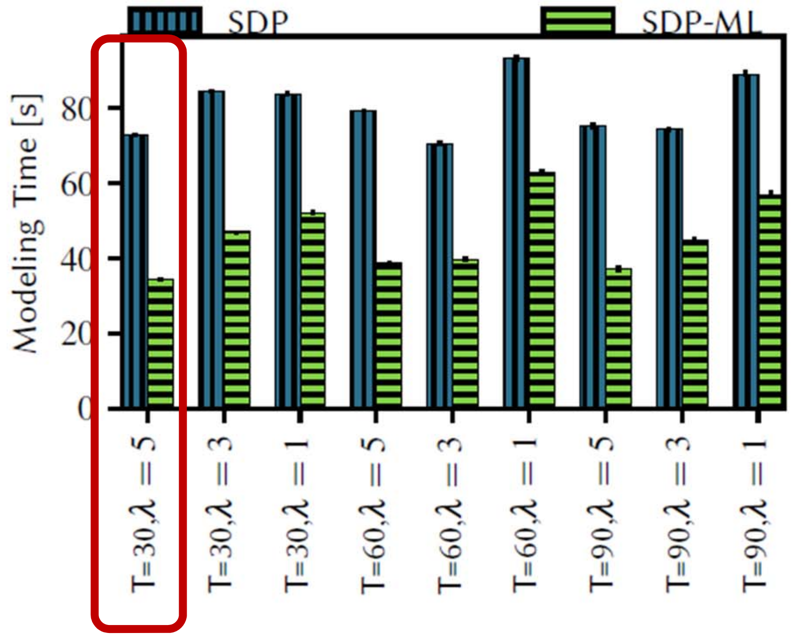
Case Study: Predicting Acceptance Probabilities of VNE Requests



- Supervised learning: use data with accepted and rejected requests! Offline training!
- Recurrent neural network (RNN) for classification
- Filter infeasible and requests with unacceptable algorithm runtime (“no solution“)

Can we speed-up optimal algorithms using admission control?



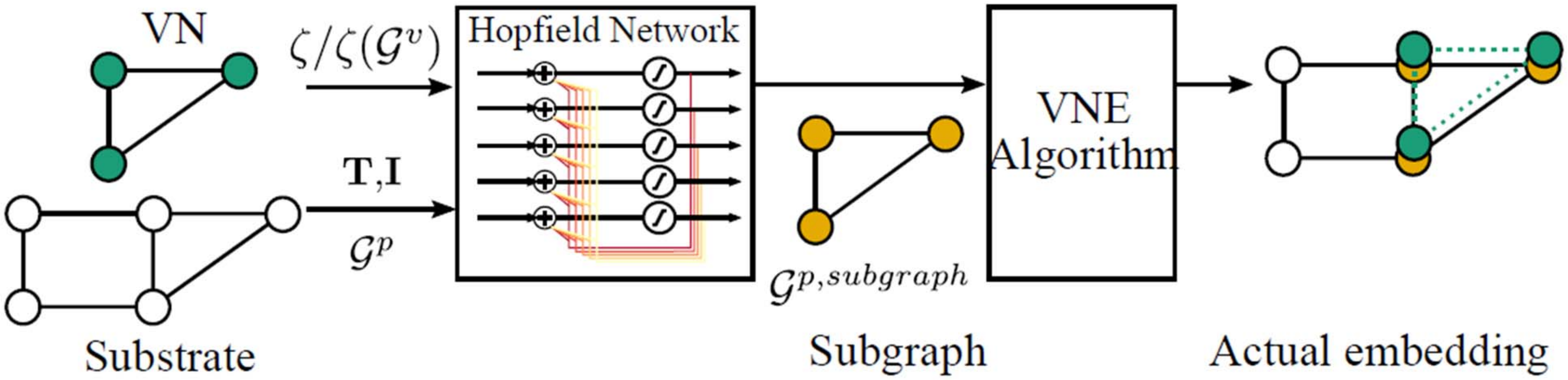


Efficient Filtering of infeasible and unacceptable requests

Efficient saving of model creation time

Latest Results: Neurovine

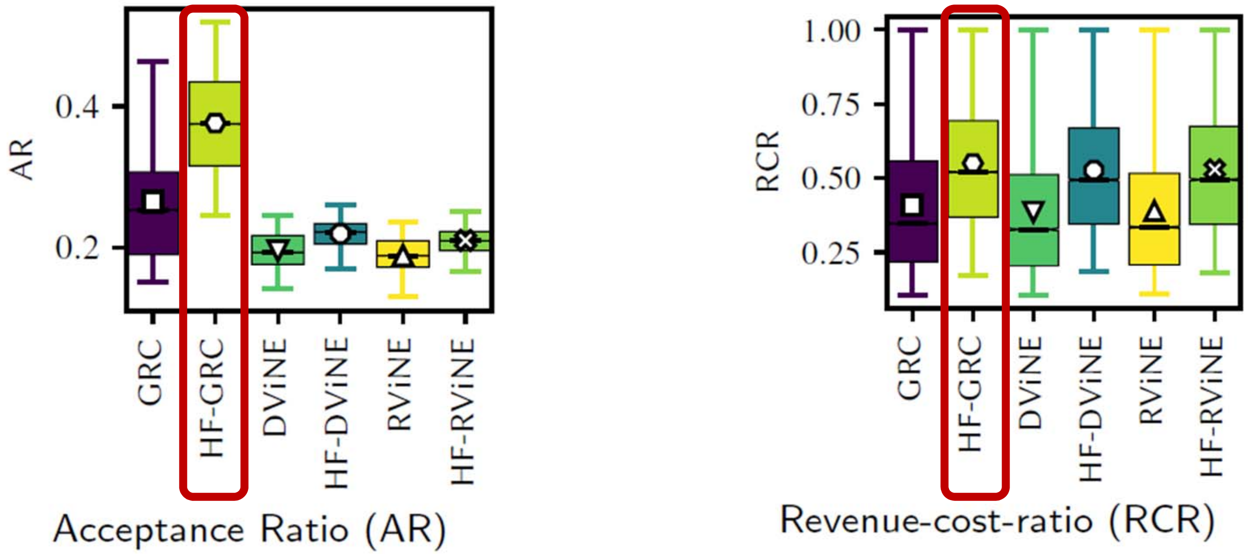
Hopfield neural network to preprocess (subgraph extraction) VNE algorithms
– tailored filtering



- Idea: Extract subgraph with physical nodes close to each other and high available capacities

A. Blenk, P. Kalmbach, J. Zerwas, M. Jarschel, S. Schmid, W. Kellerer: ***NeuroViNE: A Neural Preprocessor for Your Virtual Network Embedding Algorithm*** IEEE INFOCOM 2018 (main conference), Honolulu, HI, USA, April 15-19, 2018.

Neurovine:
Efficiency on Real Network Topologies



- VNE algorithms (GRC, DViNE, RViNE) vs. Hopfield variants (HF-GRC, HF-DViNE, HF-RViNE)
- NeuroViNE accepts more networks with less costs

Optimize for Flexibility

- **Design of methods to optimize for flexibility**

  - offline optimization

  - **online** optimization
    - adaptation time speedup through machine learning
      - ➢ we still have a clear objective here to optimize for

    - *empower* a network to **optimize for the future**

Empower your network

optimize for the (unknown) future:

- prepare for possibly unexpected events → **flexibility**

we need:

- **(online) self-optimization**

self-driving networks (Rexford, Feamster): *networks which measure, analyze and control themselves in an automated manner, reacting to changes in the environment*

- **prepare for the unknown**

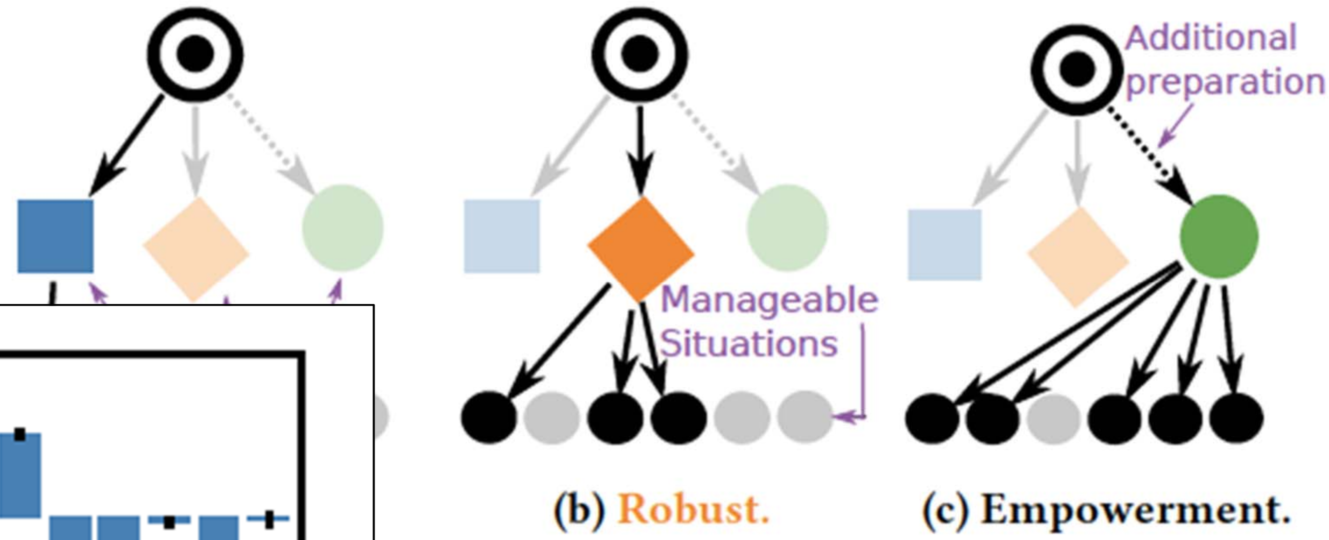
We propose:

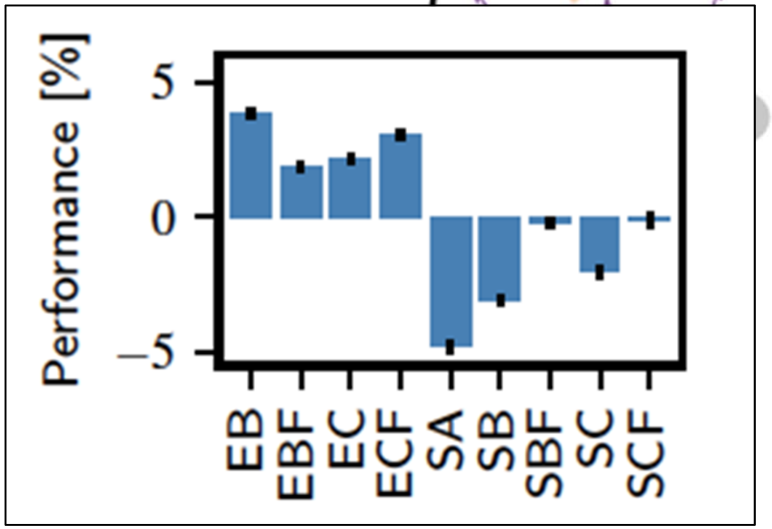
- use ***empowerment*** for preparedness

P. Kalmbach, J. Zerwas, P. Babarczi, A. Blenk, W. Kellerer, S. Schmid: ***Empowering Self Driving Networks***, ACM SIGCOMM 2018 workshop on self-driving networks August 2018.

Empowering Networks

empowerment: quantify the influence of an agent on its environment: agent (several actuators, 1 sensor) restructures networks to maximize options (c) - not an objective as in optimization (a) and (b)





initial result:
empowerment-based approaches (EB, EC) can outperform ILP („0“-line) and heuristics (SA)

P. Kalmbach, J. Zerwas, P. Babarczi, A. Blenk, W. Kellerer, S. Schmid: ***Empowering Self Driving Networks***, ACM SIGCOMM 2018 workshop on self-driving networks August 2018.

Summary (1)

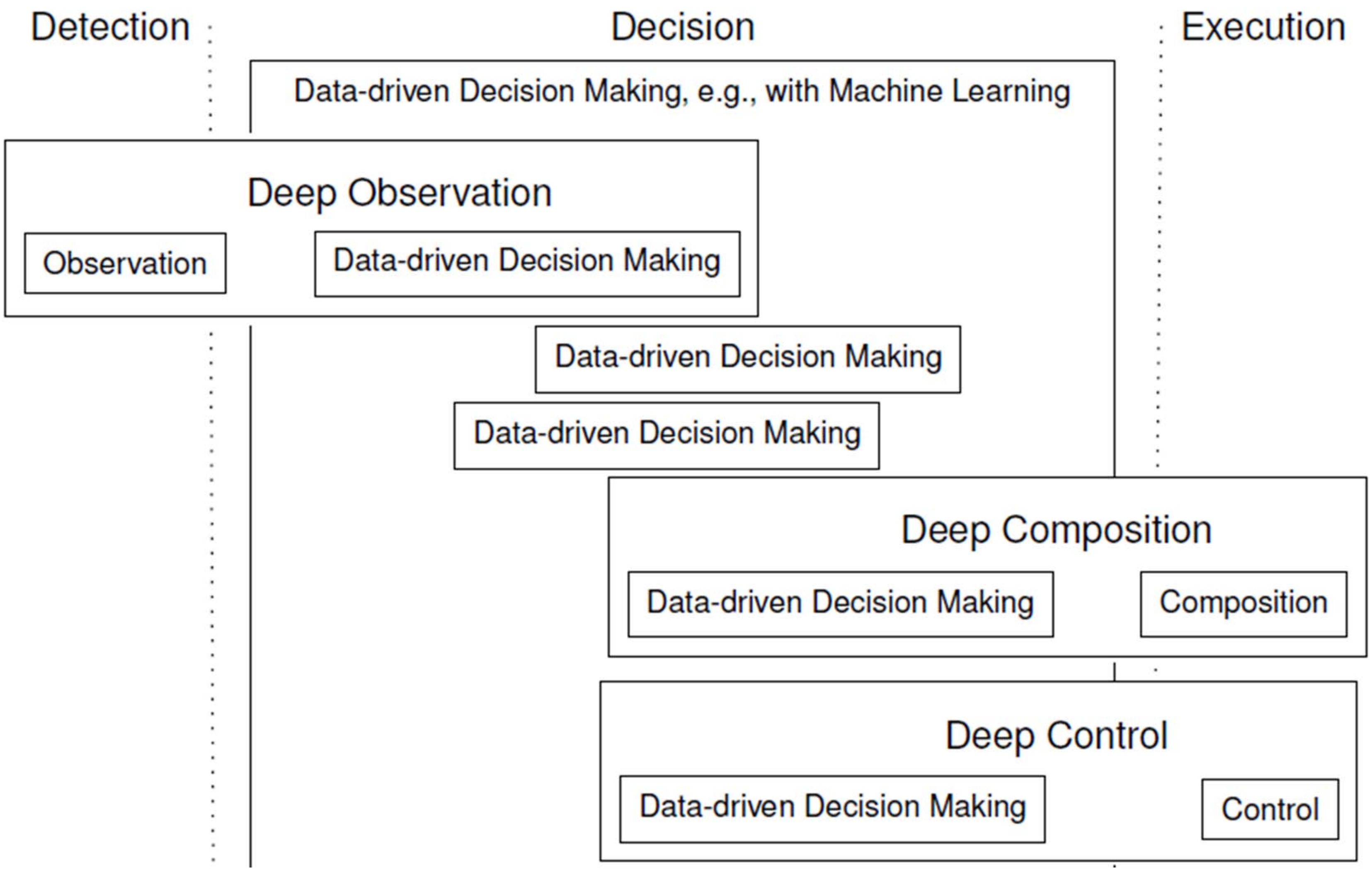
- Softwarized Networks provide **flexible network adaptation**
- Flexibility need to be quantified **(Challenge 1)**
  - to compare flexible systems
  - to explicitly **design for flexibility**

decision making

- **Challenge 2: (online) optimization for flexibility**
  - Speedup of opt. algorithms through ML-preprocessing
  - **Empowerment** to optimize for flexibility to cover the future

join us on networkflexibility.org

Big Picture: *Let's go for*
Deep Observation, Composition and Control



W. Kellerer, et al. Adaptable and Data-Driven Softwarized Networks: Review, Opportunities and Challenges. Proc. of the IEEE, 2019.

Summary (2)

- Softwarized Networks provide **flexible network adaptation**
- Flexibility need to be quantified **(Challenge 1)**
  - to compare flexible systems
  - to explicitly **design for flexibility**

decision making

- **Challenge 2: (online) optimization for flexibility**
  - Speedup of opt. algorithms through ML-preprocessing
  - **Empowerment** to optimize for flexibility to cover the future

join us on networkflexibility.org

References

W. Kellerer, P. Kalmbach, A. Blank, A. Basta, S. Schmid, M. Reisslein: *Adaptable and Data-Driven Softwarized Networks: Review, Opportunities and Challenges*. **Proc. of the IEEE**, 2019 (open access).

M. Klügel, M. He, W. Kellerer, P. Babarczi: *A Mathematical Measure for Flexibility in Communication Networks*. **IFIP NETWORKING 2019** (to appear).

M. He, A. Martinez Alba, A. Basta, A. Blenk, W. Kellerer. *Flexibility in Softwarized Networks: Classifications and Research Challenges*. **IEEE Communication Surveys & Tutorials**, 2019.

P. Kalmbach, J. Zerwas, P. Babarczi, A. Blenk, W. Kellerer, S. Schmid, *Empowering Self-Driving Networks.* **ACM SIGCOMM 2018 Workshop** on Self-Driving Networks - SelfDN 2018

A. Blenk, P. Kalmbach, J. Zerwas, M. Jarschel, S. Schmid, W. Kellerer: *NeuroViNE: A Neural Preprocessor for Your Virtual Network Embedding Algorithm.* **IEEE INFOCOM** 2018 (main conference), Honolulu, HI, USA, April 15-19, 2018.

W. Kellerer, A. Basta *et al.*, *How to measure network flexibility? A proposal for evaluating softwarized networks*, ***IEEE Communications Magazine***, 2018.

A. Blenk, P. Kalmbach, S. Schmid, W. Kellerer: *o'zapft is: Tap Your Network Algorithm's Big Data!* **ACM SIGCOMM 2017 Wrks**. on Big Data Analytics and Machine Learning for Data Communication Networks (Big-DAMA), 2017.

M. He, A. Basta, A. Blenk, W. Kellerer, *How Flexible is Dynamic SDN Control Plane?,* **IEEE INFOCOM Workshop**, SWFAN‘17, Atlanta, USA, May 2017.

W. Kellerer, A. Basta, A. Blenk, *Using a Flexibility Measure for Network Design Space Analysis of SDN and NFV*, **IEEE INFOCOM Workshop**, SWFAN’16, SF, USA, April 2016.