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

Flexible Networking based on Network Function Virtualization and In-Network Processing

Wolfgang Kellerer

Technical University of Munich

Darmstadt, April 7, 2017

SFB MAKI

Scientific Workshop 2017

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)**.

Introduction

- Networking today: new requirements from vertical industries, dynamically changing user behavior, and global digitalization

- Less (explicitly) addressed: ***flexibility*** and hence ***adaptation***

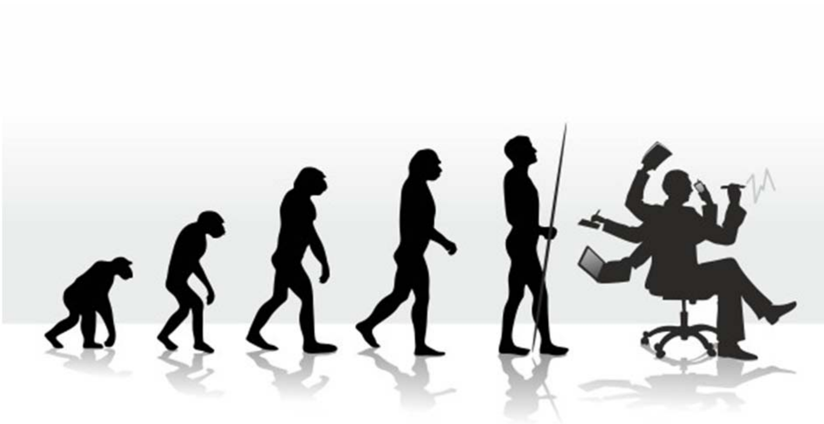


Image source: http://www.paleoplan.com

- In this talk, I will …
  … explain some technologies for network adaptation …
  … give some concrete examples …
  … and present our way to measure flexibility

FlexNets

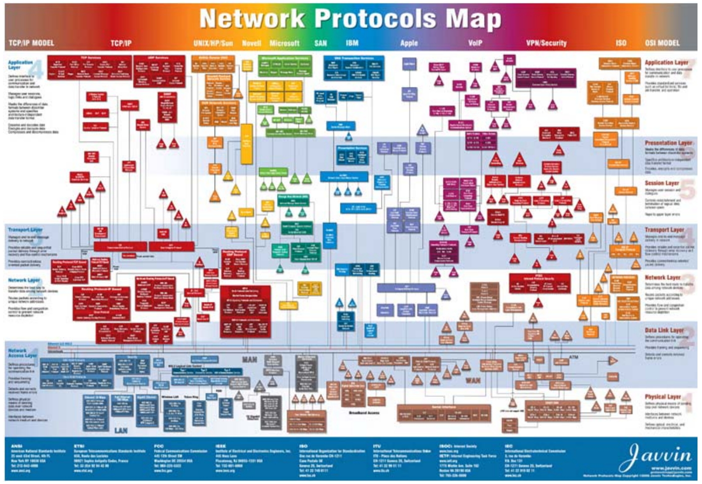
2015 - 2020

European Research Council
Established by the European Commission

The Internet

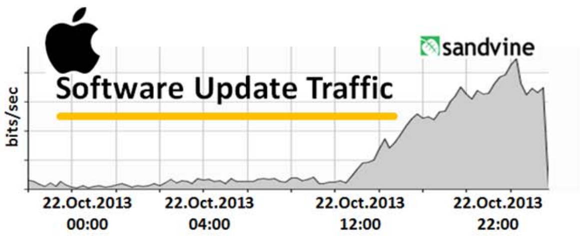
… is able to adapt its resources
… *somehow* (best-effort, TCP elasticity, BGP, OSPF)

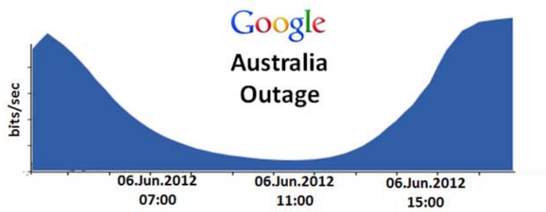
early-days simplicity
→ complex and ossified network system

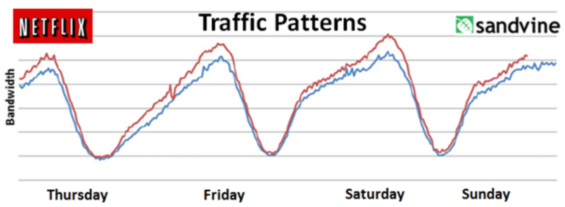


source: SFB MAKI

very slow adaptation to new requirements
→ reaction to dynamic changes hardly possible



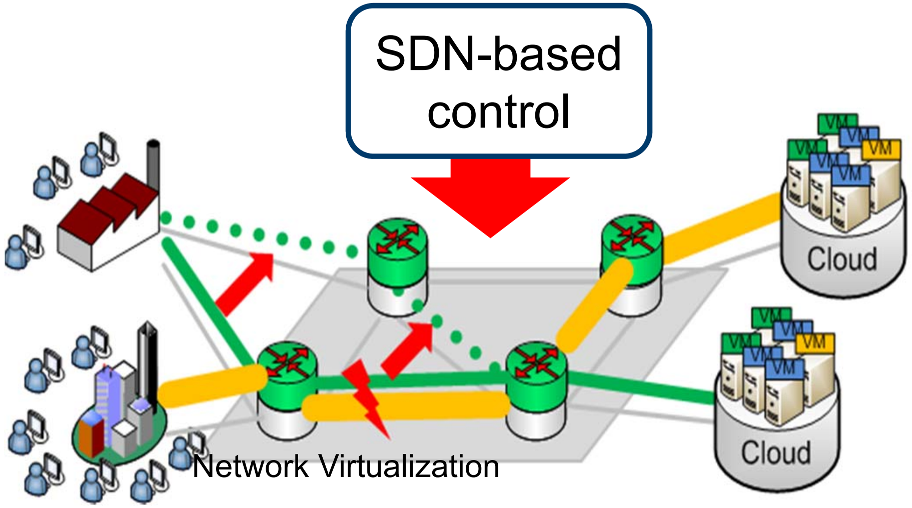


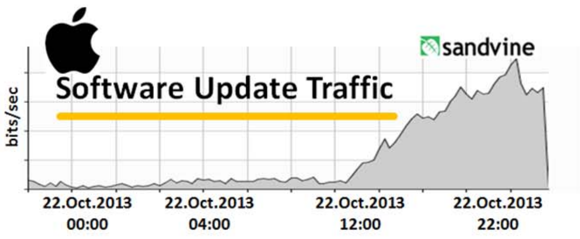


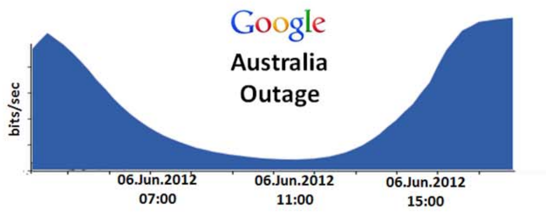
In particular, emerging concepts such as …

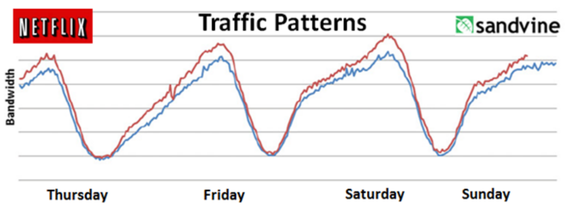
Software Defined Networking (SDN) and Network Function Virtualization (NFV)

… *promise* to adapt networks and functions on demand in software









Outline of this Talk

- Technologies: NFV, SDN and In-Network Processing (INP)

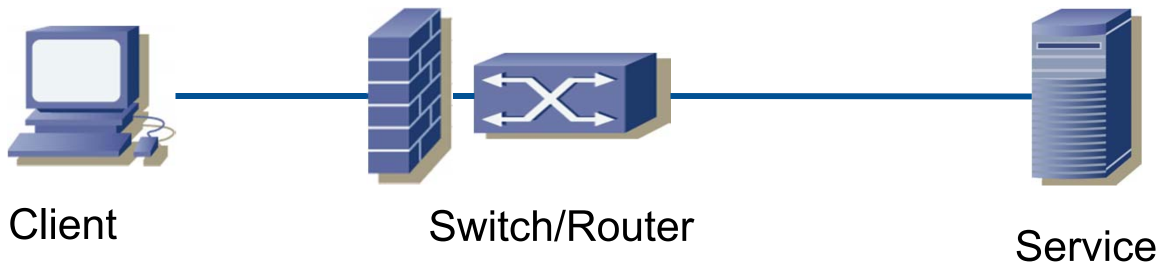
- Use Cases:
  - NFV + INP/SDN → Function Placement Problem (FPP)
  - Transition between concepts

- Towards a flexibility measure for NFV/SDN/INP networks

- Use Cases: Controller Placement and Migration
  (and the issue of time to adapt and cost to spend)

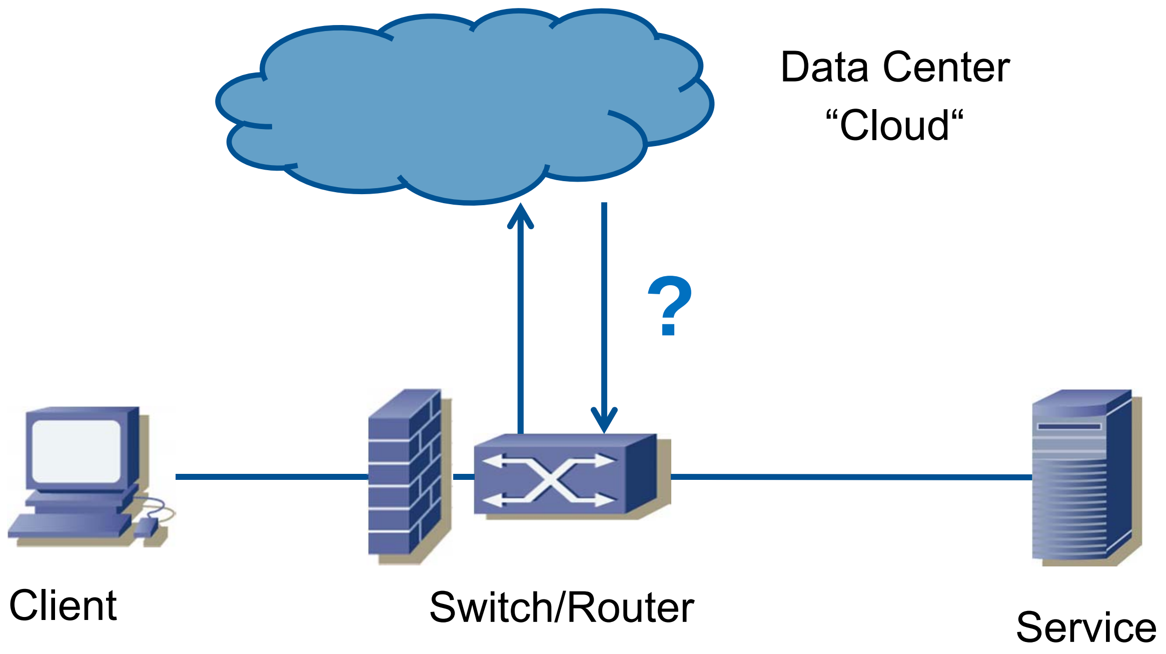
NFV, SDN and INP in a Nutshell

- Let‘s take a network function: firewall – a special function on a special device („middlebox“)



NFV, SDN and INP in a Nutshell

- Let‘s take a network function: firewall – function on a special device
- Network Function Virtualization:
  virtualize the firewall and move it into a data center

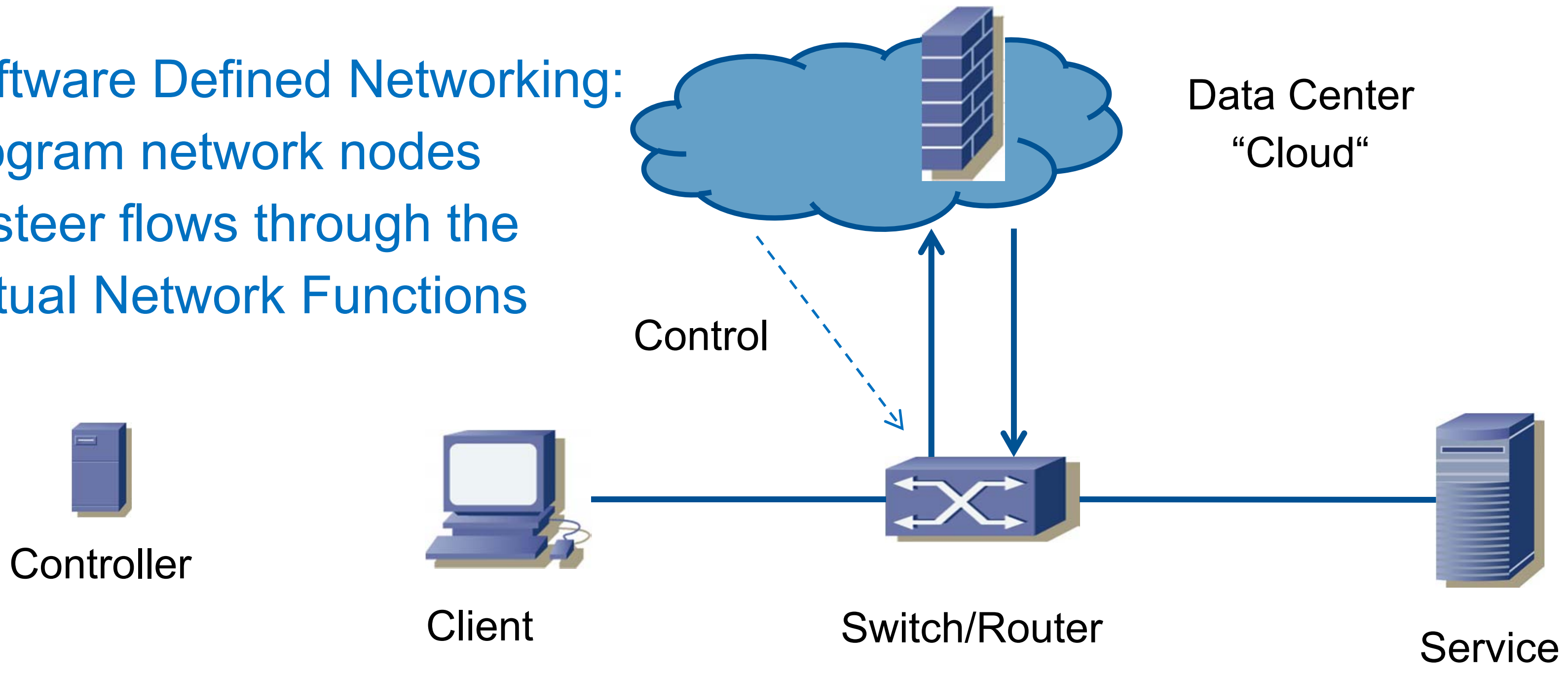


NFV, SDN and INP in a Nutshell

- Let‘s take a network function: firewall – function on a special device

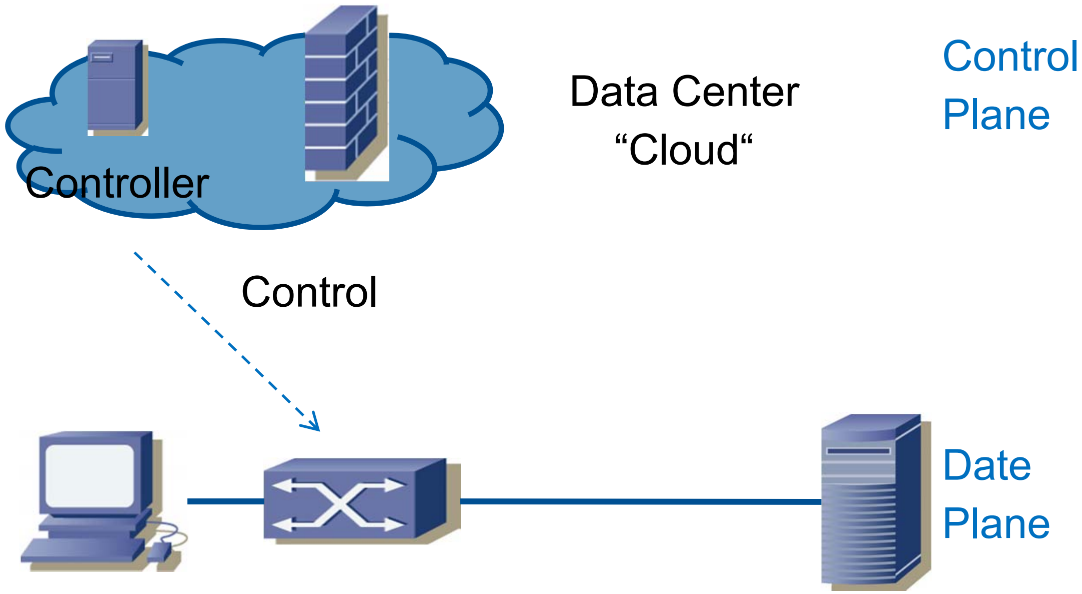
- Network Function Virtualization:
  virtualize the firewall and move it into a data center

- Software Defined Networking:
  program network nodes
  to steer flows through the
  Virtual Network Functions



NFV, SDN and INP in a Nutshell

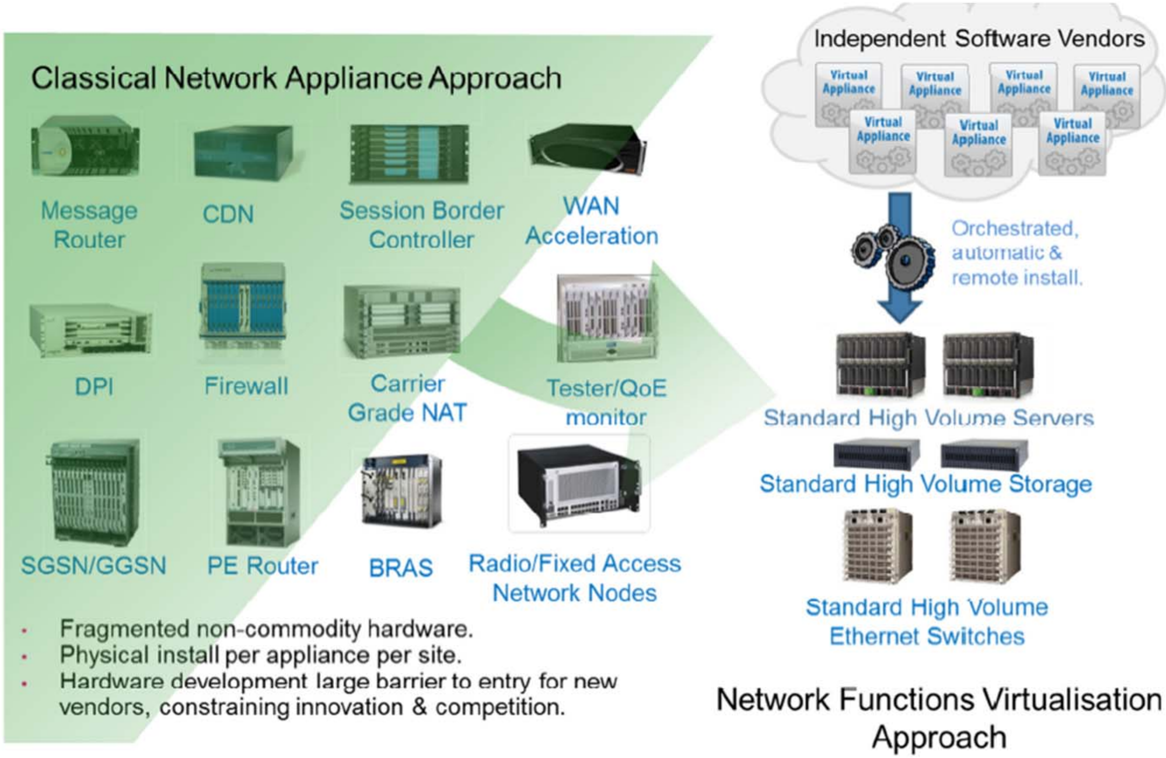
- Let‘s take a network function: firewall – function on a special device
- Network Function Virtualization:
  virtualize the firewall and move it into a data center
- Software Defined Networking:
  program network nodes
  to steer flows through the
  Virtual Network Functions
- In-Network Processing:
  program a network node
  to act as a firewall



via SDN: Separation of Data Plane and Control Plane

Network Function Virtualization (NFV)

Today: network functions run on dedicated, proprietary hardware (middleboxes)
Goal: realize functions in software and run on standard hardware



Advantages:
- CAPEX savings through COTS platforms
- OPEX savings due to centralization of the administration
- OPEX+CAPEX savings through higher resource efficiency (scaling of functions to need)

Network Function Virtualization

What is needed?

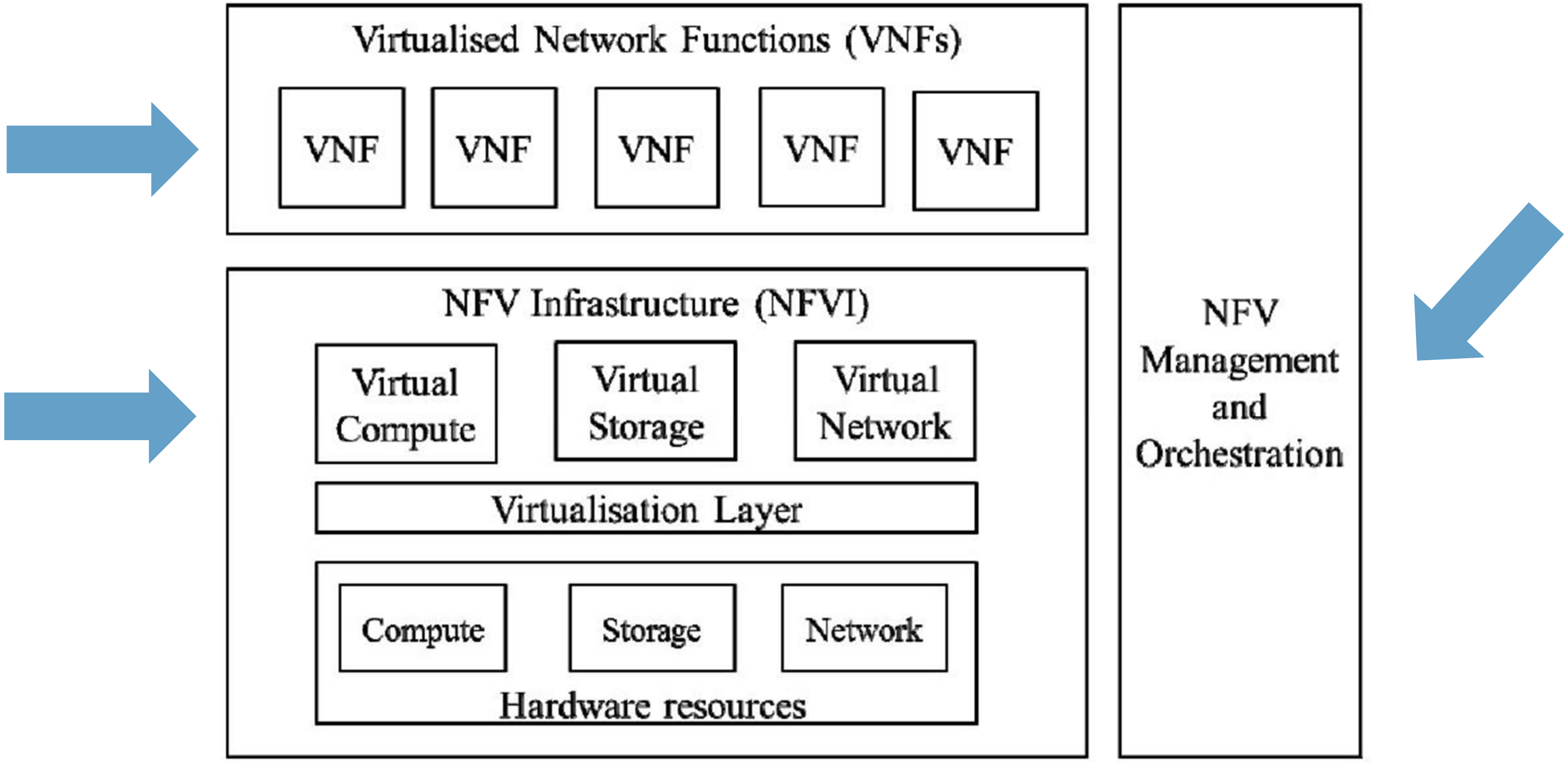


Figure 1: High-level NFV framework

ETSI GS NFV 002 V1.1.1 2013 „Network Function Virtualization (NFV) Architectural Framework"

End-to-End Network Service with NFV

Function chain: logical link

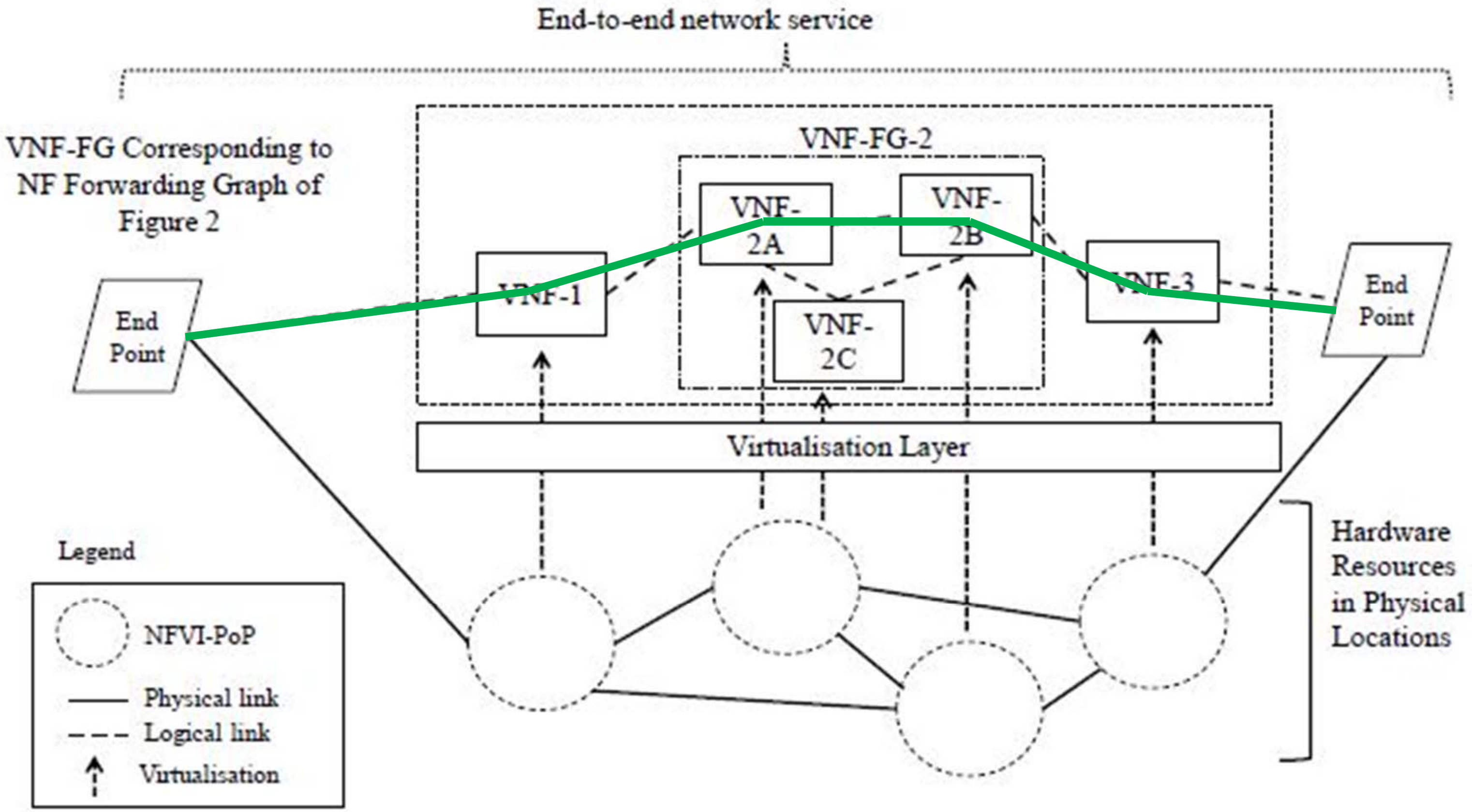


Figure 3: Example of an end-to-end network service with VNFs and nested forwarding graphs

ETSI GS NFV 002 V1.1.1 2013 „Network Function Virtualization (NFV) Architectural Framework“

Software Defined Networking (SDN)

- **Virtualized Network Functions** require that
  the data flows can be forwarded dynamically to the respective network function(s) residing in data centers

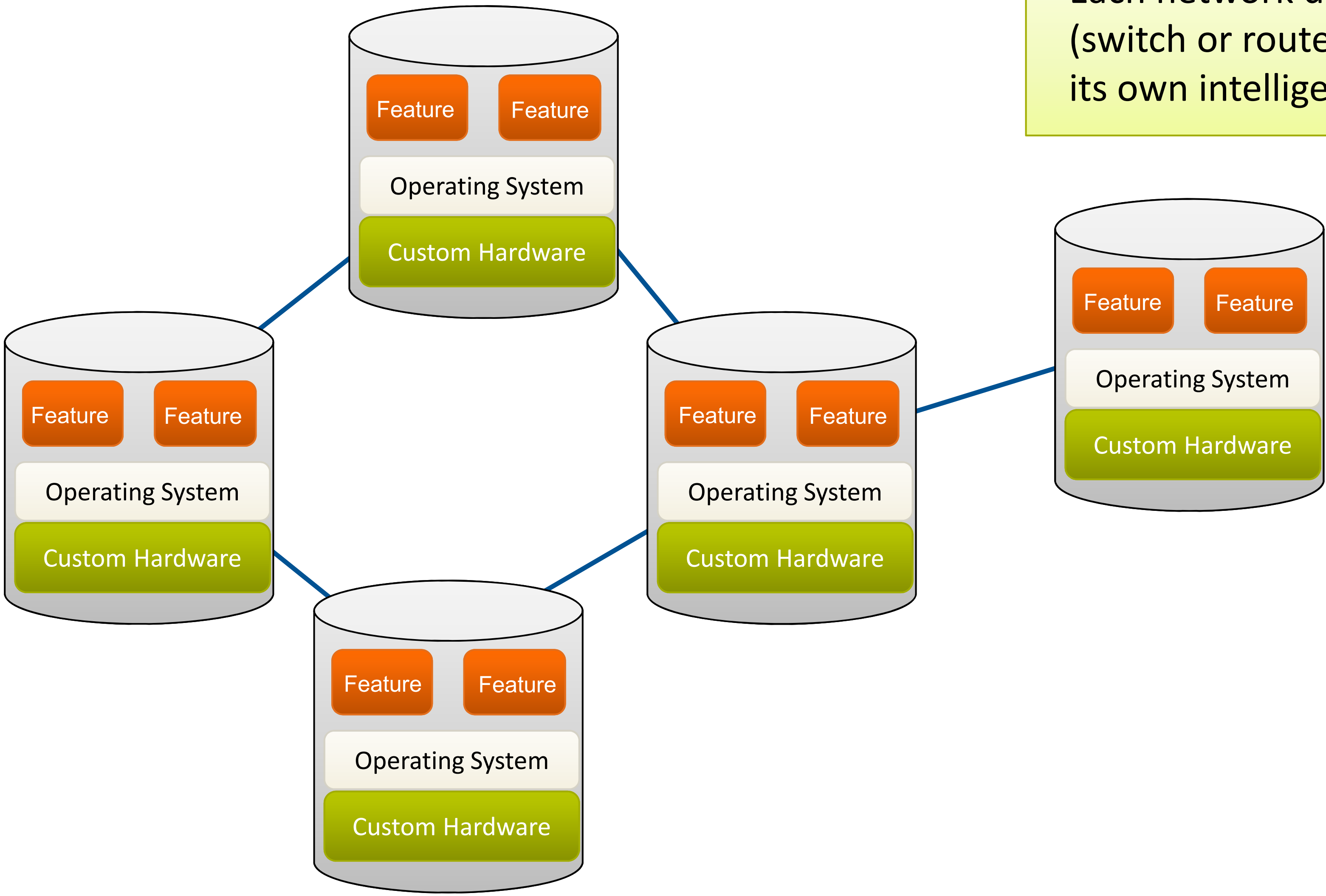
- Requires dynamic and flexible networking

- **Software Defined Networking** is a solution
  - Separation of data plane (hardware) and control plane (software) on a switch
  - Logically centralized SDN controller operates control plane
  - Programming of the network
  - ONF OpenFlow protocol as a standard protocol realizing the interface between forwarding hardware and controller
  - An SDN eco system is developing

What is Software Defined Networking?

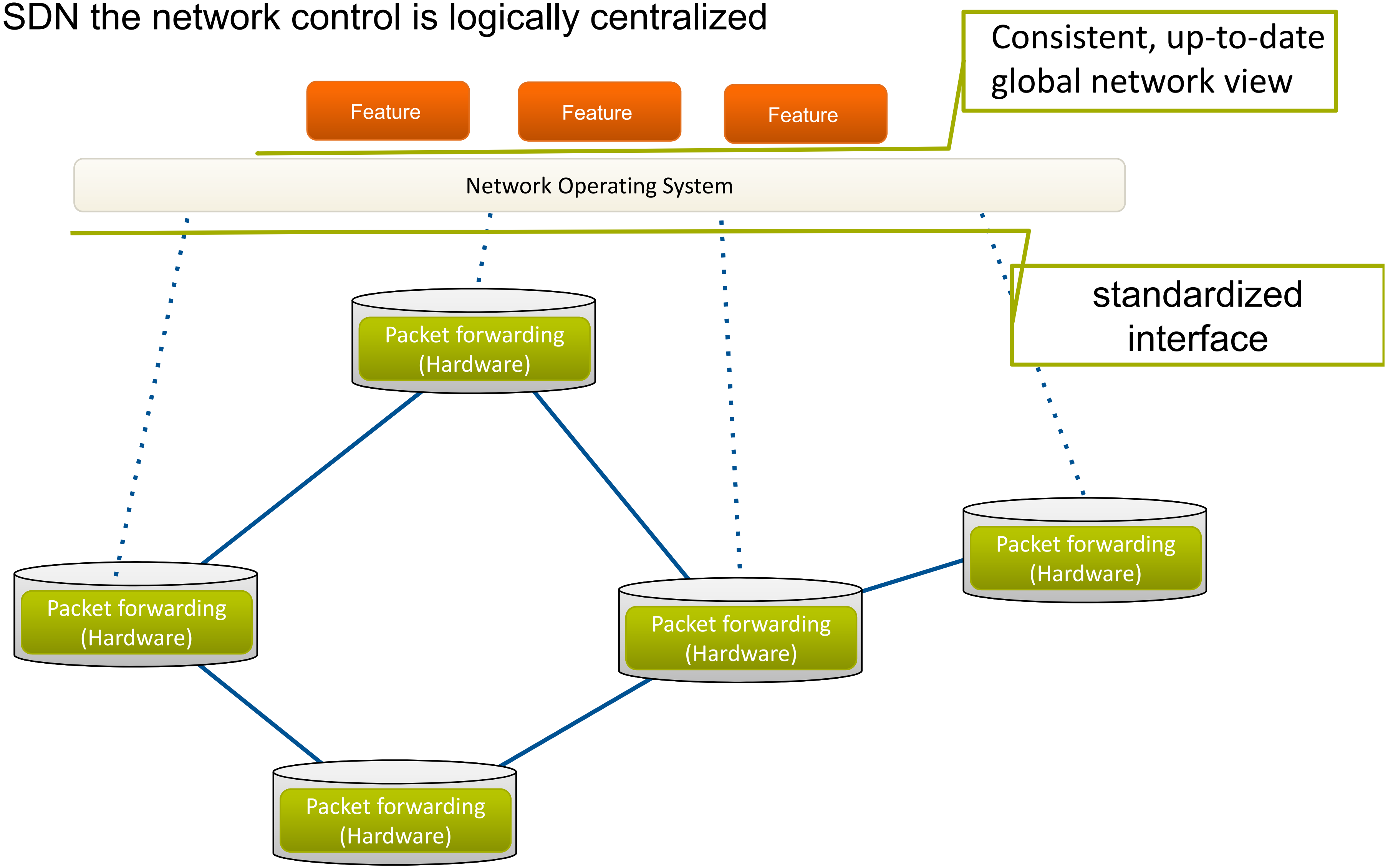
Network intelligence is distributed in traditional networking

Each network device (switch or router) has its own intelligence



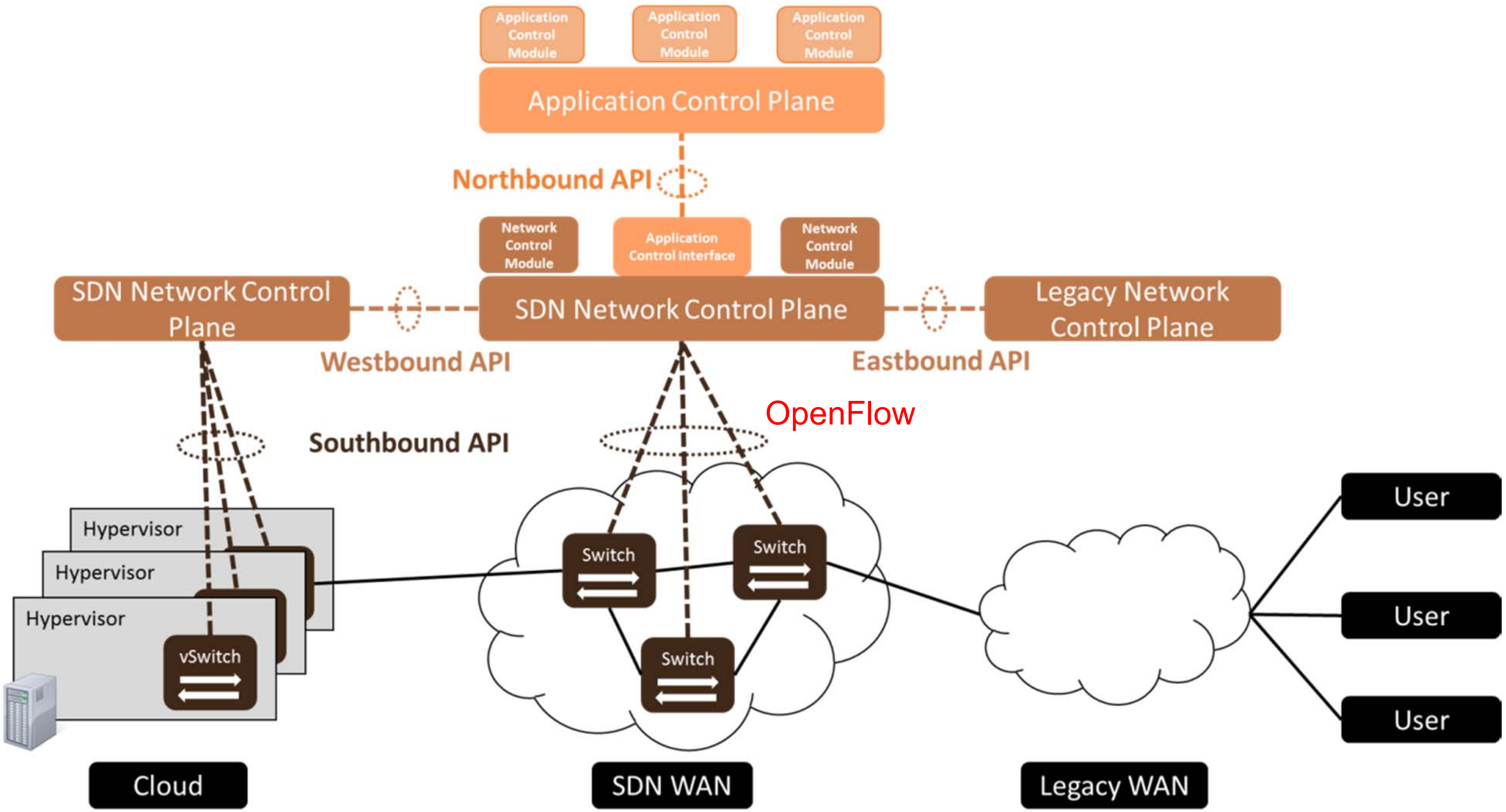
What is Software Defined Networking?

In SDN the network control is logically centralized



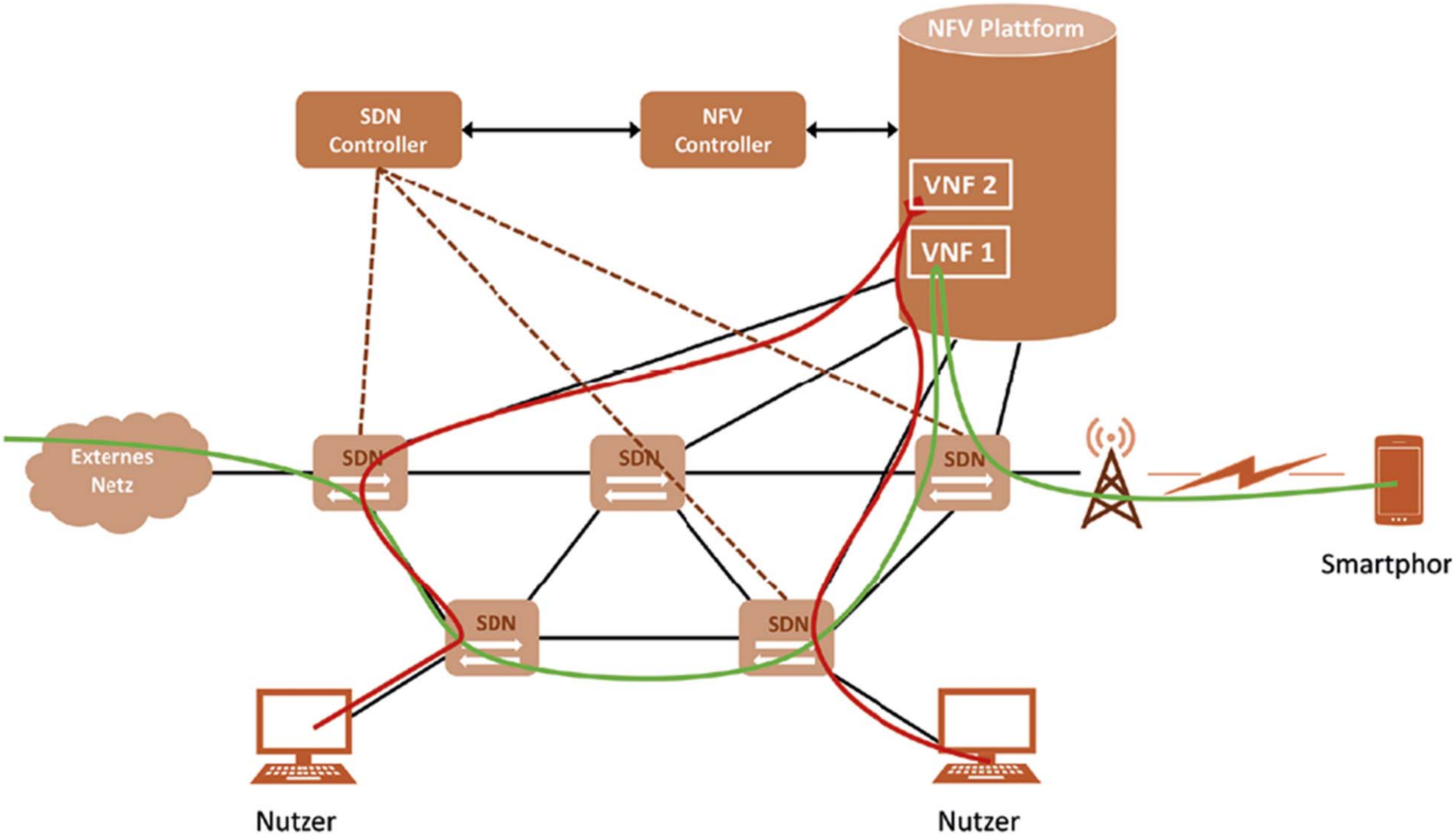
SDN Eco System

A Compass for SDN *



*T. Zinner, M. Jarschel, T. Hossfeld, P. Tran-Gia, W. Kellerer, **Interfaces, Attributes and Use Cases – A Compass for SDN,** IEEE Communications Magazine, June 2014.

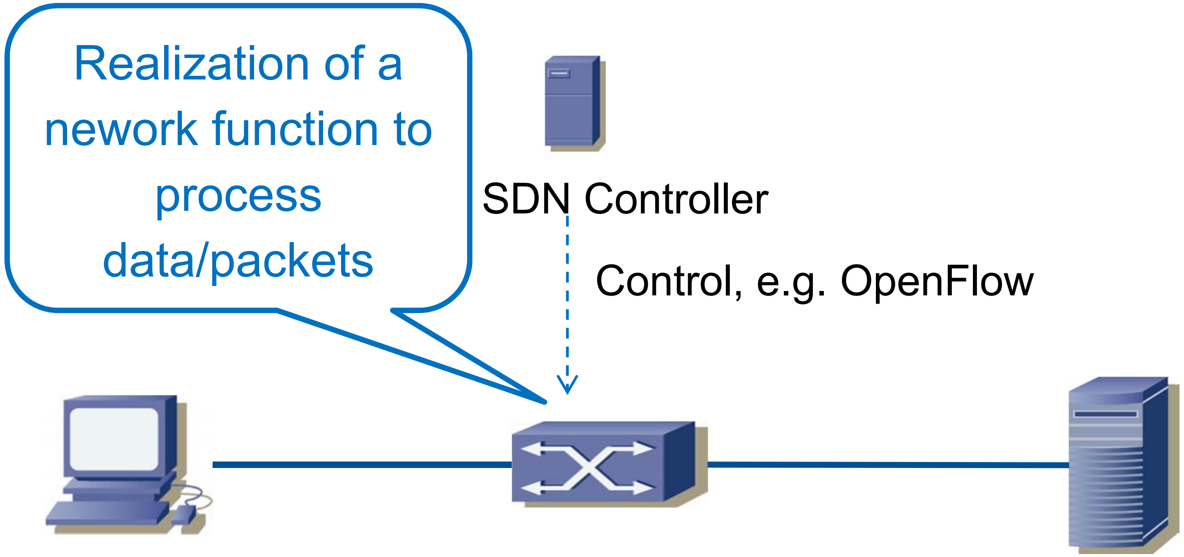
Illustrating SDN and NFV interworking



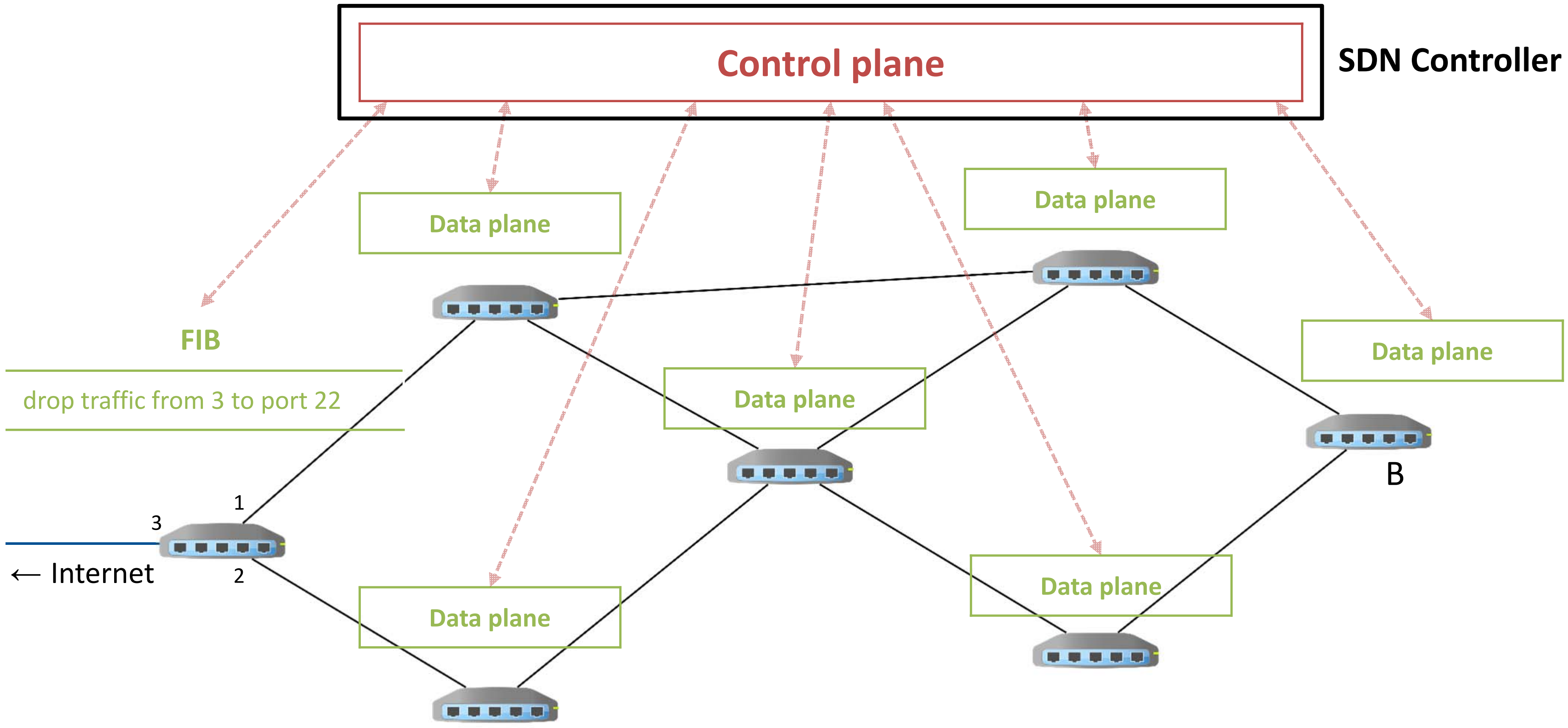
A. Blenk, A. Basta, W. Kellerer, T. Zinner, F. Wamser, P. Tran-Gia. Network Functions Virtualization (NFV) und Software Defined Networking (SDN): Forschungsfragen und Anwendungen. ITG News 1/2015, January 2015.

SDN can do more: In-Network Processing

- SDN controller can realize a network function via rules in a programmable network element
- advantage:
  processing in the network is faster (no detour in/out cloud)



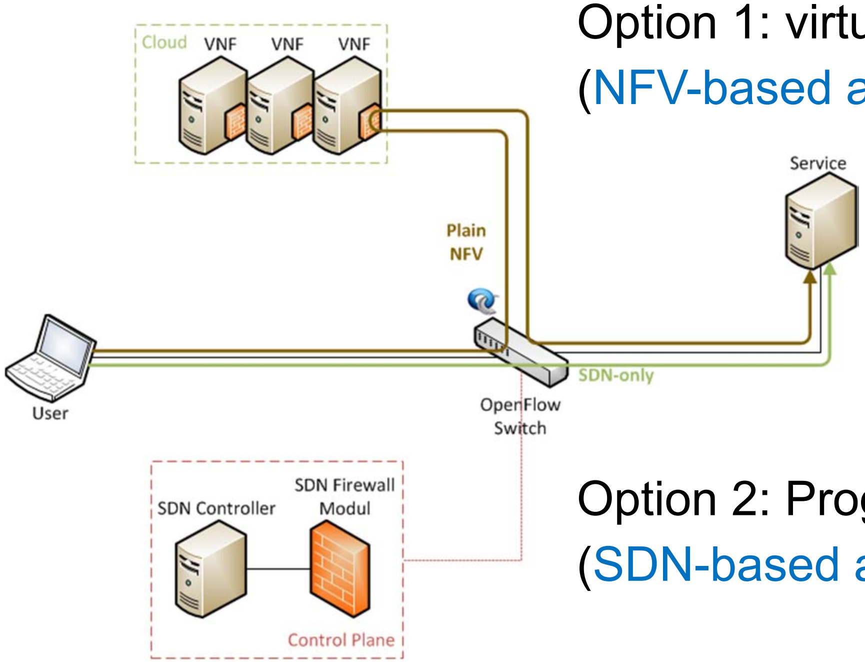
Use Case: Firewall



Use Case: Firewall (with NFV vs. SDN/INP)

Enterprise networks: Need for fine-grained and flexible security solutions

Our approach [3]: combine SDN and NFV to adapt to changing demands



Option 1: virtualized firewall in the cloud
(NFV-based approach)

Option 2: Programmable Switch → firewall
(SDN-based approach)

[3] C. Lorenz, D. Hock, R. Durner, W. Kellerer, etal.: An SDN/NFV-enabled Enterprise Network Architecture Offering Fine-Grained Security Policy Enforcement. Accepted for IEEE ComMag, 2016.

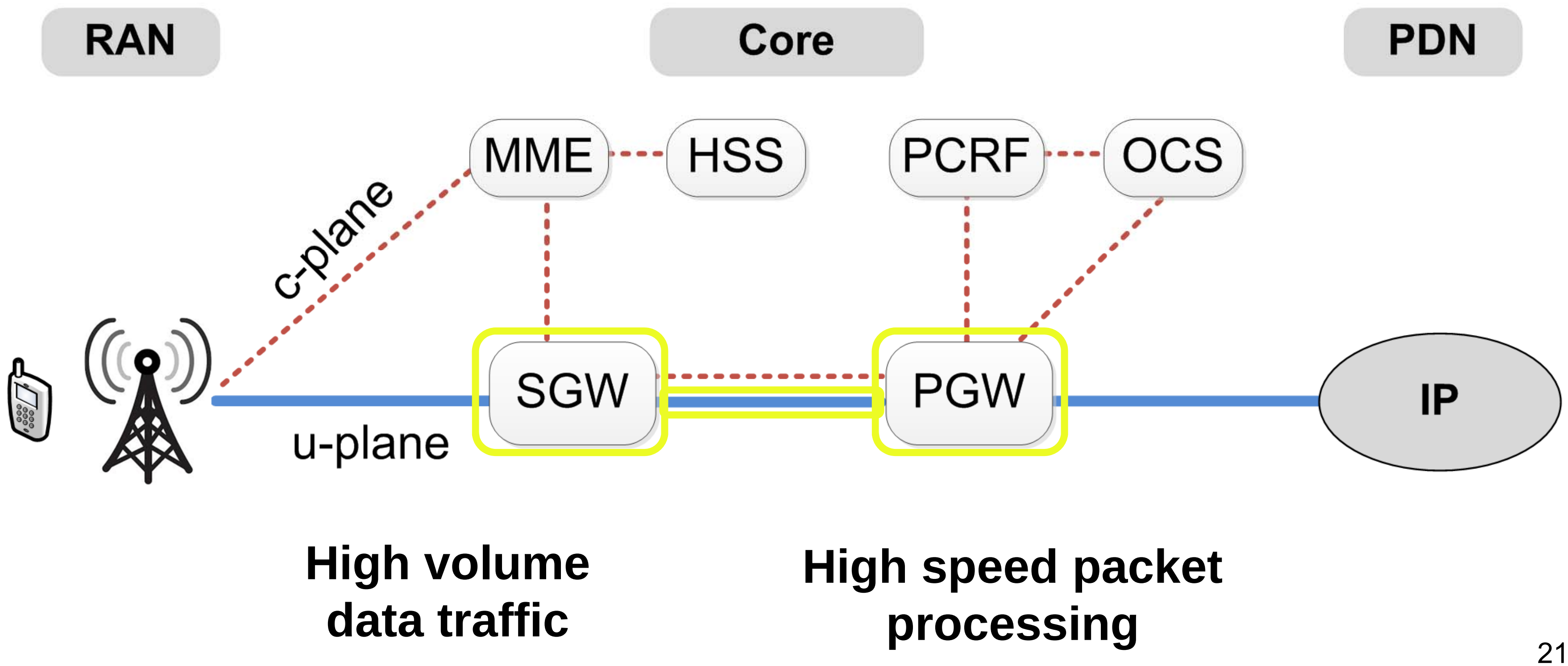
SarDiNe - Netzsicherheit in Unternehmen und Behörden basierend auf Software Defined Networking
(funded by the BMBF under grant number 16KIS0260)

Bundesministerium
für Bildung
und Forschung

Use Case: NFV + SDN/INP

- NFV = virtualize & move function (= everything) to DC
- Consider components/dependencies carefully: function decomposition and corresponding chain

Example: mobile core network functions



Function Realization based on NFV

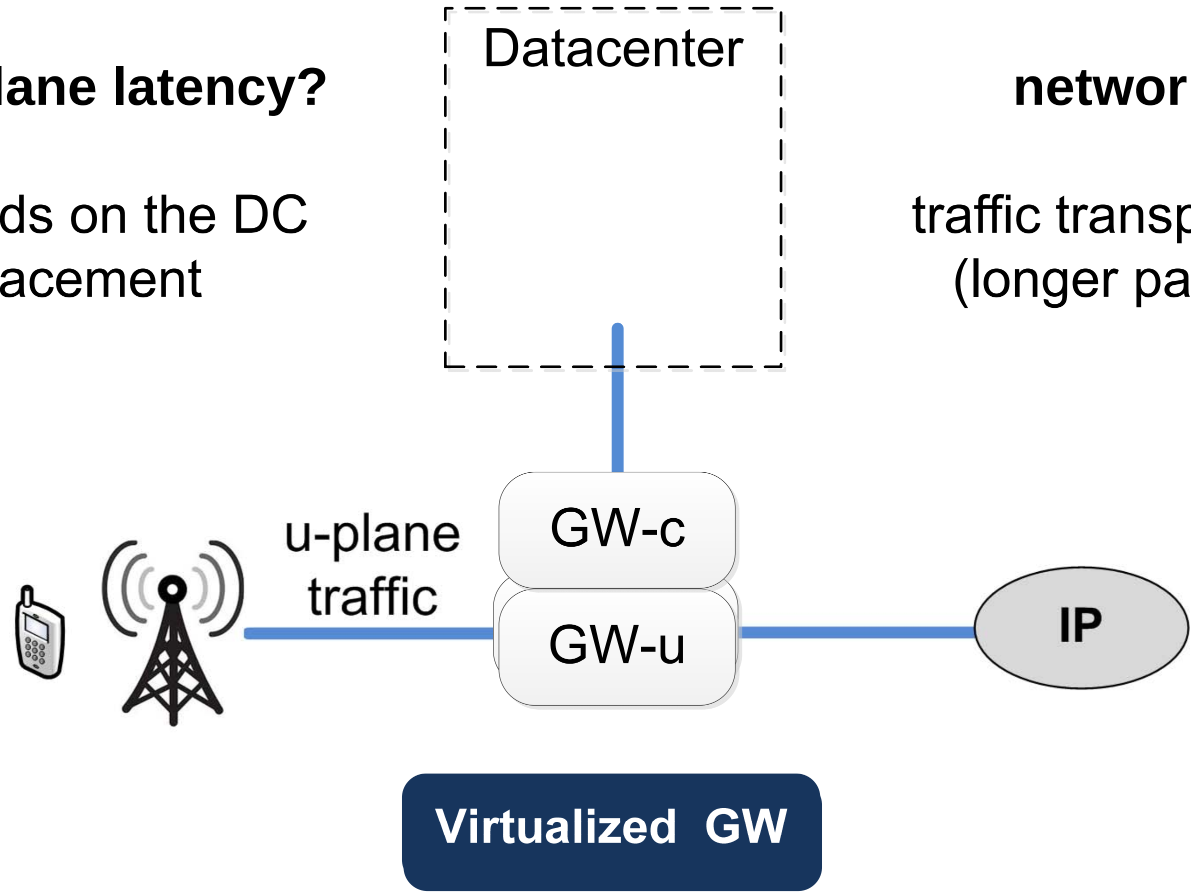
- Virtualization of GW functions [1] → NFV

data-plane latency?

depends on the DC placement

network load?

traffic transported to DC (longer path → cost)



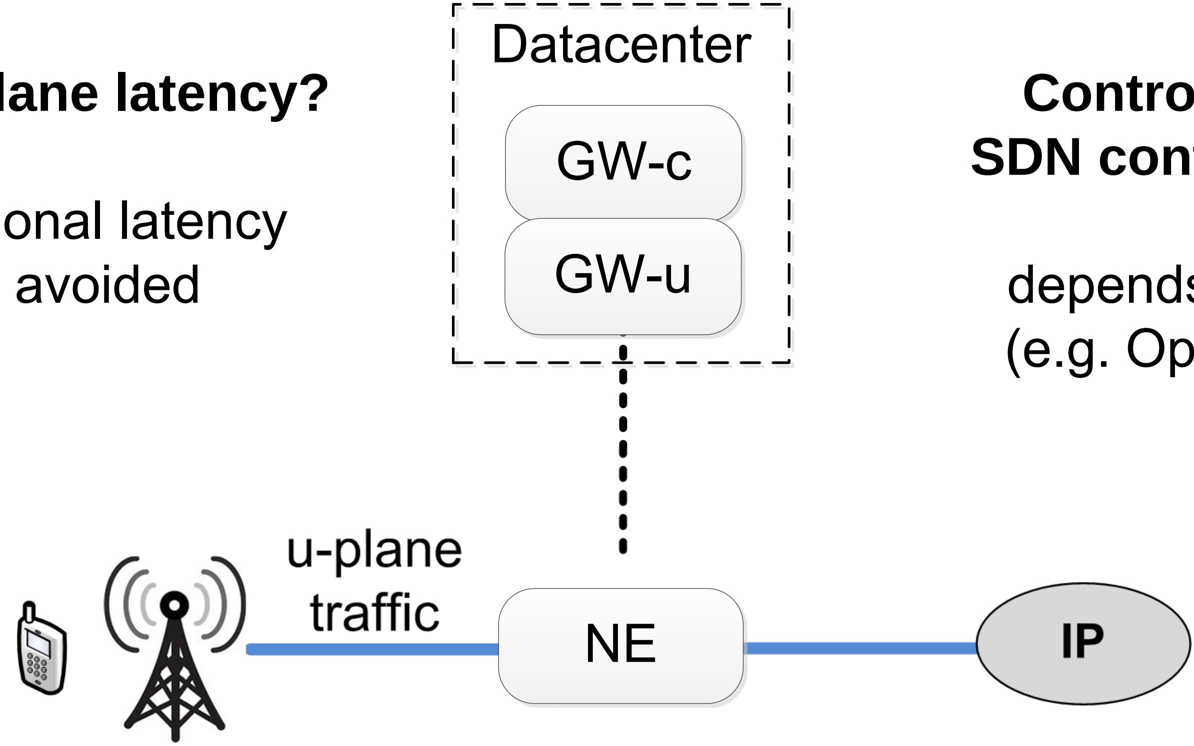
[1] A. Basta et al., A Virtual SDN-enabled EPC Architecture : a case study for S-/P-Gateways functions, SDN4FNS 2013.

Function Realization based on SDN/INP: move functions back

- Decomposition of GW functions [1] via SDN

data-plane latency?

additional latency
is avoided



Control load?
SDN control load!

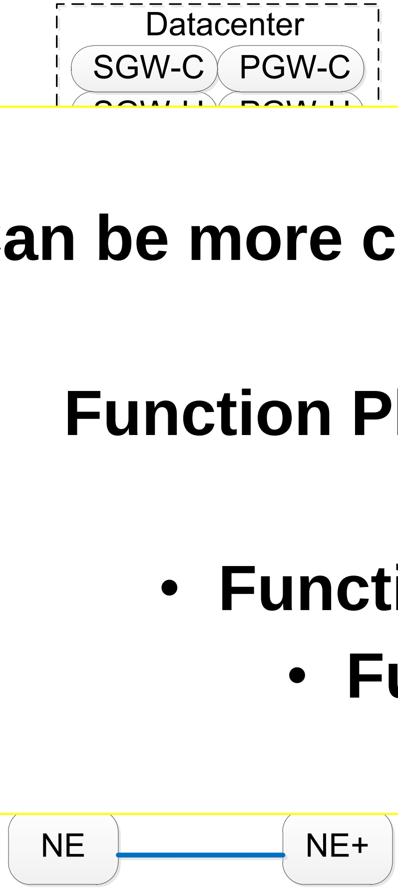
depends on API
(e.g. OpenFlow)

Decomposed GW

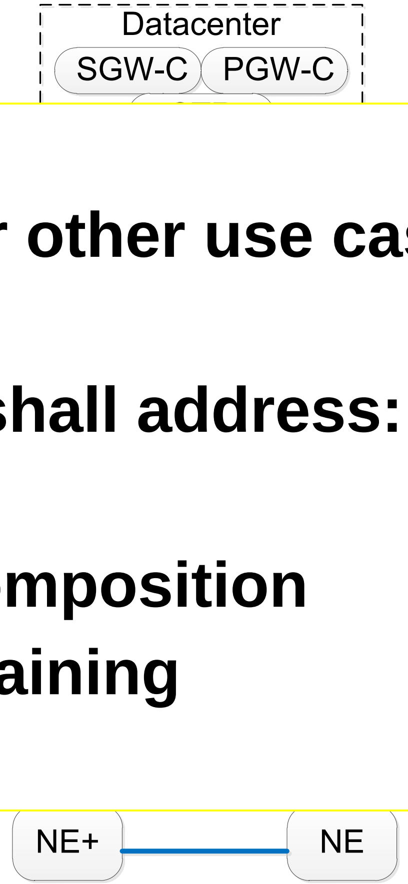
[1] A. Basta et al., A Virtual SDN-enabled EPC Architecture : a case study for S-/P-Gateways functions, SDN4FNS 2013.

Interdependencies → Function chains (mixed design)

- Propagation latency depends on function chain = path SGW - PGW



(c) SGW Virtualized
PGW Decomposed



(d) PGW Virtualized
SGW Decomposed

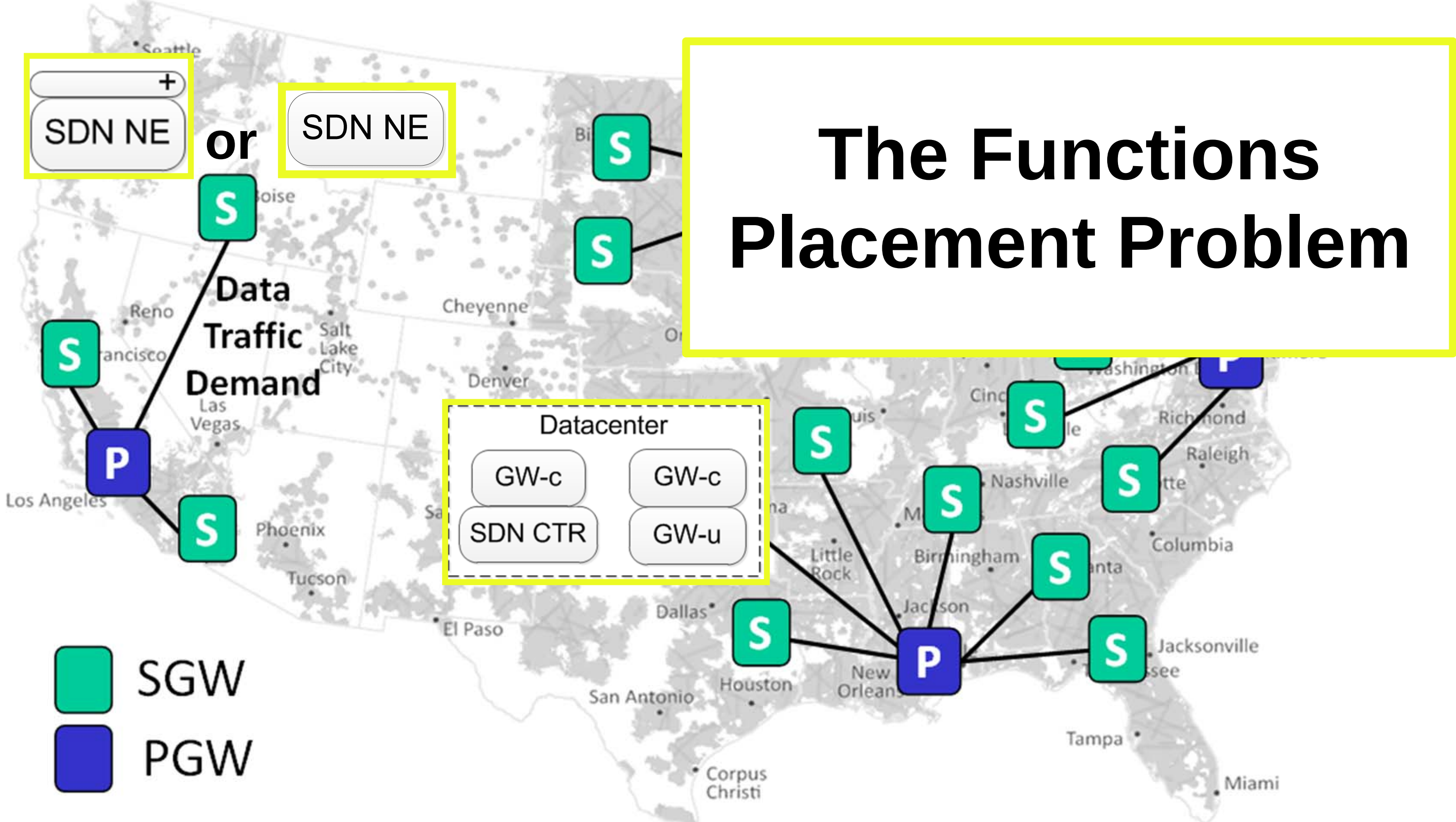
Can be more complex for other use cases

Function Placement shall address:

- **Function (de-)composition**
- **Function chaining**

Some Evaluation Studies

- Virtualize all GWs? decompose all? mixed deployment?
  - Which GWs should be virtualized? decomposed? DC(s) placement?
    - minimize core load
    - satisfy data-plane latency

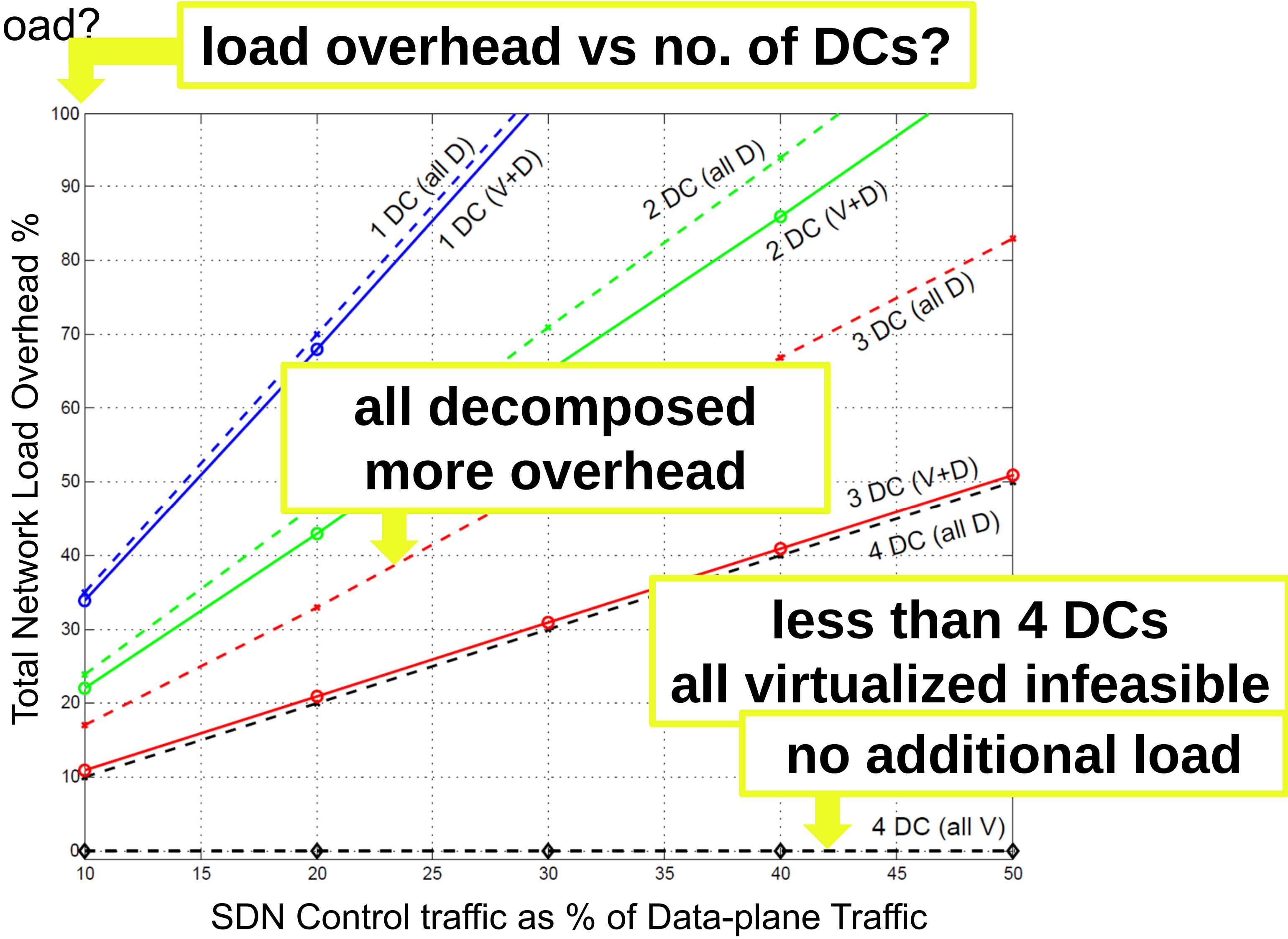


[2] A. Basta, W. Kellerer, M. Hoffmann, H. Morper, K. Hoffmann, Applying NFV and SDN to LTE Mobile Core Gateways; The Functions Placement Problem, AllThingsCellular14, Workshop ACM SICGOMM, Chicago, IL, USA, August 2014

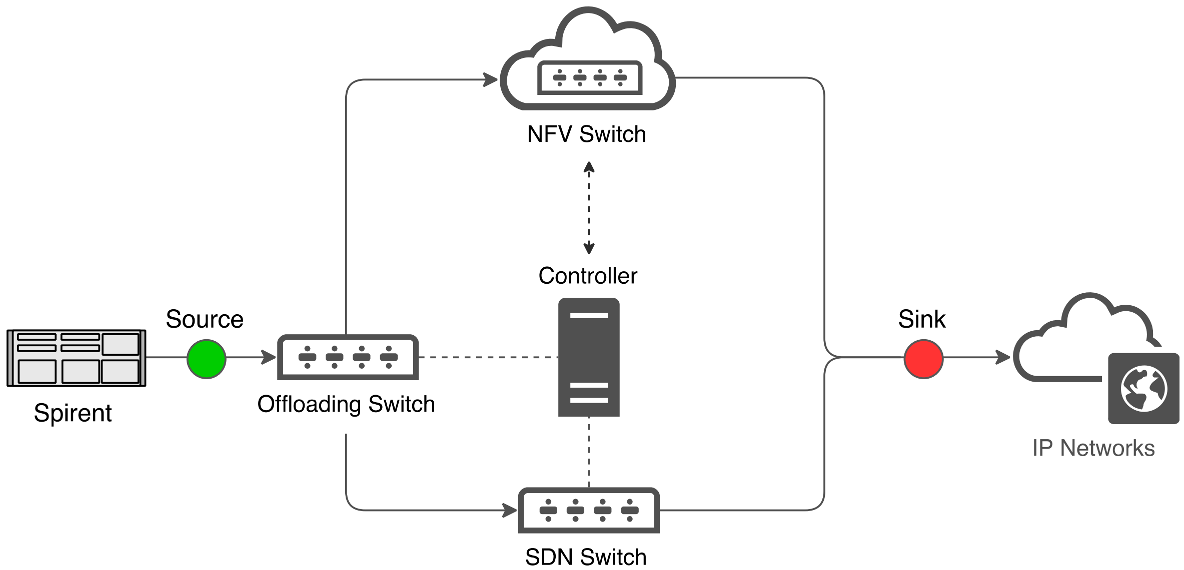
Evaluation

- Network load?

normalized by traffic in legacy topology



Use Case: Transition



Use Cases

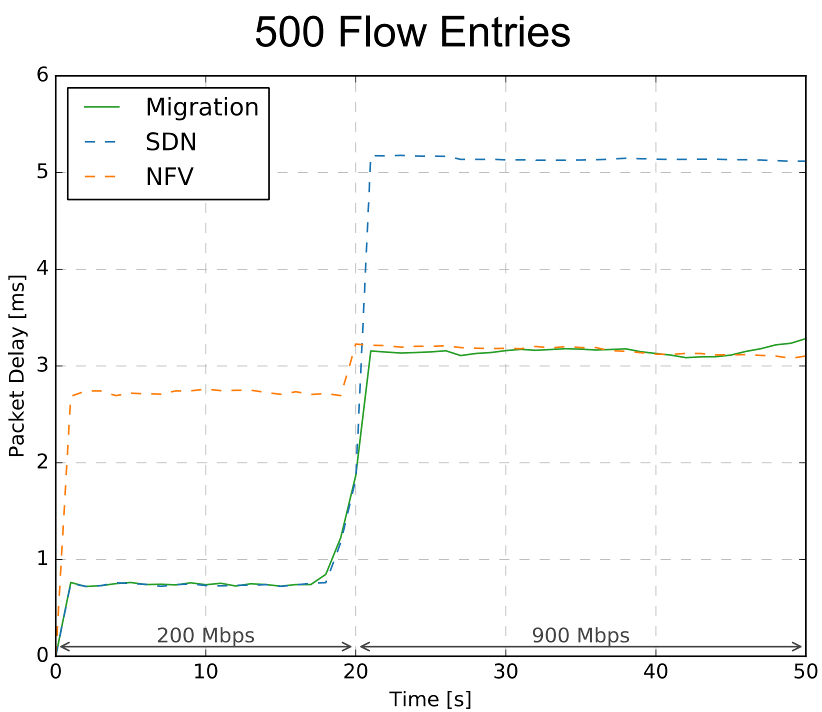
- Migration to NFV in case of intolerable traffic or SDN hardware failure
- Migration to SDN for critical time-based services

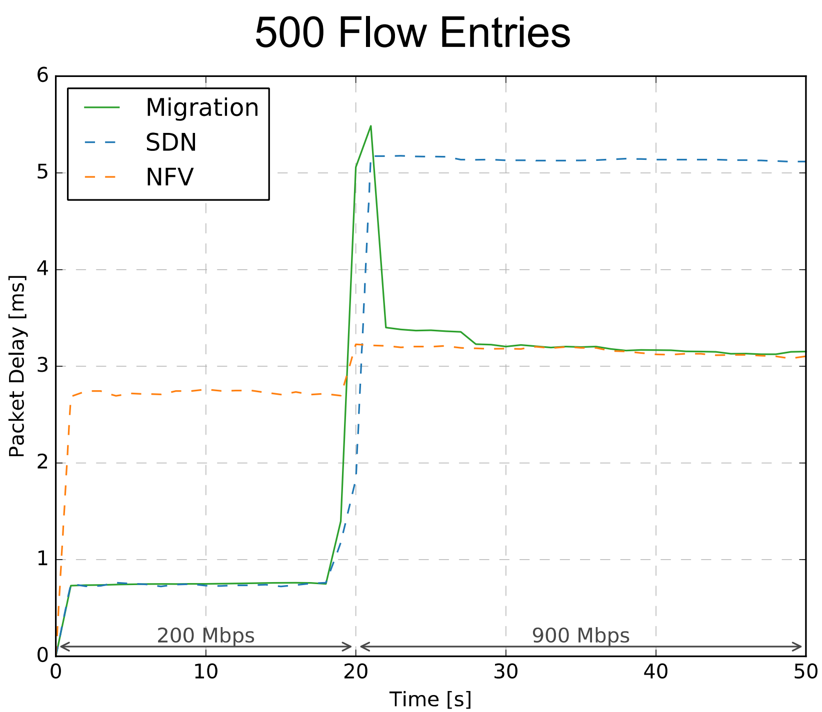
Hot-Standby Migration: rules synchronized periodically

Event-Triggered Migration: rules transfered on event at once

Transition between SDN/INP and NFV

Initial results





SDN → NFV
Hot-Standby-Migration

- SDN shows lower delay

SDN → NFV
Event Triggered Migration

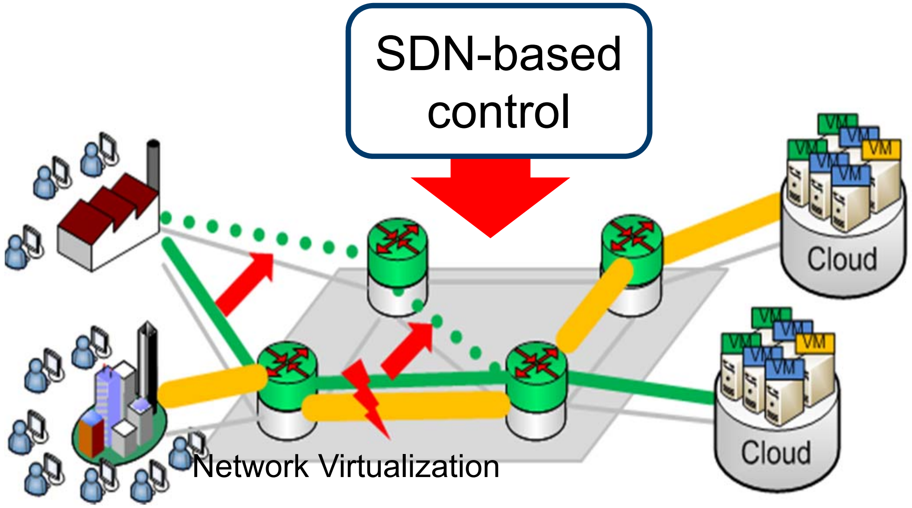
- extra time needed for the transition due to synchronization

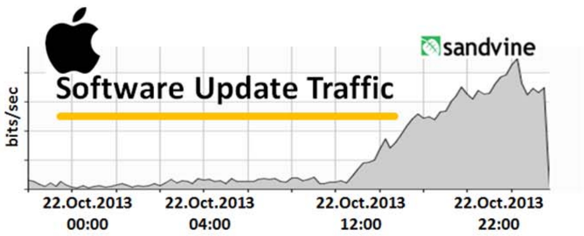
Recap: There are Multiple Mechanisms to Adapt the Künftige Internet

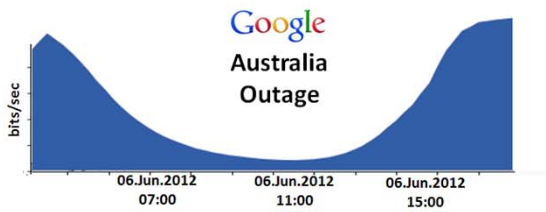
In particular, emerging concepts such as …

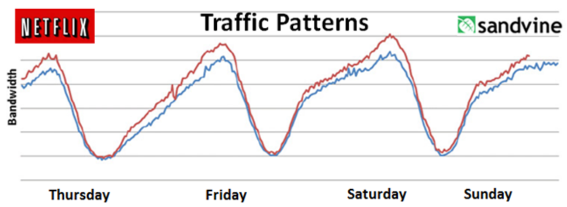
Software Defined Networking and Network Function Virtualization

… *promise* to adapt networks and functions on demand in software









All problems solved?

- Are we fully flexible already?

- How far can we go? What is the right network design?

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

For networks, **flexibility** = ability to *adapt* resources (flows, topology,…) *to change requests* of design requirements (traffic pattern, latencies,…)

A simple measure

For networks, **flexibility** = ability to *adapt* resources (flows, topology,…) *to change requests* of design requirements (traffic pattern, latencies,…)

e.g., *placement*

$$\varphi^{aspect}(S) = \frac{|supported\ requests|}{|all\ requests|}$$

- fraction of the number of **change requests** that can be supported of all requests
- w.r.t. to a certain **flexibility aspect** of a system S

- φ (S) ϵ [0,1] „percentage“

Use Case: EPC Function Placement

3 design choices to compare for future mobile core network [5]:
(1) SDN/INP design
(2) NFV design
(3) mixed SDN/NFV design

Parameter in focus:

- Flexibility to support different latency requirements for
  - control plane latency and data plane latency
    e.g.: {5, 10, 15,…, 45, 50} ms

all requests:
10 x10 =100

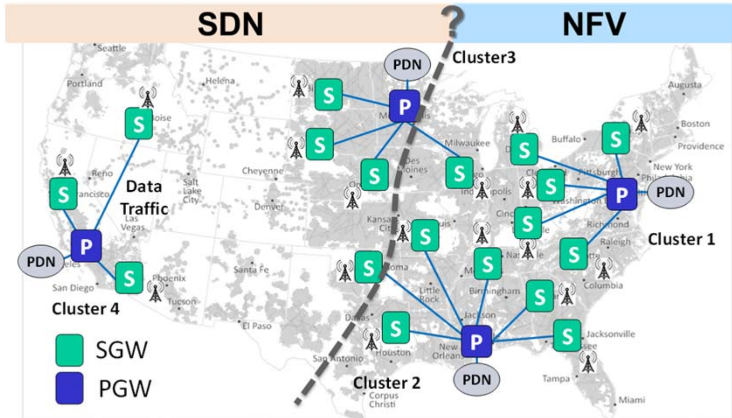
[5] W. Kellerer, A. Basta, A. Blenk,
Using a Flexibility Measure for Network Design Space Analysis of SDN and NFV, SWFAN'16,
IEEE INFOCOM Workshop, April 2016.

Flexibility measure and evaluation setup

Use Case

Flexibility measure:

$$\varphi^{placement}\ (design.x) = \frac{\left(\sum_i \sum_j feasibleSol_{i,j} \cdot w_{i,j}\right)}{\sum_i \sum_j w_{i,j}}$$



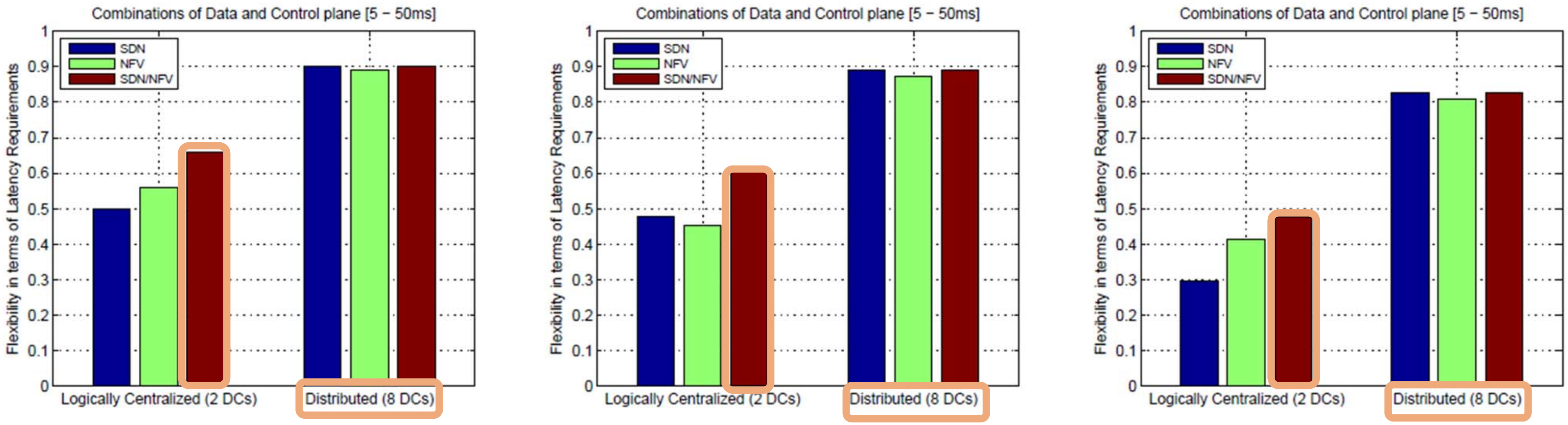
Function placement problem formulated as a MILP [6]

- SGW and PGW (VNF) placement
- constraints on data and control plane latency
- weights

$$w_{i,j} = \frac{\alpha}{dataLatency_i} + \frac{\beta}{controlLatency_j}$$

[6] A. Basta, W. Kellerer, M. Hoffmann, H. J. Morper, K. Hoffmann, Applying NFV and SDN to LTE mobile core gateways, the functions placement problem, All things cellular Workshop ACM SIGCOMM, Chicago, August, 2014.

Results [5]
Use Case



(a) equal weights for data and control latencies (b) weights biased by data latency (c) weights biased by control latency

With respect to the support of latency requirements in function placement:

- mixed SDN/NFV is more flexible for a logically centralized data center infrastructure
- for distributed data centers all three design choices are equally flexible

[5] W. Kellerer, A. Basta, A. Blenk, Using a Flexibility Measure for Network Design Space Analysis of SDN and NFV, SWFAN'16, IEEE INFOCOM Workshop, April 2016.

Something missing?
The time aspect of flexibility

"Heatposter" by Source. Licensed under Fair use via Wikipedia – http://en.wikipedia.org/wiki/File:Heatposter.jpg#/media/File:Heatposter.jpg

What Robert de Niro says on ***flexibility***

in HEAT (1995) as Neil McCauley:
"Don't get attached to anything you can't walk out on in 30 seconds flat if you feel the heat around the corner."

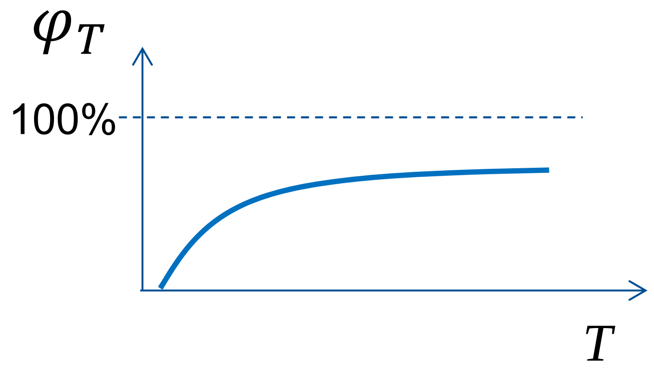
Not only the number of options, but the time matters for ***flexibility***

Quality of Flexibility – proposed definition

$$\varphi_T^{aspect}(S) = \frac{|supported\ requests\ fulfilled\ in\ T|}{|all\ requests|}$$

- fraction of the number of requests that can be supported in a time interval T of all requests
- T is small to capture system and request dynamics (sec to ms)

$$\varphi_{T->\propto}^{aspect}(S) = \frac{|supported\ requests|}{|all\ requests|}$$



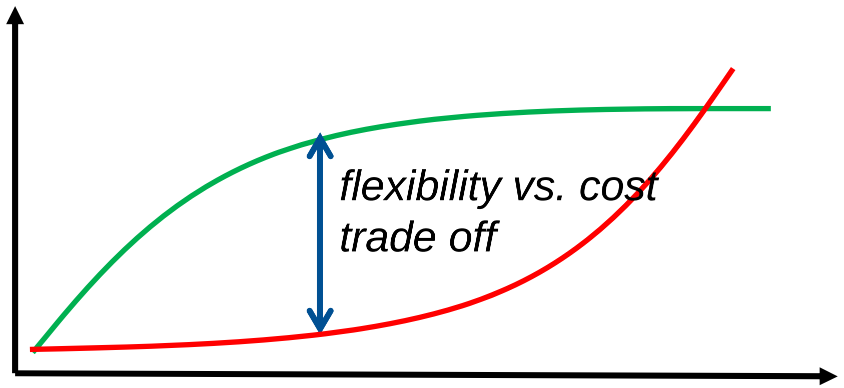
Nothing is for free: Cost of Flexibility

What are the costs of a design for flexibility?

- in terms of signaling overhead, number of data centers,…

Possible relationship (to be confirmed):

flexibility vs. cost



multidimensional **design space**

Use Case: Dynamic Controller Placement Problem

- Dynamic Controller Placement Problem:
  place 1 ..n SDN controllers for time varying input
  → controller migration/reconfiguration

- Evaluation parameters
  - Abilene network topology (11 nodes, 14 links)
  - 100 different flow profile requests over time (random)
  - N = 1,…, 4 controllers *(designs for comparison)*
  - Algorithm finds optimal controller placement and flow to controller assignment
  - How many controllers can be migrated (incl. control plane update) in time T? (success ratio → **Flexibility**)
  - Migrations and reconfigurations → **Cost**

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

M. He, A. Basta, A. Blenk, W. Kellerer, *Modeling Flow Setup Time for Controller Placement in SDN: Evaluation for Dynamic Flows,*
IEEE International Conference on Communications (ICC), Paris, France, May 2017.

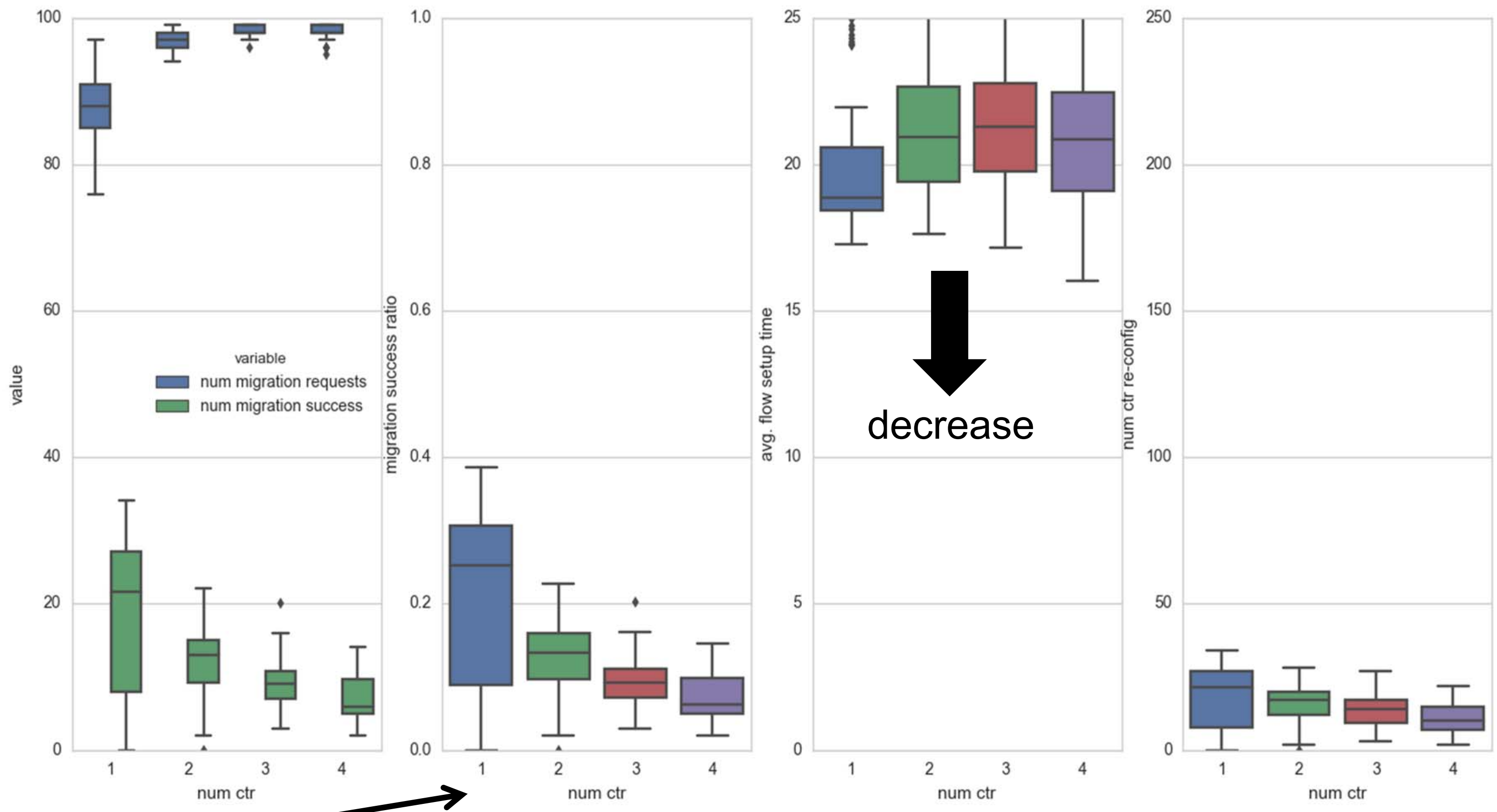
Simulation Results

Use Case

Flexibility — success ratio

Performance — avg. flow setup time

Cost — reconfigurations



Number of controllers N

migration time threshold = 803 ms

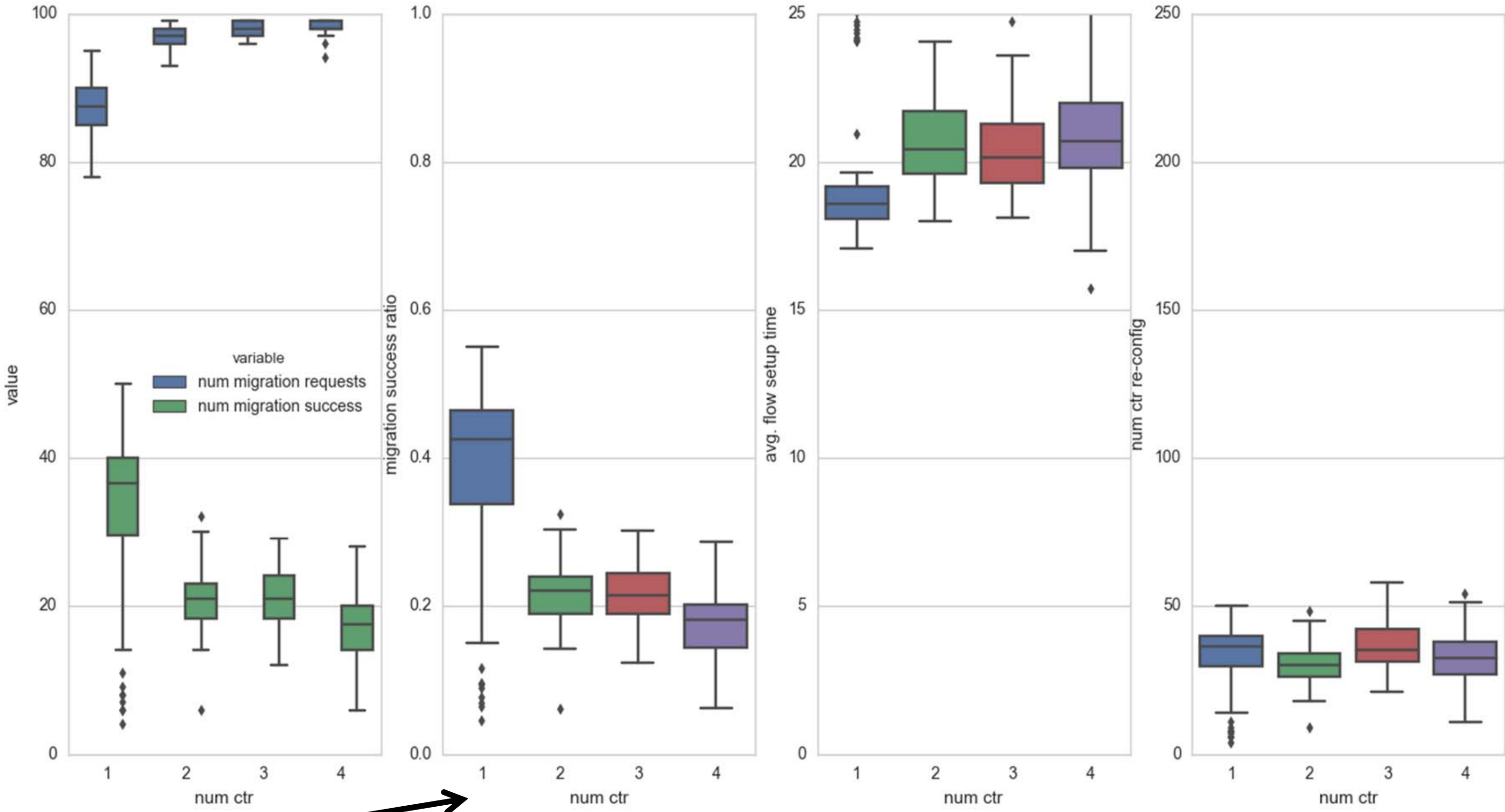
T is very short (800 ms is transmission delay of 1 controller)

Simulation Results

Use Case

Flexibility | Performance | Cost

success ratio | avg. flow setup time | reconfigurations



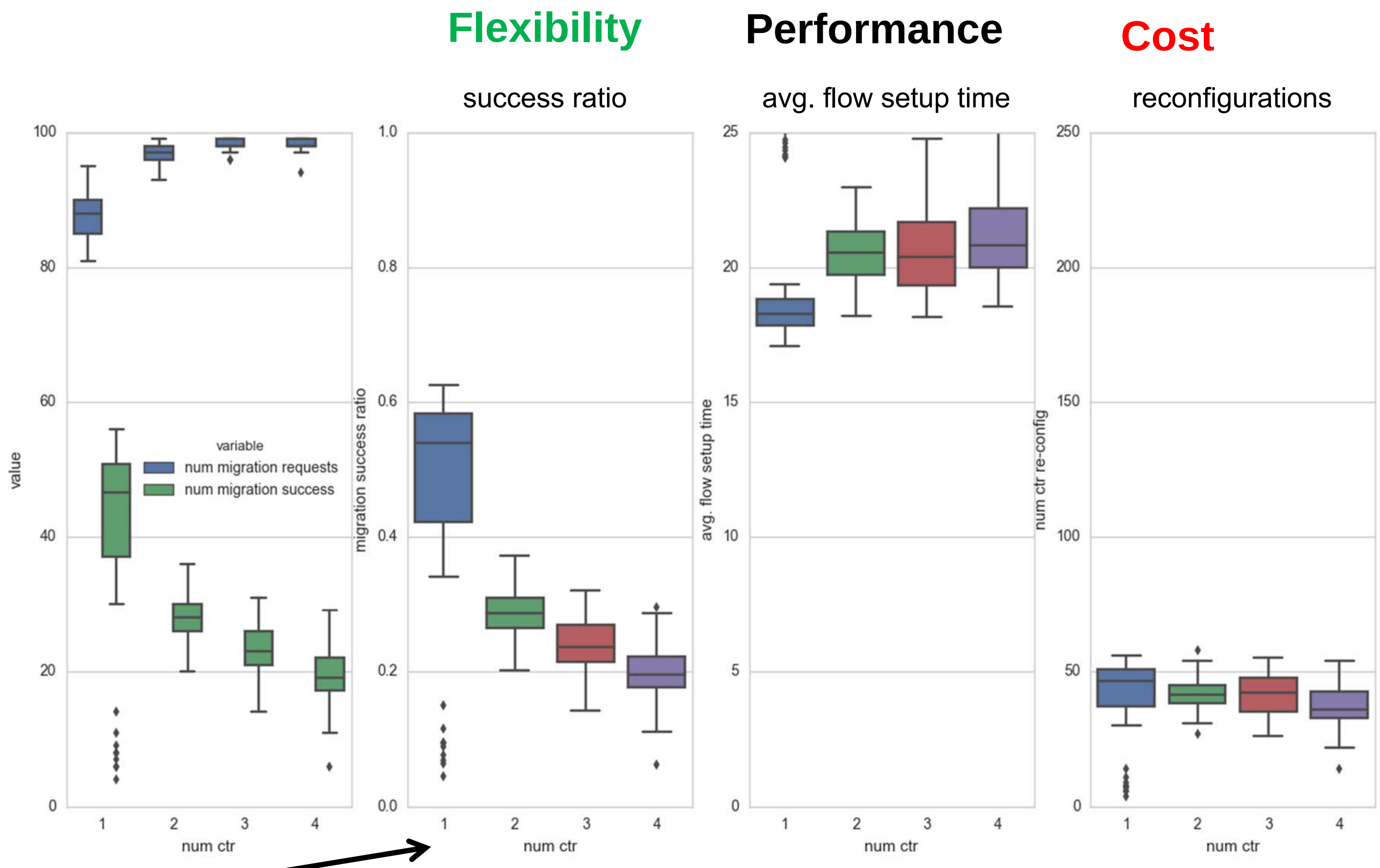
Number of controllers N

migration time threshold = 804 ms

T is very short (800 ms is transmission delay)

Simulation Results

Use Case

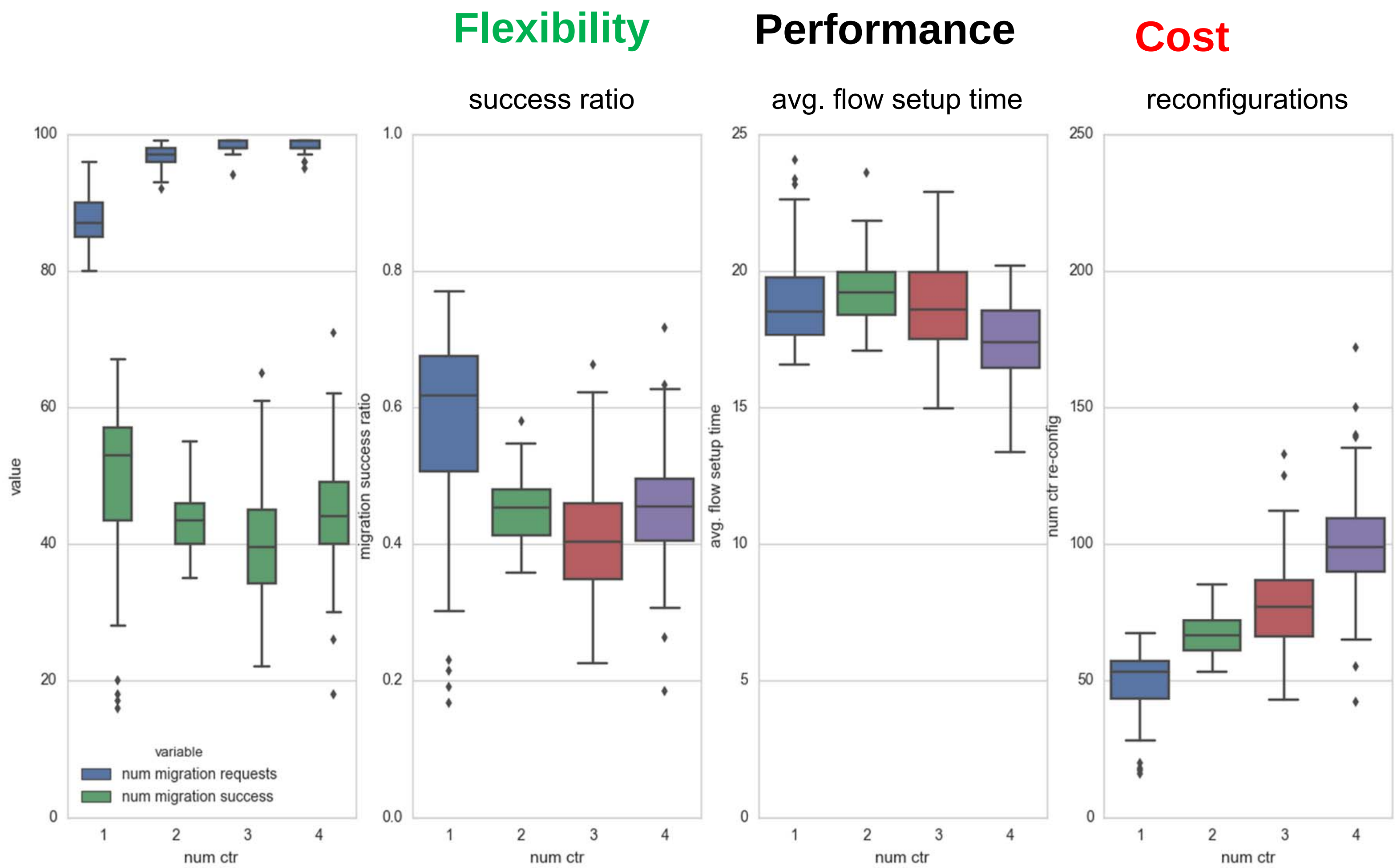


Number of controllers N

migration time threshold = 805 ms

Simulation Results

Use Case



migration time threshold = 806 ms

1 controller has highest flexibility at low cost

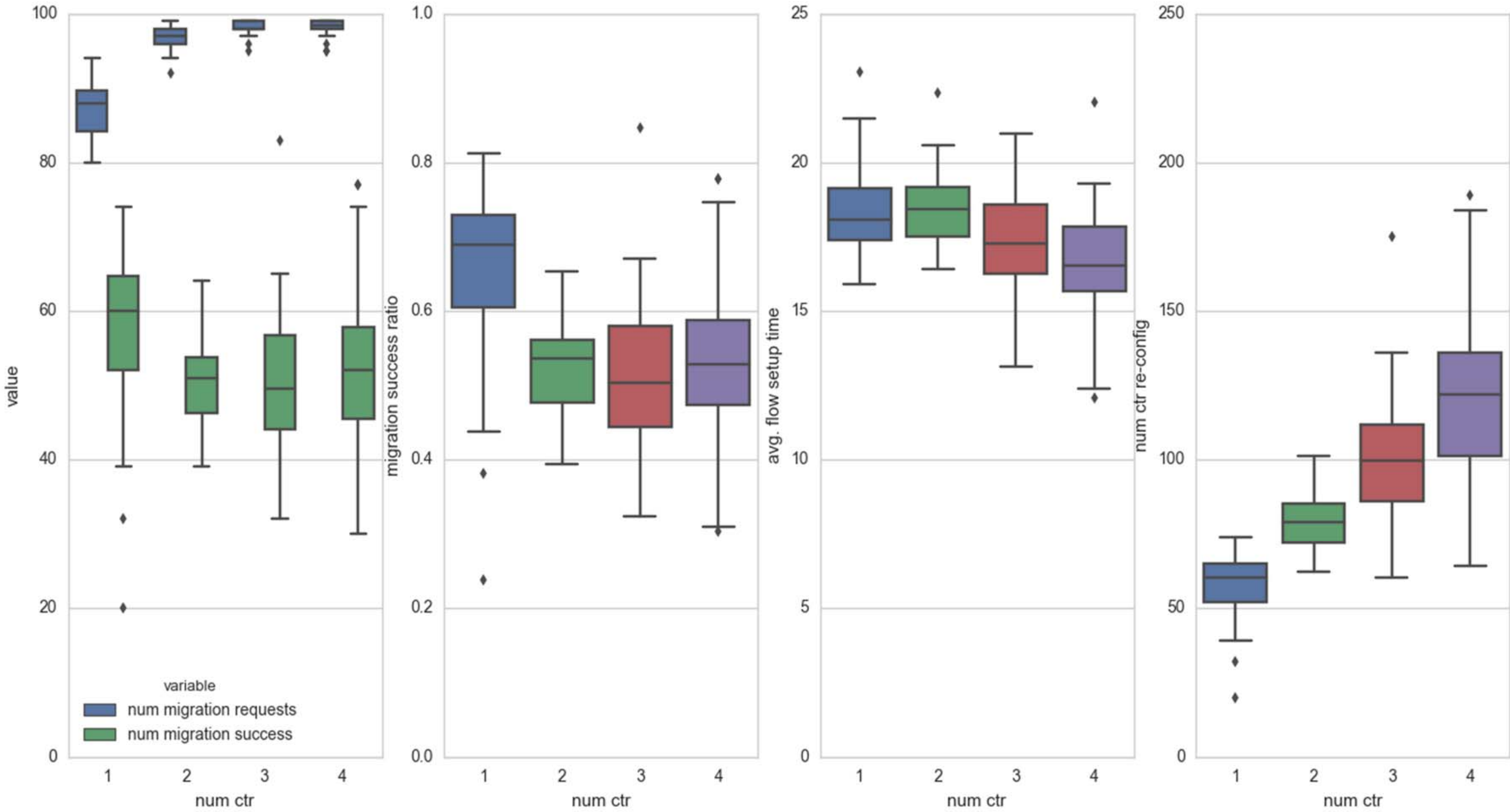
But: performance is not good (flow setup time)

Simulation Results

Use Case

Flexibility | Performance | Cost

success ratio | avg. flow setup time | reconfigurations



migration time threshold = 807 ms

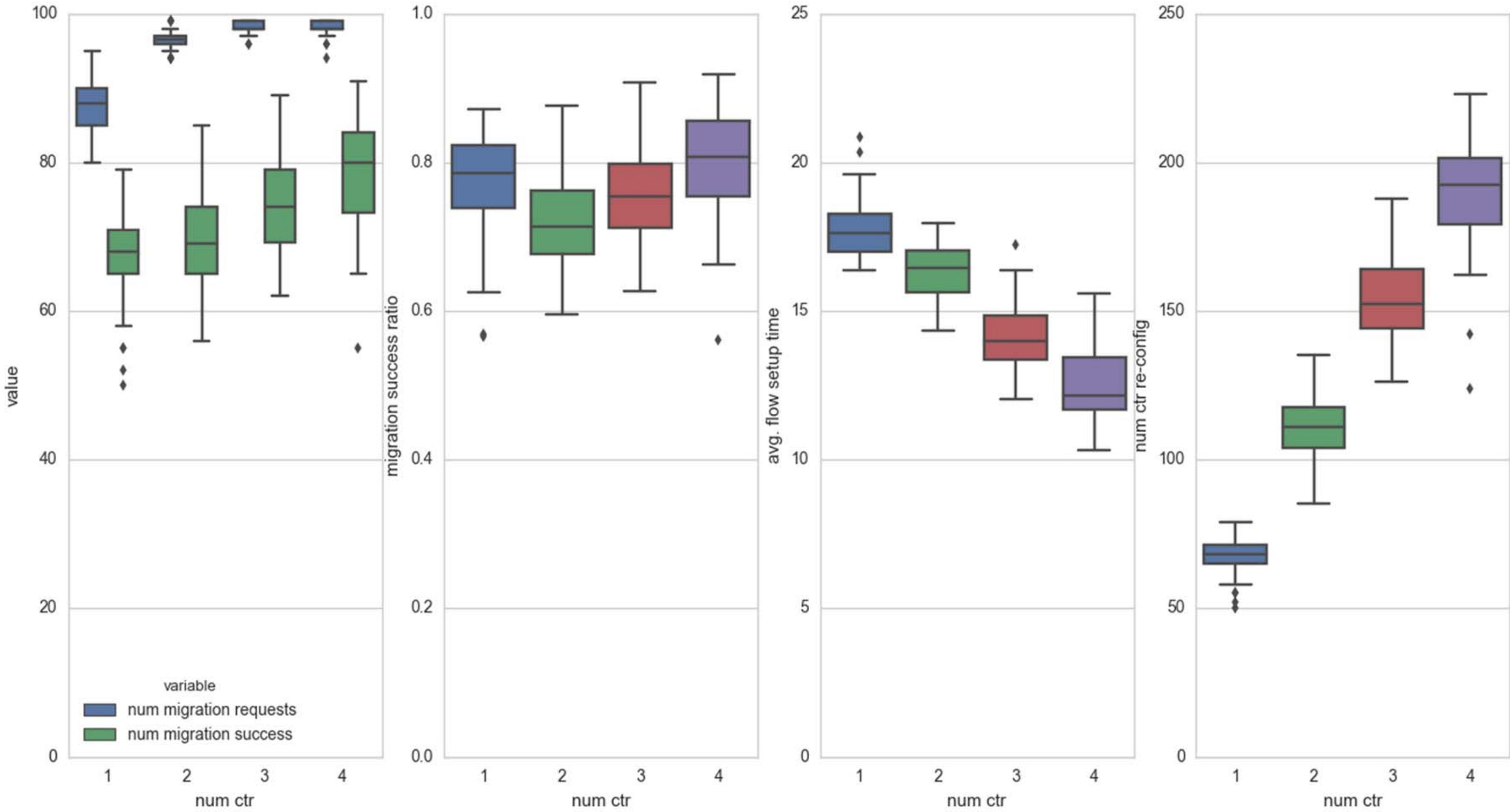
Simulation Results

Use Case

Flexibility — success ratio

Performance — avg. flow setup time

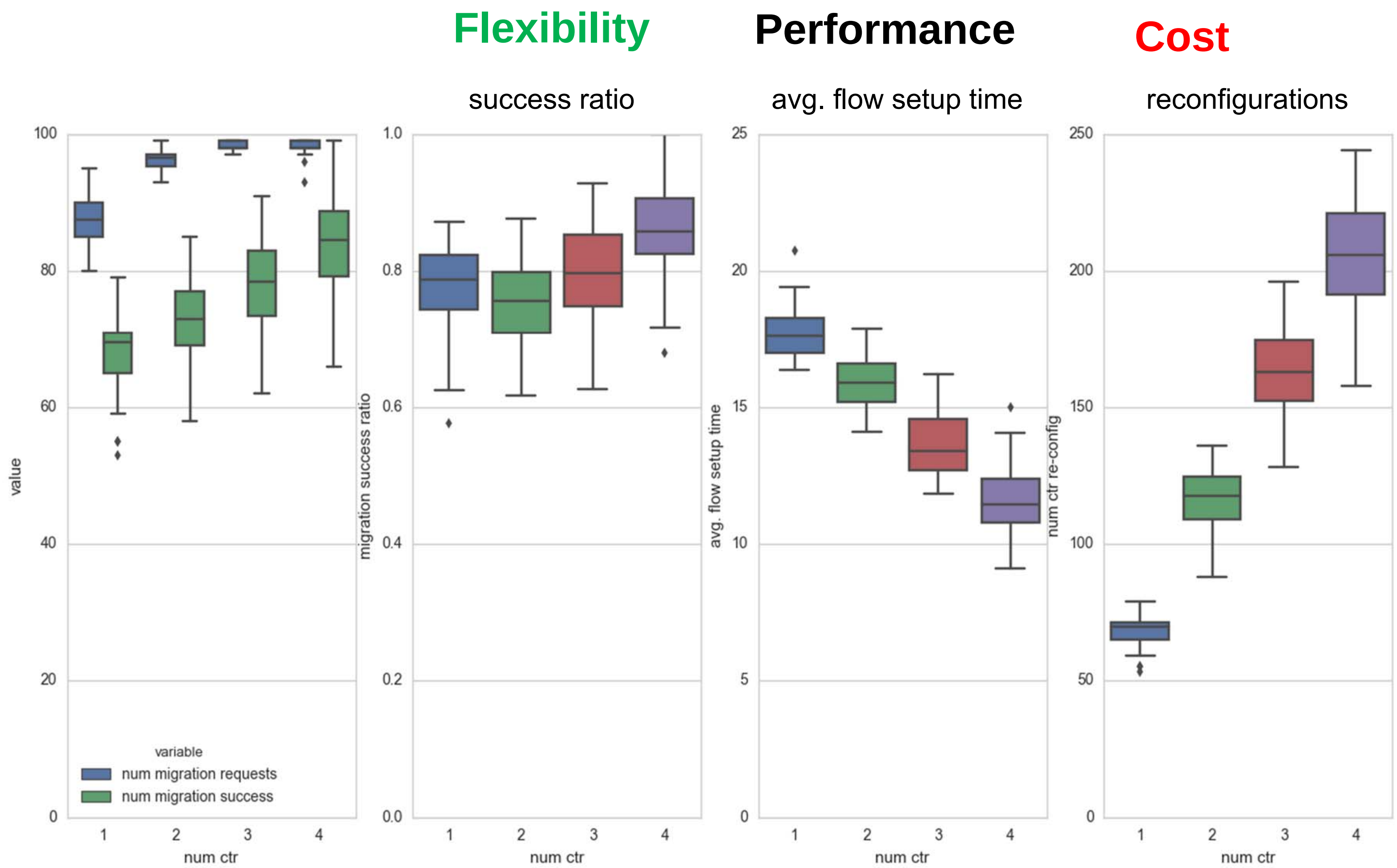
Cost — reconfigurations



migration time threshold = 808 ms

Simulation Results

Use Case



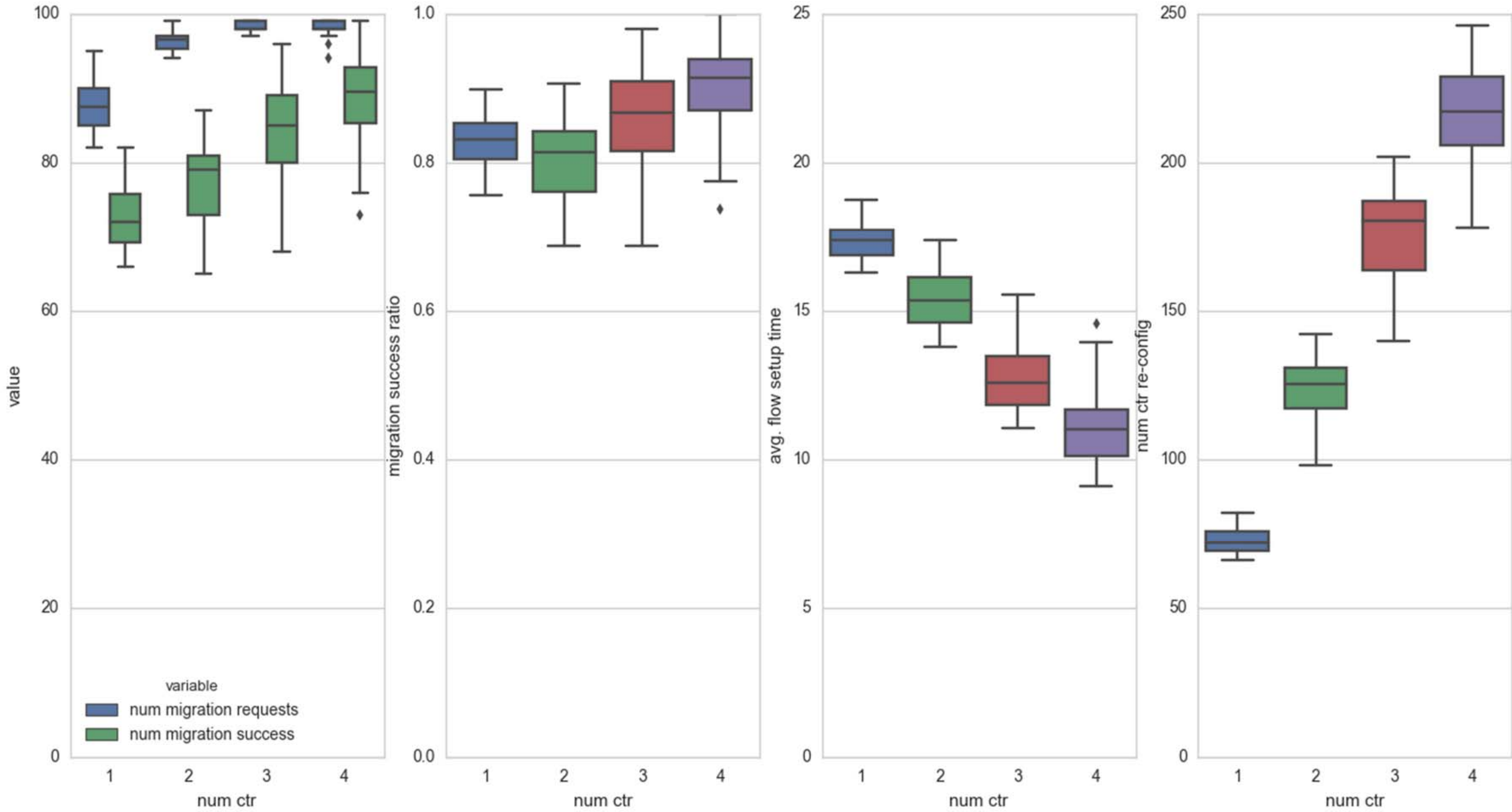
migration time threshold = 809 ms

Simulation Results

Use Case

Flexibility **Performance** **Cost**

success ratio avg. flow setup time reconfigurations

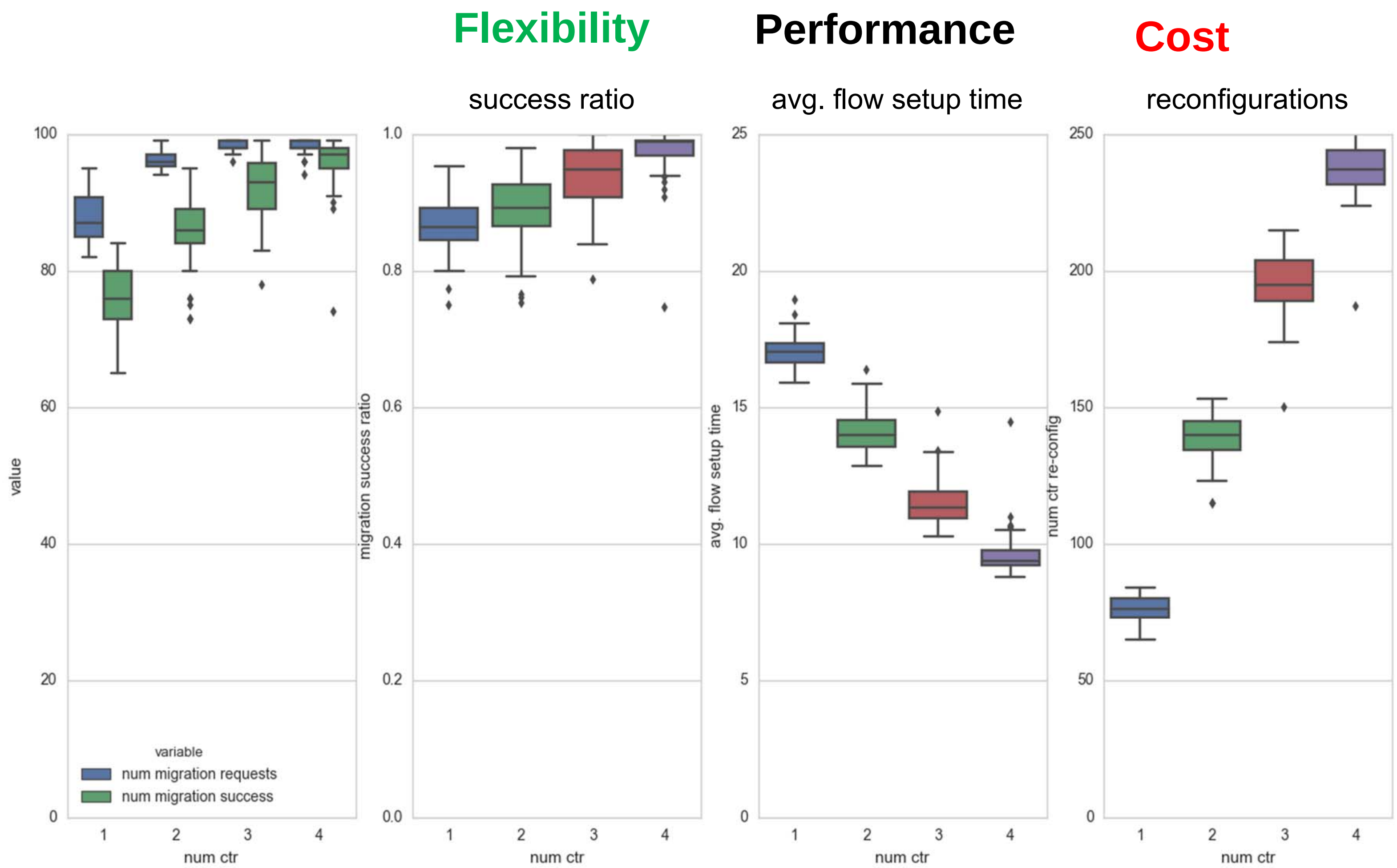


migration time threshold = 810 ms

T is moderate: more controllers → higher flexibility at higher cost

Simulation Results

Use Case



migration time threshold = 811 ms

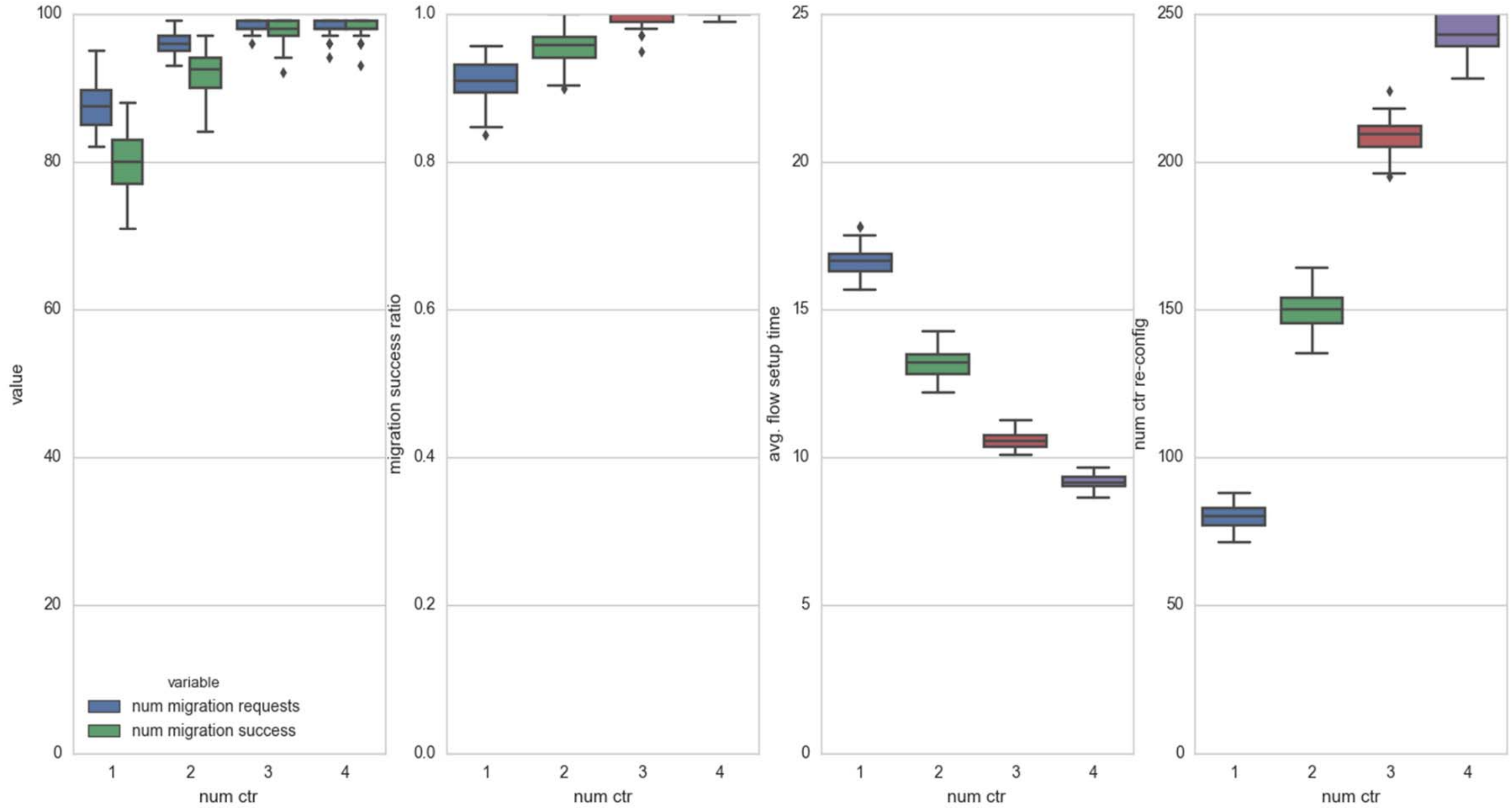
T is moderate: more controllers → higher flexibility at higher cost

Simulation Results

Use Case

Flexibility **Performance** **Cost**

success ratio avg. flow setup time reconfigurations

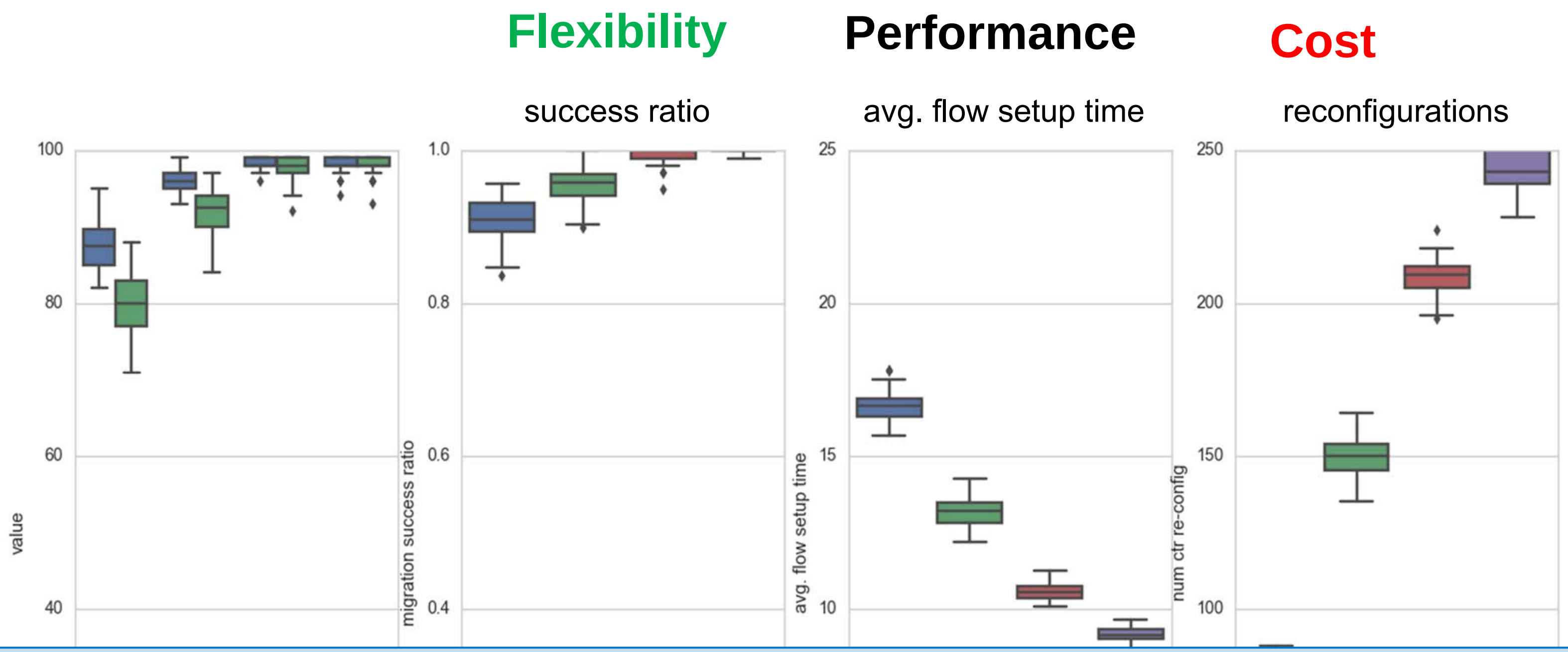


migration time threshold = 812 ms

T is moderate: more controllers → higher flexibility at higher cost

Interpretation

Use Case



- Some cases: 1 controller is more flexible (short T)
- T considerable for adaptation: more controllers → more flexible
- There is a cap in gain – cost is rising

Key Takeaways

- NFV + SDN + INP provide an excellent basis for adaptation
- Network functions
  - **(De-)compose and chain functions with care**
- **Consider Dynamics** – time matters

Most important:

- **Flexibility** as a new measure for analysis

References for further reading

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

- M. He, A. Basta, A. Blenk, W. Kellerer, *Modeling Flow Setup Time for Controller Placement in SDN: Evaluation for Dynamic Flows,* IEEE International Conference on Communications (ICC), Paris, France, 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, San Francisco, USA, April 2016.

- A. Basta, W. Kellerer, M. Hoffmann, H. Morper, K. Hoffmann, *Applying NFV and SDN to LTE Mobile Core Gateways; The Functions Placement Problem*, AllThingsCellular14, Workshop ACM SICGOMM, Chicago, IL, USA, August 2014.

- A. Basta, A. Blenk, M. Hoffmann, H. Morper, K. Hoffmann, W. Kellerer, *SDN and NFV Dynamic Operation of LTE EPC Gateways for Time-varying Traffic Patterns,* 6th International Conference on Mobile Networks and Management (MONAMI), Würzburg, Germany, September 2014.

- W. Kellerer, A. Basta, A. Blenk, *Flexibility of Networks: a new measure for network design space analysis?,* arXive report, December 2015. http://www.lkn.ei.tum.de/forschung/publikationen/dateien/Kellerer2015FlexibilityofNetworks:a.pdf