Fluid Network Planes
An Overview of Ongoing Network Softwarization Refactoring Trends

Prof. Dr. Christian Esteve Rothenberg
(University of Campinas), Brazil
chesteve@dca.fee.unicamp.br

https://intrig.dca.fee.unicamp.br/christian
http://www.dca.fee.unicamp.br/~chesteve/
Agenda

- A view on 10+ years of SDN
- Fluid Network Planes
  - The ‘Concept’
  - Instances

Disclaimer

The ‘origins’ of the SDN term

TR10: Software-Defined Networking

Nick McKeown believes that remotely controlling network hardware with software can bring the Internet up to speed.

4 comments
KATE GREENE
March/April 2009

For years, computer scientists have dreamed up ways to improve networks’ speed, reliability, energy efficiency, and security. But their schemes have generally remained lab projects, because it’s been impossible to test them on a large enough scale to see if they’d work: the routers and switches at the core of the Internet are locked down, their software the intellectual property of companies such as Cisco and Hewlett-Packard.
SDN in 2009 - 2010
Canonical/Open SDN

Control-plane component(s)

Data-plane component(s)

OSS/BSS

Mgm. Apps

Business / Control Apps

Northbound APIs

Network Controller / OS

Southbound Protocol (e.g. OF)

Mgm.

Southbound Agent (e.g. OF)

HAL APIs / Drivers

Data Plane

Source: C. Rothenberg (INTRIG/UNICAMP)
SDN in 2011 – 2012
SDN to the rescue!

Software Defined Networking

Warning: Contains optimism
(Plug to http://PacketPushers.net for Unicorn Humor!)
So, what is **SDN**?

“*OpenFlow is SDN, but SDN is not OpenFlow*”

(does not say much about SDN)  — Networking community

“*Don’t let humans do machines’ work*”

(probably right...)  — Networking Professional

“*Let’s call SDN whatever we can ship today*”

(aka ‘SDN washing’)  — Vendor X

“*SDN is the magic buzzword that will bring us VC funding*”

(hmmm... N/A, N/C)  — Startup Y

“*SDN is the magic that will get my paper/grant accepted*”

(maybe, but not at Tier-1 Conferences / Journals!)  — Researcher Z
“Will OpenFlow commoditize networks? Impact Cisco margins?”
—Several media publications, Bloggers

“Prediction: OpenFlow Is Dead by 2014; SDN Reborn in Network Management”
—Mike Fratto, Network Computing

“We share a more pragmatic view, noting Cisco (for example) is likely to view SDN as a TAM expansion opportunity…”
—Deutsche Bank Research note, Wired, April 2012

“SDN - Still Does Nothing”
“SDN - Smells Dollars Now”
“SDN - Software Defined Not-working”

“SDN needs a bigger definition”
—Lippis report, 2012

Source: Adapted from A. Retana @ Lacnog’12
SDN in 2013 - 2015

Vendor A
Vendor B
Vendor C

Academia
Start-up 1
Start-up 2
...
Start-up n
SDN in 2015 – 2020 → Network Softwarization (i.e. NFV + SDN + IBN + xyz)

Old / Existing
- CLIs & Manual labour
- Closed Source
- Vendor Lead
- Classic Network Appliances (HW)

New / Softwarized
- APIs & Automation
- Open Source
- Customer Lead
- Virtual Network Functions (NFV/SW)

Source: Adapted from Kyle Mestery, Next Generation Network Developer Skills
Different Network Softwarization Models

- Canonical/Open
- Compiler
- Hybrid/Broker
- Overlay
- Traditional
- Whitebox / Baremetal + PISA / P4

Source: C. Rothenberg (INTRIG/UNICAMP)
Models & Approaches to Program / Refactor the Netsoft Stack

Orchestrator (SO/RO/LCM)

VNF
(Manager)

VIM
(Infra-M)

Legacy

Management Software

Network Controller / OS

Orchestrator (SO/RO/LCM)

VNF
(Manager)

VIM
(Infra-M)

Legacy

Management Software

Network Controller / OS

SDN

Program

Compiler

Bare Metal Switches

Network OS

Virtualization

HW Resources

GP-CPU
(x86, ARM)

Source: C. Rothenberg (INTRIG/UNICAMP)
Network programmability? By whom?
Technical Expertise + Single Throat to Choke

**Players** with sufficient SW Eng. + Network Eng. & in-house Devops (NoOps?)

- Intent-based (languages + APIs)
- Design + Run-time (NS)DKs
- ML/AI assistance
- Automation of Test + Benchmarking (pre-deployment + + day0 & day-2 ops)

The **long tail** of players (e.g. smaller SPs, ISPs, enterprises, campus, governments, etc.)
The Fluid Networking landscape
The **Fluid Networking** landscape
The **Fluid Networking** landscape
Fluid Networking: HW-SW Continuum

- Performance
- Portability
- Programmability

Source: D. Meyer (Courtesy by J. Doyle)

Source: G. Pongracz. "Cheap silicon". HotSDN 13

Source: C. Rothenberg. P3 Trade-offs. 2017
Fluid Networking: HW-SW Continuum

**Flexibility**
(programmability + portability)

- Containers
- User space
- Kernel space
- Drivers, I/O SDKs
- General-purpose CPU
- HW-accelerated features**
- FPGA
- GPU, TPU,
- Programmable NIC, ASIC
- Domain Specific Architectures (DSAS) e.g., P4 + PISA

**Performance**

---

* M. He et al. *Flexibility in Softwarized Networks: Classifications and Research Challenges*. IEEE Survey & Tutorials, 2019


Fluid Networking: Quest for Latency / Fog & Cloud Continuum

- 15 Data centers
- 100 Points of Presence (PoPs)
- 1000+ Edge nodes

Source: Google Cloud Infrastructure
Fluid Networking: Optimizing the E2E Compute Pool

Service related functionality

Optimizing distribution of functions & state

Distribution of service functionality

Source: EU FP7 UNIFY
Fluid Networking: **Decoupling functionality / location**

[Diagram showing the relationship between various components such as Access Network, Aggregation Network, Core Network, microserver platform, micro-DC platform, PoP, Data center, and Cloud DCs.]

Source: EU Superfluidity
The **Fluid Networking** landscape

- **Control plane component(s)**
- **Data plane component(s)**

Optimize for

- **Performance/Cost**
- **Latency** (Latency-sensitive Source to Function)

---

**Customer Premises**  
**BS MEC**  
**Access Cloud**  
**Edge**  
**PoP DC**  
**Cloud DCs Core**
The **Fluid Networking** landscape
The **Fluid Networking** landscape
The **Fluid Networking** landscape

- Customer Premises
- BS MEC - Access Cloud Edge
- PoP DC
- Cloud DCs Core

HW

SW
The **Fluid Networking** landscape

- The Fluid Networking landscape is a diagram that illustrates the various components and levels of networking, from HW (hardware) to SW (software), across different premises and cloud DCs (data centers).
- Key components include:
  - Customer Premises
  - BS MEC (Base Station Multi-Access Edge Computing)
  - Access Cloud
  - Edge Cloud DC
  - PoP DC (Point of Presence Data Center)
  - Cloud DCs Core

- The diagram uses arrows and boxes to show the flow and connections between these components, indicating the fluid nature of networking in modern environments.
- The presence of a poison bottle suggests a cautionary note or a risk factor in the networking landscape.
The **Fluid Networking** landscape
Instances of Fluid Network Planes
RouteFlow (2010 - )

Customer Premises - BS MEC - Access Cloud Edge - PoP DC - Cloud DCs Core

HW

SW

IXP

BGP

legacy
NFV layers of SW, Virtualization and HW platforms

VNF offloading to Hardware

VNF offloading to Hardware

VNF offloading on multi-vendor P4 fabric controlled by ONOS via P4Runtime

Source: https://p4.org/assets/P4WS_2018/7_Carmelo_Cascone_VNF.pdf
Related work at TUD and UNICAMP


Slicing

Cloud – Slice Orchestr & Management System

Network – Slice Orchestr & Management System

Access Cloud

uRLLC GW
eMBB GW

vCDN

Edge Cloud

Central Cloud

5G CP

mMTC

vCDN

Slice 1: 100M, eMBB
Slice 2: 1G, Video
Slice 3: 90G Other Slices

Slice 1: 100M, eMBB
Slice 2: 1G, Video
Slice 3: 90G Other Slices

Slice 4: mIoT – latency sensitive
Slice 5: mIoT – computing intensive

IoT Analytics

Source: http://www.h2020-necos.eu/
Slicing

- **Resources (incl. NFs)** need to be allocated for the new situation.
- **Proper Control and Mgmt Interfaces** offered by the remote domains.

**Network Function**

Network Provider 0

NFV Infrastructure PoP Provider 0

Network Provider 1

Network Provider 2

Opportunity for instantiating NFs in proximity
Better service fit

User demand changes
(maybe unexpectedly or bursty)

Multi-Domain
Administrative & Technological

Protocol stack
Choice & Control

Isolation
under massive multi-tenancy

Source: Adapted from slide courtesy by Luis M. Contreras, Telefonica. http://www.h2020-necos.eu/
Control Plane functions (BGP) offloading to HW

E. Costa Molero et al. **Hardware-Accelerated Network Control Planes.** HotNets’18

T. Holterbach et al. **Blink: Fast Connectivity Recovery Entirely in the Data Plane.** NSDI’19
Computation in the Network

* D. Ports and J. Nelson. When Should The Network Be The Computer?. HotOS'19

IRTF Computation in the Network (COIN)
J. Vestin et al. *In-Network Control and Caching for Industrial Control Networks using Programmable Data Planes*. 2018
X. Jin et al. **Netcache: Balancing key-value stores with fast in-network caching.** SOSP’17
**P4xos:** Time to reach consensus: RTT/2

**Paxos:** Time to reach consensus: RTT x 3/2

H. Tu Dang et al. *P4xos: Consensus as a Network Service.* 2018
SwitchML: the network is the ML accelerator

A. Sapio et al. Scaling Distributed Machine Learning with In-Network Aggregation. 2019
Fig. 1: Architecture of the preprocessing: A linear regression learns its parameters from historic values of a forecast and pushes them to the data-plane elements where they are used to estimate the importance of measurements and filter irrelevant measurements.

Fig. 3: Abstract switch architecture divided in control and forwarding layer. The control layer serves as interface to the control-plane and maintains that communication. The forwarding layer behaves according to the installed P4 program whenever packets arrive.
Xiaoqi Chen, Hyojoon Kim, Javed M. Aman, Willie Chang, Mack Lee, and Jennifer Rexford. **Measuring TCP Round-Trip Time in the Data Plane.** In Workshop on Secure Programmable Network Infrastructure (SPIN ’20)

P4 Tofino implementation of TCP RTT Measurement:
https://github.com/Princeton-Cabernet/p4-projects/tree/master/RTT-tofino

Mojgan Ghasemi, Theophilus Benson, and Jennifer Rexford. **Dapper: Data Plane Performance Diagnosis of TCP.** In Proceedings of the Symposium on SDN Research (SOSR ’17)

Figure 2: Dapper’s architecture: (1) data plane monitoring on edge, (2) control plane diagnosis techniques
Fig. 4: Traditional scenario without in-network control.

Fig. 5: In-network P4-based implementation.

Fig. 6: TCP session approach.

Fig. 8: Stop position error without in-network actions. Acceleration of $(30^2/s^2)$.

Conclusions

- **Fluid Networks** are here to stay
- Just a *term* to refer to the confluence of technological advances that are re-shaping networks (functions and architectures)
  - High-performance SW I/O and Virtualization Stacks
  - True Programmable Networking HW (NICs and ASICs)
  - Vast amount of Computing, from the Edge to the Core
  - Many instances in the literature and many opportunities ahead
SMARTNESS (Smart Networks and Services for 2030)

XR
Virtual (VR), augmented (AR), and mixed reality (MR)
Multi-modal Man-Machine interaction
Cloud Gaming
Tactile Internet
Immersive Media & 3D Holograms
Holographic Telepresence

Robotics & Intelligent Industry 4.0 process automation
Industrial IoT & Smart manufacturing
V2X & Internet of Vehicles (IoV)
Intelligent Transport Systems (ITS)
5G/6G-based high-precision positioning
Domain-specific APIs for 3rd party apps

Future Applications

SMARTNESS 2030

Scientific & Technology Advancements

Customized Edge Computing
- Telco Cloud & Edge Native OS, SW/HW Architectures
- Edge compute systems across devices & distributed cloud
- Decentralized data mgm: ingestion, processing, routing, storage, streaming
- Lightweight Virtualization

Cognitive Architectures
- ML/AI 4 Networking
- Distributed Machine Intelligence
- Network + Service Big Data Repository

Fluid Control & Data Planes
- Network Architecture matches what Applications need
- Intent-based Networking
- Deep Slicing / Network Slicing 2.0
- Programmability and Automation
- Flexible, Adaptable CP/DP functions
- In-Network Processing (e.g., IETF Computing in the Network)
- Time-Sensitive Networking (TSN) & Deterministic Networking (detnet)
- Network Service Mesh (NSM)

Trustworthiness: Security, Privacy, Safety, Ethics

Sustainability
References

References

References

- H. Tu Dang et al. P4xos: Consensus as a Network Service. 2018
Credits

- http://www2.technologyreview.com/article/412194/tr10-software-defined-networking/
- Fluid 1 image source: https://www.trzcacak.rs/detail/199233/
- Fluid 2 image source: http://www.pngall.com/water-png/download/1933
- Intelligent Brain image source: https://ui-ex.com/explore/transparent-brain-artificial-intelligence/
- Orchestrator image source: https://apievangelist.com/2015/02/06/when-you-are-ready-for-nuanced-discussion-about-who-has-access-to-your-api-i-am-here/
- Poison image source: https://www.stickpng.com/cat/miscellaneous/poison?page=1
Danke!

Questions?
BACKUP
Control Loops (in a Loop)
Flexibility

- M. He et al. “Flexibility in Softwarized Networks: Classifications and Research Challenges”

<table>
<thead>
<tr>
<th>Category</th>
<th>Aspect (see Sec. III-B)</th>
<th>SDN</th>
<th>NFV</th>
<th>NV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adapt configuration</td>
<td>Flow Configuration: flow steering</td>
<td>✓</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Function Configuration: function programming</td>
<td>-</td>
<td>✓</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Parameter Configuration: change function parameters</td>
<td>-</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Locate functions</td>
<td>Function Placement: distribution, placement, chaining</td>
<td>-</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Scale</td>
<td>Resource and Function Scaling: processing and storage capacity, number of functions</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Topology Adaptation: (virtual) network adaptation</td>
<td>-</td>
<td>-</td>
<td>✓</td>
</tr>
</tbody>
</table>
The Fluid Networking landscape

- UE
- BS MEC
- Access Cloud
- PoP DC
- Cloud DCs

Customer Premises

Edge

Cloud DCs Core
The Fluid Networking landscape

- UE
- Customer Premises
- BS MEC
- Access Cloud
- Edge
- PoP
- DC
- Cloud DCs
- Core
What is a Slice?

[Image of cartoon characters with elephant and text boxes labeled ETSI, ONF, IETF, 3GPP, MEF, and Academia]
Towards Deep Slices

Fragmented Standardization

Business & Technological challenges
From infrastructure sharing to any-layer any-resource sharing (from PHY to APP)

Deep
End-to-End, Multi-Domain (tech + admin)
Tenant Choice & Control
Isolation
Scalable

any resource, any function anywhere
Deep Slicing: Challenges up front

Standardization gap goes hand by hand with a series of key challenges from provider’s perspective on (i) scalability, (ii) arbitration, (iii) slice planning and dimensioning, and (iv) multi-domain (cf. [FG-NET- Contribution]). Both business and technical implications can be deemed necessary for such multi-operator slice provisioning context.

From the business side, some key implications include: (i) coordination models, (ii) inter-provider SLAs, (iii) pricing schemes, (iv) service specification, and (v) customer facing advertisement.

From a technical perspective we highlight (i) slice decomposition, (ii) discovery of domains, (iii) common abstraction models, (iv) standard interfaces/protocols, APIs.

**Network Function**

**NFV Infrastructure PoP Provider 0**

- Resources (incl. NFs) need to be allocated for the new situation
- Proper Control and Mgmt Interfaces offered by the remote domains

**Network Provider 1**

- Opportunity for instantiating NFs in proximity
- Better service fit
- User demand changes (maybe unexpectedly or bursty)

**Network Provider 2**

**Multi-Domain Administrative & Technological**

**Isolation under massive multi-tenancy**

**Protocol stack Choice & Control**

Source: Adapted from slide courtesy by Luis M. Contreras, Telefonica
Acknowledgments

Work by Christian Rothenberg was supported by the Innovation Center, Ericsson Telecomunicações S.A., Brazil under grant agreement UNI.64.

Ack. Mateus Santos and Pedro Gomes for input insights

This work includes contributions funded was partially funded by the EU-Brazil NECOS project under grant agreement no. 777067.

Luis M. Contreras and Alex Galis, co-authors of ITU-T FG 2030 input Doc.6: Network 2030 Challenges and Opportunities in Network Slicing.

Raphael Rosa (PhD candidate at UNICAMP), for his contributions to the vision around Unfolding Slices, Control Loops (in a Loop), Disaggregated Metrics/Prices, and Smart Peering
Slicing Journey: from 5G towards 2030

Executive Summary

- From siloed slices to generalized network cloud slicing
- Deep, massive resource sharing & multi-tenancy
- New Tenant-Provider relationships and power of choices

Source. Adapted from slide courtesy by Luis M. Contreras, Telefonica
Different Slicing Models & Approaches

- **Mode 0**: VIM-independent [Infra Slice aaS] [Bare-metal Slice]
- **Mode 1**: VIM-dependent [Platform Slice aaS]
- **Mode 2**: MANO-based [NFV aaS]
- **Mode 3**: Service-based [Service Slice aaS]

Types of Slices and Control Responsibilities

- **External / Tenant-managed Slices**
- **External / Provider-managed Slices**
- **Internal Slices**

Source: A Network Service Provider Perspective on Network Slicing. Luis M. Contreras and Diego R. López. IEEE Softwarization, January 2018
Slicing under massive any resource multi-tenancy (gone wild) … or when sharing economy meets cloud network slicing

Source: [http://www.h2020-necos.eu/](http://www.h2020-necos.eu/)

Source (image “sharing economy”): [https://www.kreezalid.com/blog/78403-what-is-sharing-economy](https://www.kreezalid.com/blog/78403-what-is-sharing-economy)
Expose just enough information to make optimal resource orchestration.

Orchestration Layer: ~1

Domain Controllers: 10s

Network nodes: 1000s