Network Function Virtualization and Flexible Service Chaining in Multi-Domain/Provider Environments: Recent Developments

Keynote at NetSoft 2016

Robert Szabo, Ericsson Research
Outline

• ETSI NFV MANO Revived
• Technical and Business Implications
• Technical details
• Use-Case: Multi-Domain Multi-Technology Service Function Chaining
• Summary
ETSI NFV MANO Revived

- ETSI NFV MANO Revived
  - Implications
  - Technical details
    - UC-SFC
    - Summary
Network Function Virtualization Orchestration (NFVO)

Virtual Machine Hosting

Cloud Controller (e.g., OSC) VIM

Comp Ctrl Storage Ctrl Net Ctrl

Virtual Network

WIM Network OS (SDN Ctrl)

Data Plane Manager (e.g., OpenFlow)

Compute Servers

Physical network
Virtualization Resource Orchestration

Virtualization Service Orchestration

Resource Orchestration (aggregated virt res)

Orchestration and Optimization (virt res)

Domain Spec. Adapt.

Technology adapters

NFVO

Network Service Orchestrator (NSO)

Common Abstraction and API

Orchestration and Optimization (virt res)

Domain Spec. Adapt.

June 9, 2016
Distributed NFVO?

Diagram showing a network of RO components connected through VIM/WIM and WIM layers, with NSO and NFVO at the top.
Hierarchical NFVOs?
Multi-Provider NFVOs?

Provider A

Provider B

NFVO

NSO

RO

VIM/WIM

Domain Spec.

VIM/WIM

Domain Spec.

VIM

Domain Spec.

VIM

Domain Spec.
Multi-Provider NFV0s?
Does It Remind YOU of

...
Does It Remind YOU of the ONF SDN Concept?

- SDN App
- SDN Ctrl
- SDN Ctrl (Virtual Dataplane)
  - SDN Ctrl
    - Domain Spec. Adapt.
  - SDN Ctrl
    - Domain Spec.
  - SDN Ctrl
    - Domain Spec.
Key Messages #1

A **different functional split** of the ETSI MANO architecture enables

- More clear separation of concerns
- Scalability by hierarchies (layers)
- New business interactions
Implications

- ETSI NFV MANO Revived
  - Implications
  - Technical details
    - UC-SFC
    - Summary
Key Assumptions

- Edge Computing or highly distributed cloud
- Multi-technology, multi-domains and multi-provider environments (e2e)
- **Network can be scarce resource**
  - Somewhere in the e2e path
Technical Implications on Resource Slice as a Service (SlaaS)

**ABSTRACTION**
- <Topology of>
- Resources and
- Capabilities for both **software** and **networking**

**CONTROL API**
- <Embedment in the given topology>
- Requirements on
  - Latency, rate
  - CPU, mem, storage (flavor)
  - Locality, affinity, ...

+ context for SLA, pricing, policies, etc...
Vision: One 5G Network – Multiple Industries

From dedicated physical networks and resources for different applications...

...to a “network factory” where resources are traded and new architectures are “manufactured by SW”

Unified Orchestration and Management Plane for Virtualization

Physical Resources
(Access, Connectivity, Computing, Storage, ..)

Network Services
(slices)

Service n
Health
Robotic communication
Media
MBB Basic

June 9, 2016
Robert Szabo, Ericsson Research @ NetSoft 2016
EU/5GPPP/ 5G Exchange - Multi-Provider Operation: SlaaS

- Data plane
- Control plane
- Management and orchestration plane

Mind the recurring Resource Slice programming interface!!!

Request Resource Slice as a Service (SlaaS) with (SLA)

Program Resource Slice

Customer (Tenant)

Multi-provider Orchestrator Administration A

Multi-provider Orchestrator Administration B

Multi-provider Orchestrator Administration C

Operator A administration

Operator B administration

Operator C administration

Datacenter

Packet

ENNI

June 9, 2016
Robert Szabo, Ericsson Research @ NetSoft 2016
EU/5GPPP/ 5G Exchange – Resource Slice Programming

Multi-provider Orchestrator Administration A

Multi-provider Orchestrator Administration B

Multi-provider Orchestrator Administration C

Customer (Tenant)

Global information exchange

Request Resource Slice as a Service (SlaaS) with (SLA)

Program Resource Slice

Operator A administration

Operator B administration

Operator C administration

Obs: Red IS the ONLY customer seen by Blue/Green

Obs: Red’s VNF not used by the Customer

Obs: Red’s shared VNF used ALSO by the Customer

Obs: Customer’s dedicated VNF

VNF

VNF

VNF

Data plane

Network Controller

Packet

Datacenter

Packet

Legacy nets.

Opto

Packet

VNF

ENNI

ENNI

June 9, 2016

Robert Szabo, Ericsson Research @ NetSoft 2016
Key Messages #2: Business Implications

- Outreach to **verticals** (5G)
- (Industry) **automation**
  - Open standardized interfaces
  - Open software (open source)
- **Faster Time to Market** and **Flexibility**
  - Seamless combination of managed (VNFAaaS) and self-operated VNFs

- **NFVO**
  - Network Service Orchestrator (NSO)
  - Resource Orchestration (aggregated virt. res)
Technical Details

EU – FP7 UNIFY: Unifying Carrier and Cloud Resources
UNIFY Architecture (Overview)

- NFV & SDN, as enablers
- Multi-level (recursive)
- Reference Points for NF-FG:
  - NF-FG: Network Function Forwarding Graph
UNIFY Virtualization & Control
@UNIFY RP

Top of Big Switch & Big Software (BiS-BiS) virtualization

1. Combined software & network resource abstraction
2. Control API

Software API
Compute Ctrl

SDN API
SDN Ctrl

Compute Node (CN)

Network Element (NE)

Joint Software and Network Abstraction & Control API

Big Switch with Big Software (BiS-BiS)
UNIFY Virtualization & Control
@UNIFY RP

“Atomic” joint configuration of
1. VNF placements
   • Defines ports!!!
2. Forwarding definition

Step 1: VNF deployment
VNF1 CN VNF2

Step 2: Overlay definition
VNF requests WITH forwarding definition

BiS-BiS

June 9, 2016
Robert Szabo, Ericsson Research @ NetSoft 2016
Topology of BiS-BiS (res: sw & net, cap, ...)

Diagram showing a network topology with nodes labeled BiS and RO (UNIFY) connected by arrows indicating connections and relationships.
UNIFY Architecture

- Logically **centralized** resource orchestration
- Joint software & network
  - Virtualization
  - Programming
- Multi-level / recursive architecture
Universal Node (UN): the native BiS-BiS node
Bridging gap between compute & networking

- UN hosts VNFs as full VMs, lightweight containers or enhanced logical switch instances
  - HW acceleration
- Achieve high performance (e.g. by Intel® DPDK) in UN Virtual Switching Engine & optionally in various VNFs.

Physical networking, virtual networking (vSwitch) & VNF (compute) in the same node
Solution Additionally Features

- NF decomposition (e.g., VNF=> VNF-FG)
- Embedding algorithms with joint SW and Network optimization with constraints
- Multi-technology support
  - OpenStack, Docker, Click
  - OpenFlow NE, POX and Ryu SDN Controllers, (ODL)
  - “native” Universal Node
    - HW acceleration by decomposition
- CP-DP split / direct elasticity control [IEEE Network]

- Open-source proof of Concept Prototype demonstrations
  - SIGCOM 2014, 2015
  - EWSDN 2015
  - BBF 2016 Q1
  - ETSI, May 2016
  - See you at IETF96 Bits’n’Bites
Key Messages #3

IT WORKS

-- OpenSource PoC prototypes

- Data model published at IRTF / NFVRG
- ETSI NFV compliant solution with NFVO and VIM (like a middleware)
- Integrating and “packaging” for IETF Bits-N-Bites @Berlin
Service Function Chaining

Revived?

- ETSI NFV MANO Revived
  - Implications
  - Technical details
    - UC-SFC
    - Summary

- **Application (Software)**

- **SFC (e.g., NFV and SDN)**

- **SW defined NW Slice**
**SFC Architecture**

- **SF**: Service Function
- **SFF**: Service Function Forwarder
- **Control interfaces**
  - **C1**: Classifier
  - **C2**: SFF
  - **C3**: SF
  - **C4**: SFC unaware SFs

Ref: [I-D.ietf-sfc-control-plane](https://datatracker.ietf.org/doc/html/I-D.ietf-sfc-control-plane)
SFC with SDN NEs

- NE: Network Element
- Mind
  - C1, C2: SDN Control
  - C3: Element Management
Steering Control (C1/C2) vs SF Configuration (C3)

- SFC Control Plane decomposed
  - SDN Controller
  - Element Management
- Mind
  - User Plane vs Management Plane traffic is situational only, handled the same
How SFs get Deployed?

- **SDN**
- Once there; SFs are just resources (aka port)
- **NFV**
- i) SFs go to the cloud, then
  ii) interconnect ➔ overlay
- **Combination needs coordination**
- Not defined!!!
Joint SW and Network Abstraction and Programming API

- Straightforward combination
- Can describe single node, nodes, domains, administrations, ...

Joint NFV and SFC Control API (Resource Slice Programming)

SFC Control Plane
NFV Orchestration
Virtualized Infrastructure Manager

SFC Classifier, SFF, SFC proxy

Big Switch with Big Software
Resource Slice Programming == SFC Programming ?!
Key Messages #4

Joint Software and Forwarding Abstraction

- Integrates SDN and NFV
  - Provides an SFC Control Plane
- Defines a programmatic interface for network slicing
Summary

1. A different functional split of the ETSI MANO architecture
2. Joint Software and Forwarding Abstraction
3. Open Source PoC implementations
4. Outreach to verticals, automation, faster TTM
5. SW Apps ➔ Service Function Chaining ➔ Network Slice
EU FP7 UNIFY: Acknowledgement

UNIFY consortium (Nov 2013 – July 2016)

This work is supported by FP7 UNIFY, a research project partially funded by the European Community under the Seventh Framework Program (grant agreement no. 619609). The views expressed here are those of the authors only. The European Commission is not liable for any use that may be made of the information in this document.
This work is partially supported by 5G-PPP 5GEx, an innovation action project partially funded by the European Community under the H2020 Program (grant agreement no. 671636). The views expressed here are those of the authors only. The European Commission is not liable for any use that may be made of the information in this presentation.
Extra: Demo Use-case

A simple robotics app

- ETSI NFV MANO Revived
  - Implications
  - Technical details
    - UC-SFC
    - Summary
SW App for the Lego Mindstorm Gyroboy

- Sensors and Servos @Robot
- Balancing PID Controller @NetworkCloud
- Requirements: **RTT < 40 msec**
  - Edge Computing
- **Goals**
  - Slice programming, flexible service creation, automation, fault tolerance with a real - industry like - use-case
  - Learn how (SW+NET) API influence the SW APP design
PoC Testbed
SW ➔ SFC ➔ Network Slice

Balancer1
192.168.1.156

Splitter

Mininet+Click Domain

Transport SDN Domain

OpenStack Domain

Docker Domain

Docker2 Domain

Balancer1
192.168.1.156

Host 1

Robot2
9c:5c:8e:af:2e:a0
192.166.1.158

Robot AP

Robot1

Robot2
9c:5c:8e:af:2e:a0
192.166.1.158

Robot AP

Robot1

June 9, 2016
SW ➔ SFC ➔ Network Slice
SW ➔ SFC ➔ Network Slice
Lego Balancing Conclusions

- Robust SW App design + (SW+Net) programming allow flexibility in service creation
- State sharing / migration
- Full cloning of the SF (incl. MAC + IP) solved ARP issue
Summary

1. A **different functional split** of the ETSI MANO architecture
2. **Joint** Software and Forwarding Abstraction
3. **Open Source PoC** implementations
4. Outreach to verticals, **automation**, faster TTM
5. SW Apps ➔ Service Function Chaining ➔ Network Slice