Unifying the Programmability of Cloud and Carrier Infrastructure

Wouter Tavernier
EU FI CLUSTER event
The project UNIFY (11/2013 - 04/2016)

Major Vendors:
- Deutsche Telekom
- Telecom Italia
- Intel

Major Service Providers:
- OTE

SMEs:
- Travelping

Research Institutes:
- iMinds
- SICS
- acreo

Universities:
- MÜKGYETM 1702
- Universidad del País Vasco
- Euskal Herriko Unibertsitatea

Project Management:
- eICT
Focus on seamless integration

Invocation of Dynamic Service Chains
UNIFY Control Plane (Programmability)

Joint Orchestration in Network and Clouds
UNIFY Control Plane (Abstractions)

Data performance optimized infrastructure virtualization
(x86 based architecture)
UNIFY Universal Node
UNIFY Control Architecture

- Combined Compute, Storage & Network abstraction over all resources (RO)
  - forwarding elements,
  - compute host capabilities,
  - hardware based network function capabilities,
  - data plane links
- Overarching Resource Orchestration for Network Function Forwarding Graphs

Relying on wide range of infrastructure
Abstractions in service programming

- Service Graph
- Abstract Network Function Forwarding Graph
- Abstract Resource Mapping
- Physical Infrastructure
  - Compute, Storage, Network and topology
  - Instantiated Service Graph
Bring agility of IT in telecom: Service Provider DevOps

• Propose a definition for integrating developer and operator roles in telecommunication service provider networks
• Build a set of tools with dual developer-operator audience, based on research challenges identified in the following areas:
  – Observability for Software-Defined Infrastructure
  – Verification for Software-Defined Networks, in particular OpenFlow
  – Troubleshooting of performance degradations in a distributed Network Function Virtualization environment
  – VNF Development support for sandboxing prototypes
Universal Node Concept

Universal Node

Local mapping

Different VM types

Intel DPDK
Ongoing development in prototypes

- ESCAPE: Extensible Service ChAin Prototyping Environment using Mininet, Click, NETCONF and POX (Demo at Sigcomm 2014), BME, iMinds

- Multi-layered Service Orchestration in a Multi-Domain Network Environment in ODL & OpenStack, BME, Ericsson Hungary (demo at EWSDN 2014)

- Universal Node prototype: Supporting Fine-Grained Network Functions through Intel DPDK (paper/demo at EWSDN 2014), Polito, BISDN
ESCAPE + ODL + OpenStack

UNIFY architecture: multi-layer orchestration

Service Provider

Application Layer: user / other provider

Service Graph manager

GUI

VNF mgmt (Clicky)

VNF catalogue editor

Service Graph editor

Service Graph config

Orchestrator

netconf client

routing module: traffic steering

resources

VNF catalog

Mapping

POX

Orchestrator/Controller

OS/ODL domain ("Node")

OpenStack datacenter

OpenDaylight controller

Mininet domain

Mininet Container #1 ("Node")

VNF #1

mgmt agent

click instance

VNF #2

mgmt agent

click instance

NC port

netconf agent

port: 830

n1-eth1

n1-eth2

n1-eth3

n1-eth4

datapath #1

OF port

n2-eth1

n2-eth2

n2-eth3

n2-eth4

datapath

transport network (OpenFlow)

Host #1

host process

h1-eth0

Host #2

host process

h2-eth0

Server1

VM1: VNF #3

Server2

VM2: VNF #4

Server

Server3

Server4

VNF #3

VNF #4

DC OF network

datapath

datapath

datapath

Nova

Neutron
Trends & challenges in service decomposition & orchestration

A **Network Service** can be implemented using any element of a potentially unlimited set of Service Chains of network functions which themselves can be decomposed:

- **Example:** 99.999% availability, support for 20K flows/users
- How to ensure + measure throughput, delay, etc. through decomposition?
- Which NFs to add and where?

**Challenge**

- **Example:** NF x and NF y require ARM architecture
- How to avoid decompositions which cannot be embedded by available infrastructure?
- How to make search process in decomposition/embedding scalable?

**Infrastructure**

**SLA ensurance**

**Scalable decomposition & orchestration**