Network Function Chaining in DCs: 
the Unified Recurring Control Approach

János Elek, Dávid Jocha and Róbert Szabó

Ericsson Research, Hungary
Agenda

1. **Background**
   - Main related concepts from the tutorial *UNIFYing Cloud and Carrier Network*
     
     *Resources: Architecture, Orchestration and Service Provider DevOps*

2. L2 forwarding

3. UNIFY example
“Today, rigid network control limits the flexibility of service creation”
UNIFY Architecture
BiS-BiS

SDN API

SDN Ctrl

Software API

Compute Ctrl

Joint

Compute and Software Control API

Network Element (NE)

Big Switch with Big Software (BiS-BiS)

Joint Orchestration

VNF1

VNF2

VNF1

VNF2

Compute Node (CN)
Agenda

1. Background
   - Main related concepts from the tutorial *UNIFYing Cloud and Carrier Network*
     
     *Resources: Architecture, Orchestration and Service Provider DevOps*

2. **L2 forwarding**

3. UNIFY example
Scenario

Transport domain | DataCenter domain

OpenStack
SDN
Transport domain
DataCenter
domain
Scenario
NF1
L2 forwarding in OpenStack

- NEUTRON Networking API v2.0
  - Supports legacy L2/L3
  - Doesn’t support SDN
- Questions
  - Out of the box vs. modified OpenStack/Neutron?
  - Generic vs. service chain aware VM?
  - Self-standing DC vs. support from neighboring domain?
  - Expose internals of DC networking vs. not to expose?
Black Box approach

- Of-the-shelf OS
- Tunneling
- External node
- VM
Local RO approach

- Modified DC network controller
- No support needed in VM or other node
Unify domain

Black Box approach

White Box approach
Agenda

1. Background
   • Main related concepts from the tutorial *UNIFYing Cloud and Carrier Network Resources: Architecture, Orchestration and Service Provider DevOps*

2. L2 forwarding

3. **UNIFY example**
Virtualization 1.

1: R\textsubscript{compute} (REST)
2: R\textsubscript{network} (REST)
3: I\textsubscript{compute} (NF-FG)
4: I\textsubscript{network} (NF-FG)
5: V\textsubscript{unified} = I\textsubscript{compute} + I\textsubscript{network} (NF-FG)
6: V\textsubscript{unified} (NF-FG)
Virtualization 2.

RO CA CA-SDN SDN Net BiS/BiS OpenStack DC

1: R\_SDN (REST)
2: I\_SDN (NF-FG)
3: I\_DC (NF-FG)
4: V\text{unified} = I\_SDN + I\_DC
5: V\text{unified} (NF-FG)

http://msc-generator.sourceforge.net v4.4
Service request 1.

1: Mapping on $V_{unified}$

2: $M_{unified} \rightarrow (NF-FG)$

3: $M_{unified} \rightarrow M_{SDN} + M_{DC}$

4: $M_{DC}$

(NF-FG)

5: Done

6: $M_{SDN}$

(NF-FG)

7: Flow rules

(REST)

8: Done

9: Done
Service request 2.

1: Mapping on $V_{unified}$

2: $M_{unified}$
   (NF-FG)

3: $M_{unified}=M_{compute}+M_{network}$

4: $M_{compute}$
   (NF-FG)

5: $D_{VNFI}$
   (REST)

6: $D_{VNFI}$
   (REST)

7: $M'_{compute}$
   (NF-FG)

8: Update $M_{network}$

9: $M_{network}$

10: Flow rules
    (REST)

11: Done

12: Done
Forwarding Rule Mapping

- Abstract tags
- Technology specific tags
- Domain border vs. internal tags
Summary

• Need for joint orchestration of Networks & Cloud
  • Unify framework solution

• Fine tuned forwarding (SDN) needed in the Cloud domain
  • OpenStack needs extra functionality, multiple options available

• The joint virtualization and control of UNIFY can be applied to OpenStack-OpenDaylight Data Centers
  • Advantages compared to standard separated cloud and WAN orchestration approaches
Black box Virtualization

1: SDN domain topology (REST)
2: SDN domain infrastructure description, including SAPa (NF-FG)
3: DC resources: nodes, CPU, mem, images... (REST)
4: OS domain infrastructure description, including SAPb (NF-FG)
5: Create Unified abstract view
6: Unified infrastructure of all domains (NF-FG)

http://msc-generator.sourceforge.net v4.6.2
Black box deploy

1. Request
2. Orchestration on Unified view
3. Request mapped on Unified view (NF-FG)
4. Convert abstract request to domain specific
5. Select IP_VXLAN_NF1 and IP_VXLAN_NF2 from SAPa's subnet
6. Split NF-FG to domains
7. Domain specific request
8. Deploy VF1 VM
9. Deploy VF2 VM
10. Deploy VF1 VM (REST req)
11. VF1 VM deployed with local IP (REST resp)
12. Assign public IP is local IP (REST)
13. Deploy VF2 VM (REST req)
14. VF2 VM deployed with local IP (REST resp)
15. Assign public IP to local IP (REST)
16. Deploy forwarding overlay
17. NF-FG deployed (BisBis' flow rules extended with public IPs)
18. Set VF1 VM metadata: VxLAN from IP_VXLAN_NF1, VXLAN_ID1 (REST)
19. Set VF1 VM metadata: VxLAN to local IP of VF2 VM, VXLAN_ID2 (REST)
20. Set VF2 VM metadata: VxLAN from local IP of VF1 VM, VXLAN_ID2 (REST)
21. Set VF2 VM metadata: VxLAN to IP_VXLAN_NF2, VXLAN_ID3 (REST)
22. NF-FG deployed (BisBis' flow rules extended with public IPs)
23. Update SDN domain's flow rules with public IPs received from OS domain
24. Domain specific request
25. Create VxLAN in public IP of VF1 VM
26. Create VxLAN in public IP of VF2 VM
27. Deploy flowrules in OVS switches (REST)
28. Install flowrule (OpenFlow)
29. Install flowrule (OpenFlow)
30. Install flowrules (OpenFlow)
31. NF-FG deployed
32. NF-FG deployed

http://msc-generator.sourceforge.net v4.6.2