Annex B (normative):
NFV ISG PoC #28 Report

The following normative disclaimer shall be included on the front page of a PoC report:

Submission of this NFV ISG PoC Report as a contribution to the NFV ISG does not imply any endorsement by the NFV ISG of the contents of this report, or of any aspect of the PoC activity to which it refers.

B.1 NFV ISG PoC Report

B.1.1 PoC #28 Project Completion Status

Indicate the PoC Project Status. Can the PoC be considered completed? If this is a multi-stage PoC project, indicate the Reported Stage status and plans for future Project Stages.

- Overall PoC Project Completion Status: 100% completed phase 1. Achieved the PoC Objectives

- PoC Stage Completion Status (Optional - for Multi Stage projects only): Phase one: 100% completed Achieved the PoC objectives. Decided not to pursue Phase 2 as one of the participating vendors has gone through acquisition, no longer had resource for the integration.

B.1.2 NFV PoC Project Participants

Specify PoC Team; indicate any changes from the NFV ISG PoC Proposal:

- PoC Project Name: _SDN Controlled VNF Forwarding Graph_
- Network Operator/Service Provider: Deutsche Telekom; Contact: Manuel.Paul@telekom.de
- Manufacturer A: Huawei Contact: Linda.Dunbar@huawei.com
  Cathy.H.Zhang@huawei.com
  louis.fourie@huawei.com
- Manufacturer B: Freescale Contact: John Myla: johnmyla@freescale.com
  Jeff.Maguire@freescale.com
  Srini Addepalli saddepalli@freescale.com
- Manufacturer B: RiverBed Contact: kevin.glavin@riverbed.com
- Additional Members: Qosmos Contact: Nicolas.BOUTHORS@qosmos.com

B.1.3 Confirmation of PoC Event Occurrence

To be considered complete, the PoC should have been physically demonstrated with evidence provided that the demonstration has taken place.

Provide details on venue and content of PoC demonstration event. Provide pictures and supporting literature where available. Please identify who was present at the demonstration event (optional).

- PoC Demonstration Event Details:
B.1.4 PoC Goals Status Report

Specify PoC Goals from NFV ISG PoC Proposal (clause A.1.2). Identify any changes from the original NFV ISG PoC Proposal with an explanation as to why the changes were made. Indicate the extent that each goal was met. Provide sufficient information for those not familiar with the PoC goals to understand what has been achieved and/or learned.

- PoC Project Goal #1: SDN controlled NFV VNF service delivery (insertion/deletion/reconfiguration)
  - SDN/OpenFlow controlling dynamic steering of various flows to their designated VNFs (with different VNF types and from different vendors) in scalable way
  - What/how OpenFlow protocols are used to control forwarding nodes, virtualized network functions, etc.

  Goal Status (Demonstrated/Met?) ___Yes__________________

- PoC Project Goal #2: NFV MANO (OpenStack, CloudStack, etc.) can leverage the SDN controller’s APIs to:
  - Control diverse (multi-vendors) VNFs
  - Consume statistics and metadata from the controller for feedback

- PoC Project Goal #3: NFV MANO interworking with orchestration engine that can translate user’s requests to SDN controllers, coordinate among SDN controller(s) and VNFs for coherent service delivery

- PoC Project Goal #4: interoperability with existing non-SDN infrastructure

List additional (optional) PoC Project Goals (follow the same format).

B.1.5 PoC Feedback Received from Third Parties (Optional)

- Where applicable, provide in a free text, feedback received from potential customers, Ecosystem partners, event audience and/or general public.

B.2 NFV PoC Technical Report (Optional)

PoC Teams are encouraged to provide technical details on the results of their PoC using the PoC Scenario Report template below.

B.2.1 PoC Scenario Report

Use the table structure below and refer back to the Scenarios in the NFV ISG PoC Proposal (clause A.2.2) and provide information for each of them. Feel free to include additional Scenarios developed during the implementation of the PoC. Do not eliminate Scenarios that were not performed, instead provide a brief status for each with a reason why the scenario was not performed. Do not hesitate to fill multiple instances of the table if several objectives have been demonstrated for each scenario.
## Objective Id: 1

### Scenario 1: User customized service chain

**Mapping to NFV ISG Work:**

- **NFV Use Case #4 (VNF Forwarding Graphs):**
- **Requirement:** Elasticity & Portability (5.2 & 5.4 of gs-NFV004v010101p)
- **E2E Arch:** “Or-Vi” interfaces

### Description:

**Description of the PoC Demo Objective:**

To demonstrate that users can specify different service forwarding graphs, and the controller can dynamically control traffic from different users to traverse through different service chains.

### Pre-conditions

Description of the pre-demo state, reference measures to be taken before demo starts (Optional)

### Procedure:

<table>
<thead>
<tr>
<th>1</th>
<th>Description of how the PoC Demo Objective has been demonstrated. Can be detailed as a sequence (intermediate steps, measures to be taken, ...)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Orchestration: Provide interface for users to specify their desired service graph, calculate optimal service graph, and interact with the controller and service functions.</td>
<td></td>
</tr>
<tr>
<td>2. Controller: control the network devices based on the interaction with the orchestration</td>
<td></td>
</tr>
<tr>
<td>3. X86 Compute Node: to host virtualized switches and virtualized L4-L7 functions</td>
<td></td>
</tr>
<tr>
<td>4. Classifier node: assign different service chain ID to flows based on instruction from the orchestration.</td>
<td></td>
</tr>
</tbody>
</table>

### Results Details:

Traffic was manually generated. Since there is no standard interface for Service Function Vendor to register their VNFs to orchestration, we had to do it manually.

### Lessons Learnt & Recommendations

What was learnt with this demo? What are the recommendations for the NFV ISG work or the industry as a whole.

Need interface for service function vendors to register their VNF capabilities and locations.
### Objective Id:
Scenario 2: Scale out and Scale in Mapping to NFV ISG Work:
- NFV Use Case #4 (VNF Forwarding Graphs);
- Requirement: (5.4 of gs-NFV004v010101p)
- E2E Arch: “Ve-Vnfmc” and “Nf-Vi”

<table>
<thead>
<tr>
<th>Description:</th>
<th>Description of the PoC Demo Objective:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>To demonstrate the VNF Lifecycle management. This scenario will show how virtual service function instances are added to or deleted from compute nodes based on user demand, and how each newly added virtual service functions are configured.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Pre-conditions</th>
<th>Description of the pre-demo state, reference measures to be taken before demo starts (Optional)</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Procedure:</th>
<th>Description of how the PoC Demo Objective has been demonstrated. Can be detailed as a sequence (intermediate steps, measures to be taken, …)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1. Orchestration: interface with VNF management system to create or delete service function instances based on user demand</td>
</tr>
<tr>
<td></td>
<td>2. VNF management: manage the VM image and additional disk storage instantiation, licensing, baseline configuration, and VNF termination.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Results Details:</th>
<th>The Vendor proprietary VNF Life Cycle management was used to add and delete VNFs.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Lessons Learnt &amp; Recommendations</th>
<th>What was learnt with this demo? What are the recommendations for the NFV ISG work or the industry as a whole.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Need interface for service function vendors to register their VNF capabilities and locations.</td>
</tr>
<tr>
<td>Objective Id:</td>
<td>Scenario 3: Service Functions Failover</td>
</tr>
<tr>
<td>--------------</td>
<td>---------------------------------------</td>
</tr>
<tr>
<td><strong>Mapping to NFV ISG Work:</strong></td>
<td></td>
</tr>
<tr>
<td>NFV Use Case #4 (VNF Forwarding Graphs);</td>
<td></td>
</tr>
<tr>
<td><strong>Requirement:</strong> Resiliency &amp; Service Continuity ;(5.5 &amp; 5.7 of gs-NFV004v010101p)</td>
<td></td>
</tr>
<tr>
<td>E2E Arch: “Ve-Vnfm”</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Description:</strong></th>
<th>Description of the PoC Demo Objective:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>To demonstrate the capability to deliver services when VNFs, switches, or links fail.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Pre-conditions:</strong></th>
<th>Description of the pre-demo state, reference measures to be taken before demo starts (Optional)</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th><strong>Procedure:</strong></th>
<th>Description of how the PoC Demo Objective has been demonstrated. Can be detailed as a sequence (intermediate steps, measures to be taken, ...)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1. Orchestration: interface with VNF management system to create or delete service function instances based on user demand</td>
</tr>
<tr>
<td></td>
<td>2. VNF management: manage the VM image and additional disk storage instantiation, licensing, baseline configuration, and VNF termination.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Results Details:</strong></th>
<th>The Vendor proprietary VNF Life Cycle management was used to add and delete VNFs.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th><strong>Lessons Learnt &amp; Recommendations:</strong></th>
<th>What was learnt with this demo? What are the recommendations for the NFV ISG work or the industry as a whole.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Need interface for service function vendors to register their VNF capabilities and locations.</td>
</tr>
</tbody>
</table>
Objective Id: Scenario 4: Dynamic by-pass service functions depending on service function run-time status

Mapping to NFV ISG Work:

NFV Use Case #4 (VNF Forwarding Graphs);
Requirement: Performance (; 5.3 of gs-NFV004v010101p)
E2E Arch: “Ve-Vnfm”

Description: Description of the PoC Demo Objective:
To demonstrate that a tenant’s flow can be re-routed on-the-fly to bypass a service function whose treatment is not needed anymore per its previous treatment result, e.g. when a FW check returns OK to the first few packets of a flow, the follow-on packets of the flow can bypass the FW, saving CAPEX on the FW appliance.

Pre-conditions: Description of the pre-demo state, reference measures to be taken before demo starts (Optional)

Procedure: 1 Description of how the PoC Demo Objective has been demonstrated. Can be detailed as a sequence (intermediate steps, measures to be taken, ...)

Results Details: We haven’t demonstrated this scenario

Lessons Learnt & Recommendations
What was learnt with this demo? What are the recommendations for the NFV ISG work or the industry as a whole.

Need interface for service function vendors to register their VNF capabilities and locations.
### B.2.2 PoC Contribution to NFV ISG

Use the table below to list any contributions to the NFV ISG resulting from this PoC Project.

<table>
<thead>
<tr>
<th>Contribution</th>
<th>WG/EG</th>
<th>Work Item (WI)</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>NFVEVE(15)000049r3</td>
<td>NFV EVE</td>
<td>ETSI GS NFV-EVE 005</td>
<td>Lessons learned in using SDN to control Service Graph Forwarding.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>One of the Key Components for Virtualized Network Functions:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Data path have to be dynamically adapted to the Virtualized Network Functions as they are moved, added, or removed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Without SDN, it is very difficult, if not impossible, for data flows to dynamically steered to their designated service functions</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- VNFs by different vendors have different features and need different provisioning. There are mechanisms today (e.g. OpenFlow or IETF’s I2RS) for SDN controller to dynamically control L2/L3 switches and routers, it is virtually impossible to control security functions, such as Authentication VNF, FW/IP/IDS VNFs because there is no standard interface to those VNFs</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SDN controlled Service Graph forwarding can be deployed in today’s network without requiring replacing existing equipment and control plan. But SDN Controller can’t control non switch/router based VNFs yet.</td>
</tr>
</tbody>
</table>

### Describe Application

**SDN Controlled VNF Forwarding Graph**

<table>
<thead>
<tr>
<th>Role of SDN</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>In this PoC, the SDN is used to control individual flows traversing through their designated virtual network functions (VNF).</td>
<td></td>
</tr>
</tbody>
</table>

Assuming that the VNF are already interconnected, which can be controlled by traditional control protocols or SDN approaches.

This PoC is SDN controlled Service layer (the Red Line) as defined in GS NFV INF 005 clause...
6.2.3 [x] and in the reference diagram extracted from GS NFV INF 005 Network Function Virtualization Infrastructure Architecture (Figure 22):
The following figures maps the PoC components to NFV Architectural framework (gs_NFV002v010101p):

Mapping the PoC components to the End to End Service Forwarding Graph (ETSI GS NFV002 V1.1.1 (2013-10)):

<table>
<thead>
<tr>
<th>Plane</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Management</td>
<td>SFC Orchestrator, i.e. SFC manager and SF VNF Manager</td>
</tr>
<tr>
<td></td>
<td>SFC applications (e.g., SFC OAM functions)</td>
</tr>
<tr>
<td>Control</td>
<td>SFC SDN controllers</td>
</tr>
<tr>
<td>Data</td>
<td>Classifier</td>
</tr>
</tbody>
</table>
SFC management and control plane elements are responsible for translating user/application requests; converting requisite policies into network topology dependent paths; and disseminating steering commands to relevant forwarding nodes. SFC data plane components are responsible for carrying out the steering commands.

In the ONF SDN paradigm, the SFC Orchestrator requests necessary changes through the northbound interface (NBI) of the OpenFlow SDN controller, which in turn implements the requested changes via OpenFlow. If service function are not OpenFlow-speaking or are not under the domain of OpenFlow SDN controllers, the SFC Manager may need to interact directly with the service functions themselves or interact with another service function instance manager.

### Role of NFV Orchestrator

The SFC Manager Component of the NFV Orchestrator (Service Orchestrator) translates clients’ high-level intent policy based network-topology agnostic requirements into flow classification constructs, and sequences of L4-L7 service function constructs associated with the flows. The SFC Manager provides an interface for service clients to specify SFC requirements including the desired classification criteria, policies, sequence of service functions, etc.

The Orchestrator consists of two key components: the SF Chain Manager and the SF Instance Manager. The Chain Manager is responsible for translating user’s SFC flow classification rules and SFC requirements into flow classification constructs and associated sequences of L4-L7 service functions. The Instance Manager is responsible for managing the life cycle of each service function instance, and for keeping track of each service function instance’s locator and chain header encapsulation and transport capabilities. The Chain Manager will then pass these classification constructs, sequence of service functions, and the L4-L7 service function instance catalogue to the SDN Controller through the SDN NBI so that the SDN Controller can further translate them into southbound OpenFlow programming constructs.

### Network

The SDN Controller is responsible for setting up service function steering/chaining paths.
**Controller details**

It translates flow classification constructs and associated sequences of SF locators into southbound OpenFlow programming commands. It locates the flow classifier and instructs it how to classify a flow and the flow packets with an appropriate SFC header. It is also responsible for locating the sequence of SF Forwarders associated with the sequence of SF Locators so as to enforce the flow going through the sequence of designated SF Nodes. That is, the SDN Controller will use those OpenFlow programming constructs to program the classifier to classify the traffic flows and program the SF Forwarders to steer the flow through the sequence of SF locators.

**Network Controller to NFV Orchestrator interface and function**

- This interface allows communication of VNF instance locator and flavor details as well as a service chain’s service function ordering information. To support different types of SDN Controllers, the Service Chain Orchestrator (can be implemented in OpenStack) will interface with the SDN Controller through Plugin Driver mechanism. The Service Chain Orchestrator will pass the following information to the SDN Controller:
  - Sequence of Requested Service Functions: This specifies the sequence of service function treatment for a tenant’s traffic flow.
  - Traffic Flow Classification rules: It consists of a set of flow descriptors.
  - List of Available Service Function Instances: each SF Instance contains information such as the Instance’s Locator, the Instance’s Flavor. An example of a SF Instance is a firewall or an IDS.

**Data Forwarding Plane**

The Service Function Forwarder (SFF) is responsible for forwarding data packets to their designated service function instances. As the case with other SFC entities, the SFF may come in physical, virtual or any other form factor. The SFF may make its forwarding/steering decision by matching the chain identification information carried in the SFC header with the next-hop information provided by the SDN controller. The SFC header should carry sufficient information for each SFF in the path to establish symmetric flows in forward and reverse directions, if necessary.
### Data Control Plane (Programming plane)

OpenFlow 1.3+-capable SFFs may make its forwarding decisions on L2-L3 header fields such as MAC or IP addresses. With an OpenFlow version that supports L4 fields such as TCP and UDP ports, the SFF will be able to make forwarding decisions at L4.

### Lessons learned and (new) requirements

One of the Key Components for Virtualized Network Functions:

- Data path have to be dynamically adapted to the Virtualized Network Functions as they are moved, added, or removed.
- Without SDN, it is very difficult, if not impossible, for data flows to dynamically steered to their designated service functions.
- VNFs by different vendors have different features and need different provisioning. There are mechanisms today (e.g. OpenFlow or IETF’s I2RS) for SDN controller to dynamically control L2/L3 switches and routers, it is virtually impossible to control security functions, such as Authentication VNF, FW/IPS/IDS VNFs because there is no standard interface to those VNFs.

SDN controlled Service Graph forwarding can be deployed in today’s network without requiring replacing existing equipment and control plan. But SDN Controller can’t control non switch/router based VNFs yet.

### Gaps Identified

There are many types of VNFs, the Video Optimization based VNF is very different from FW/IPS/IDS based VNF, or Router/switch based VNF.

In order for NFV Orchestration to automatically utilize various types of VNF, ETSI NFV should have an initiative to categorize VNFs, and then move on to define interfaces to various types of VNFs.

For example, for the category of Flow based Security Functions (VNFs), “Subject – Object – Function – Action” paradigm can be used to manage them ([I2NSF@ietf.org](mailto:I2NSF@ietf.org))

- Subject - Match values based on packet data Packet header or Packet payload
- Object - Match values based on context. E.g. State, time, geo-location, etc.
- Action- Egress processing, such as Invoke signaling; Packet forwarding and/or transformation; Possibility for SDN/NFV integration
- Functional Profile – E.g. IPS:<Profile>, signature file, Anti-virus file, URL filtering file, Threat Database, etc. Integrated and one-pass checks on the content of packets. The functional profile is vendor specific and differentiates vendor unique innovation.

### B.2.3 Gaps identified in NFV standardization

Use the table below to indicate Gaps in standardization identified by this PoC Team including which forum(s) would be most relevant to work on closing the gap(s). Where applicable, outline any action(s) the NFV ISG should take.
B.2.4 PoC Suggested Action Items

- Provide suggested Action Items and/or further work required from the NFV ISG and/or external forums.

Different vendors have different interfaces and different ways to instantiate their virtual functions. In order for Orchestration to work properly, it is necessary to have an interface to each vendor to register their VNFs. There are many types of VNFs, the Video Optimization based VNF is very different from FW/IPS/IDS based VNF, or Router/switch based VNF.

In order for NFV Orchestration to automatically utilize various types of VNF, ETSI NFV should have an initiative to categorize VNFs, and then move on to define interfaces to various types of VNFs.

NFV should work with other SDOs to standardize the interface for vendors to register their Service Functions, and have standard interface to dynamically sending policies to the Service Functions via SDN controller.

NFV should provide more concrete requirement to the IETF I2NSF WG. IETF I2NSF WG is to specify the standard interface for control and monitor the Network Security functions. The goal is to allow SDN controller (or security controller) to get unified reports from different vendors, and to pass application initiated rules to the network security functions.

This Wiki has more detailed information on this new initiative: https://sites.google.com/site/interface2nsf/home

Mailing list: i2nsf@ietf.org
To Subscribe: https://www.ietf.org/mailman/listinfo/i2nsf

---

**Gap Identified** | **Forum (NFV ISG, Other)** | **Affected WG/EG** | **WI/Document Ref** | **Gap details and Status**
---|---|---|---|---
NFVEVE(15)000049r3 | NFV EVE | ETSI GS NFV-EVE 005 | Different vendors have different interfaces and different ways to instantiate their virtual functions. In order for Orchestration to work properly, it is necessary to have an interface to each vendor to register their VNFs. There are many types of VNFs, the Video Optimization based VNF is very different from FW/IPS/IDS based VNF, or Router/switch based VNF. In order for NFV Orchestration to automatically utilize various types of VNF, ETSI NFV should have an initiative to categorize VNFs, and then move on to define interfaces to various types of VNFs. NFV should work with other SDOs to standardize the interface for vendors to register their Service Functions, and have standard interface to dynamically sending policies to the Service Functions.
B.2.5 Any Additional messages the PoC Team wishes to convey to the NFV ISG as a whole?

- Provide any feedback in a free text format to the NFV ISG. Please indicate whether the team wishes any specific message to be published or publically quoted.

PoC is a great way for vendors to work together to validate how virtualized network functions can be integrated with switches and SDN controller.

B.2.6 Any Additional messages the PoC Team wishes to convey to Network Operators and Service Providers?

- Are there any specific requests/messages that the team would like to convey to Network Operators and Service Providers?