Paper Review by

Seshagiri Prabhu Narasimha

Paper Information

Paper Title: OpenVirteX: Make Your Virtual SDNs Programmable
Type of Paper

	Full paper (9 or more pages)	»	Short paper (Less than 9 pages)

Summary of the Paper

OpenVirteX (OVX), is a network virtualization platform. It enables operators to create and maintain virtual Software Defined Networks (vSDN). Its the first framework which enables tenants to specify their own Network Operating System (NOS) and control it through a programmable API. OVX serves as an Infrastructure as a Service (IaaS) in the context of SDN. Some of the previous research works including FlowVisor, VeRTIGO and FlowN are the inspiration for the OVX. FlowVisor and FlowN provides shared address space among the tenant networks and hence any misconfiguration would lead to overlapping or security concerns. By introducing virtualization, OVX enables complete modularity between tenant networks and thus it supports any arbitrary topology and addressing schemes, configured as per tenant request. Network embedder tool is used to convey the tenant requests via API calls to OVX. A user could specify a virtual network's addressing scheme, topology and NOS link, this generates virtual to physical mapping. The internal OVX architecture does a loose decoupling of virtual elements from physical counterparts.

OVX enforces the tenant traffic isolation using the physical address fields of virtual components: Flow ID (looks up the particular flow that entered the virtual link and restore its values when the traffic exits the virtual link), Tenant ID (Isolates the different tenants on the virtual link) and Virtual link ID (keeps tracks of virtual links). Hence whenever a link layer packet arrives with a certain outport specified, OVX “knows” where exactly the other end of that link is.

The only limitation put by the OVX is while mapping the switches. No single physical switch cannot be partitioned. One virtual component can be mapped to one or more physical components. This increases the resilience of the network. As the virtual to physical mapping is logically centralized in the global map, these can be changed at runtime enabling tenants to do dynamic vSDN reconfigurations. Another minor feature is the ability to port the virtual networks under two conditions given that operator network support the same control protocol as OVX and the vSDN topology can be mapped onto the network without partitioning the physical switches.

The results shown by the OVX are remarkable when compared to previous related works. The control channel latency to create virtual networks through API has been reduced by half when compared to the existing solutions. The OVX was deployed and tested for a production scale on Internet 2, a nation wide research network in United States.

Contributions of the Paper

(What are the major issues addressed in the paper? Do you consider them important? Comment on the degree of novelty, creativity, impact, and technical depth in the paper)

[100-150 words]

The major issues addressed by this research work is inability to create customizable topology, isolated addressing schemes and control his/her virtual SDN with NOS of his/her choice. The importance of this research work is that it gives more freedom to the tenants to create on demand, re-configurable and customizable virtual networks with resilience. The security of the tenant network is quite high as there is complete network isolation.

Programmable API for tenants with customizable topology & full address space and resiliency of the topology network are some of the new techniques introduced by this paper.

Even though the core concepts of network virtualization, complete network isolation and resilience of tenant network are inspired from the standard OS virtualization, the impact of developing these are something great as other related works couldn’t succeed in achieving the same.

The paper does describe the atomic details of their implementation of architecture. It has mentioned very high level details about the internal and external OVX architecture.

Strengths of the Paper

(What are the main reasons to assess this paper as high-quality? e.g. innovative ideas/technologies, paper presentation, detailed performance evaluation.)

[at least 2-3 strengths, more are better]

1. The ideas put forward by the authors for the virtual networks gives a new dimension to the traditional networking.
2. The paper is presented in such a way that even a bachelor degree computer science student with little or no networking knowledge can understand the concept.

Weaknesses of the Paper

(What are the most important reasons to assess this paper as low-quality? e.g. writing skill, technical errors, unrealistic assumptions, unanswered questions, limited measurements or evaluation)

[at least one weakness, more are better]

1. Runtime load has not be measured and included in the results. The paper only describes the latency of the control channel during the creation of virtual networks.
2. The paper mentions at several places that physical components cannot be partitioned and keeps it is as a strong assumption. However, they do not mention the reason of keeping such a assumption/rule.

Open Issues and Future Work

(What are the options for future work that you think are important? If you are asked to work on the problem investigated in this paper, what will you do differently?)

[try to find something]

1. Snapshotting of the running VM is not enabled in the current version of OVX, once enabled, then the system would allow the tenants to port their virtual topology to work under different network operator.
2. The current system uses `openflowj` library which is strictly tied with `OpenFlow v1.0`. Once upgraded to v1.3 with `LOXI`, a marshalling and unmarshalling engine that generates it generates version-agonistic `OF` libraries in multiple languages in the southbound interface, thus allowing common open source `OFv1.0` controllers like NOX and Floodlight to run on an `OFv1.3` network.
3. OVX with `OF1.3` uses flow-based meters to enforce QoS for the virtual networks. This would enable a fully isolated environment with performance guarantees.