Virtualization, SDN and NFV – HOW DO THEY FIT TOGETHER?

Traditional networks lack the flexibility to keep pace with dynamic computing and storage needs of today’s data centers. In order to implement changes, each networking device needs to be configured individually. The increasing number of functions built into these networking devices results in even more complex, closed and proprietary setups. Software Defined Networking (SDN) and Network Function Virtualization (NFV) are the latest topics in the industry to address these problems. But what is the difference and how do they fit together?

SERVER VIRTUALIZATION

Historically, data centers consisted of specific server nodes providing computing power to run applications as well as their associated storage and the network interconnecting these servers with each other and with their clients. Over the past decade, one of the major industry trends was the virtualization of compute and storage resources, mainly to consolidate hardware and licenses. The basic concept of virtualization is the simulation of hardware platforms in software, i.e. the functionality is separated from the hardware. With compute virtualization, a thin layer of software called a hypervisor is implemented on the server that allows running virtual machines (VM) on top of it. A virtual machine is an instance that runs its own operating system, middleware and applications and looks like a physical server to its user, providing the same interfaces as a physical server.

To begin with, the major driver for compute virtualization was server consolidation in order to save money by buying fewer servers and therefore also reducing the power consumption. However, once virtualization technology was adopted, more interesting use cases were found such as backup and disaster recovery or provisioning of on-demand compute resources. A huge benefit of server virtualization is VM mobility which means the ability to migrate a VM from one server to another live (i.e. without shutdown and restart). VM mobility has become a key element of workload agility.
In addition, compute virtualization also simplifies the setup of multitenant data centers. Multitenancy refers to the capability of the data center to host multiple separate zones, each of which can serve a separate group of users with a specific service profile, for example departments within an enterprise or external customers. Two important requirements for multitenant data centers are on the one hand address space and traffic separation, i.e. different tenants can use the same IP address space whilst traffic from one tenant is not visible to another tenant, and on the other hand the placement of virtual machines anywhere within the data center without restrictions. Furthermore, each tenant might have its own security requirements.

With a growing number of servers and virtual machines in the data center, tools are necessary that allow managing virtual machine images, computing resources, storage resources and IP addresses, and provide the ability to create, move and delete virtual machines from compute resources. These tools are commonly known as cloud orchestration platforms, e.g. OpenStack or VMWare vCenter.

What’s the impact of server virtualization to the network? In traditional data centers, there are physical switching networks. It does not really matter how this switching infrastructure looks like. Connected to the switching network, there are servers hosting a number of VMs. All the network functionality is implemented within physical infrastructure today. For instance, the isolation of tenants is usually done in layer-2 using VLANs. With VM mobility, networking gets even more complex as VMs move from one server to another maintaining their MAC address as well as their IP address. Visibility of network flows and debugging becomes very difficult.

SOFTWARE DEFINED NETWORKING (SDN)

Applying the same principles of virtualization which are used in server virtualization today to the network, we end up with virtual networks which are a logical software-based view of hardware resources. The physical network devices simply are responsible for forwarding packets through the network while the virtual network is implemented in software and provides an abstraction layer.

The basic idea of software defined networking (SDN) is the physical separation of data plane and control plane to provide an abstraction layer and make it easier to optimize each. The concept is not totally new. Separation of data plane and control plane has been done before, e.g. in MPLS networks. What’s new is that the control plane and data plane are physically separated (i.e. two different boxes and not just two components within the same box) and the communication between both entities is an open standard-based protocol. This separation only makes sense if the control plane gets somehow centralized and controls multiple devices. As a result, customized applications may run on top of the SDN controller which allow taking decisions based on end-to-end visibility across the network.
The control plane in an SDN environment runs on the SDN controller which allows communication with the data plane running on a networking device via the southbound API, for instance OpenFlow, PCE, BGP-TE or XMPP. In addition the SDN controller provides an interface for applications running over the network to make use of the virtualized network, for instance orchestration and automation services. This interface is called the northbound API.

**The key benefits of SDN are**
- faster software development,
- programmable network elements,
- faster provisioning, and
- centralized intelligence.

Let’s return from the formal definition of the SDN architecture to the exercise of how to implement virtual networks in data centers. First of all, with VM mobility in mind the physical switching infrastructure in a data center does not seem to be the right place to enforce virtual networking as each VM migration would require touching the configuration of multiple switches. Basically following the concept of server virtualization, the introduction of some sort of network hypervisor is needed. This network hypervisor is called a virtual switch (vSwitch) or sometimes a virtual router and is a piece of software running on the server and handling every packet. The virtual network that is implemented on the vSwitch provides the same APIs that physical networks have and follows the same operational model as the VM (including create, move, snapshot, etc.). Using the SDN architecture, the vSwitch usually only provides the data plane of the network while the control plane remains in a centralized SDN controller. Existing vSwitch are using overlay technologies (e.g. VxLAN, NVGRE or MPLSoGRE) to implement the virtual network.
Although SDN and NFV are mentioned in the same breath with each other, they are not the same and each one can exist without the other one. While SDN is an approach to virtualize the network infrastructure, NFV aims to virtualize the network functions such as load balancing, firewalls, deep packet inspection, NAT and more.

Virtualization again results in running these services in software on standard off-the-shelf hardware instead of dedicated proprietary hardware appliances. This approach makes total sense because most of these appliances are already built using standard CPUs inside because layer-4 to layer-7 functions are hard to implement in ASICs. So, why not run the application software on the off-the-shelf hardware? A drawback of dedicated appliances is once you move a VM or a virtual network within your data center, you have to move the appliance as well. It is much simpler to move a virtual function. In addition, using NFV has the benefit that you can give away control of the network function to tenants which might not be possible with a single box. Also, the upgrade of network function gets much simpler as you can just clone the existing function, upgrade it, move the configuration, and shut down the old network function. In case, the upgrade network function causes problems, you can easily switch back to the old one.

**The key advantages of NFV are**

- **reduction in CAPEX** because there is no longer a need to buy purpose-built hardware and pay-as-you-grow model avoids overprovisioning
- **reduction in OPEX** due to more efficient space, power and cooling consumption as well as simplification in rollout and management
- **improvement of time-to-market** by reducing the time needed to deploy new services and by reducing the risks involved with rollout of new services
- **flexibility to meet innovation and changed requirements**

---

**SDN/NFV USE CASES FOR DATA CENTERS**

In order to illustrate the concepts discussed so far, let’s turn to some real-world SDN use cases for the data center. Consider a multitenant data center where tenants run a tiered application, e.g. a front-end web server and a backend database. Customers connected to the data center via public internet should be able to reach the web server while the database server is only reachable by the web server. To automatically setup such an application, a workflow has to be setup including provisioning of a pre-built VM image using an orchestration system such as OpenStack, definition of virtual networks and associated IP ranges, creation of VMs and assignment of IP addresses from virtual network pool (e.g. using OpenStack Neutron), distribution of routes to WAN router and definition and enforcement of policy for the server communication. The creation of the virtual networks implies that an SDN controller establishes the necessary overlay tunnels to implement the virtual networks once the VMs are created.
The second real-world scenario involves a virtualized service function, e.g. a virtualized firewall. Again a pre-built VM image is provided by the orchestration system. In addition to the definition of virtual networks and associated IP ranges, a service template needs to be defined which includes information about the forwarding mode and the service image. Afterwards it is defined which virtual network uses which service template. This use case describes a simple service chaining scenario.

As already mentioned, by using VM mobility debugging becomes more difficult as it may not be clear to the network administrator on which server a particular VM or application is running on. Therefore, a third possible use case is composed of a traffic analyzer instead of virtualized network function. The main difference in this situation is that the traffic has to flow transparently from virtual network one (VN1) to virtual network two (VN2) and has to be copied to the analyzer port. Under normal circumstances, this service is only temporary, i.e. highly dynamic. The advantage of this scenario is that the analyzer can be placed anywhere in the data center and does not have to be within the data path.

Basically, all the three use cases described above can be combined into a more complex setup.
PUTTING IT ALL TOGETHER

The real problem in today’s data centers is the orchestration. Although there are tools to do the orchestration for various parts of the data center individually, e.g. for the virtual machines, the end-to-end orchestration is still unsolved. Mapping new service requests into virtual machines, provisioning and configuring these virtual machines, implementing network policies and inserting the virtualized network functions into the data path (sometimes called service chaining) are problems which can be addressed by an SDN approach. Thus, the use of SDN in data centers with huge number of VMs and frequent movement of VMs enables agility and elasticity.

SDN and NFV Relationship (Source: Open Networking Foundation)

Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>BGP-TE</td>
<td>Border Gateway Protocol – Traffic Engineering</td>
</tr>
<tr>
<td>MPLS</td>
<td>Multiprotocol Label Switching</td>
</tr>
<tr>
<td>NAT</td>
<td>Network Address Translation</td>
</tr>
<tr>
<td>NFV</td>
<td>Network Function Virtualization</td>
</tr>
<tr>
<td>NVGRE</td>
<td>Network Virtualization using Generic Routing Encapsulation</td>
</tr>
<tr>
<td>PCE</td>
<td>Path Computation Element</td>
</tr>
<tr>
<td>SDN</td>
<td>Software Defined Networking</td>
</tr>
<tr>
<td>VxLAN</td>
<td>Virtual Extensible LAN</td>
</tr>
<tr>
<td>XMPP</td>
<td>Extensible Messaging and Presence Protocol</td>
</tr>
</tbody>
</table>
About Xantaro

Xantaro is an international Service Integrator for carriers and service providers, as well as for enterprises and organisations, whose core business relies on the network. With a comprehensive service portfolio, Xantaro supports its customers in designing, implementing, maintaining and developing high-performance networks innovatively and economically.

Being vendor-independent, and through its partnerships with numerous technology leaders, Xantaro is at the forefront of new developments and market trends. Its close co-operation with customers and partners, combined with the experience of Xantaro’s in-house technology experts, ensures an excellent understanding against the demands of complex carrier-class networks. This expertise is concentrated by Xantaro into the unique idea of „Service Integration“: The holistic consideration of a network – from the transport up to the application layer – and the appropriate development and integration of customised solutions based on technologies of different network layers and vendors.