CHAPTER

7

SLICE MANAGEMENT IN VIRTUAL SOFTWARE DEFINED NETWORK

Mohamed Khalafalla Hassan, Nurzal Effiyana Ghazali, and Mutaz Hamad

7.1 INTRODUCTION

Due to the massive virtualisation success in the computer domain, network virtualisation has become an attractive research area in networking (Berman et al., 2014). Concerning software-defined networks (SDN), network virtualisation produced several virtual networks (or slices), each with its abstractions of the underlying network topology (Blenk, 2018). Concepts like "slicing" already exist in the field of networking. For instance, VLAN and multiprotocol label switching (MPLS) are layer two technologies in the traditional network.

However, network virtualisation now tends to create slices for the overall physical network regardless of the traditional layering approach. Resource abstractions that would include forwarding tables, central processing unit (CPU) resources, and network bandwidth can be assigned to each slice. The merits that would have accrued in traditional network slicing are also to be found in software-defined virtual network slicing, albeit with more freedom and agility. The flexibility and agility can be owed to the ease of programming provided by SDN hypervisors. This allows for fast service provisioning, dynamic resource allocation and network automation.

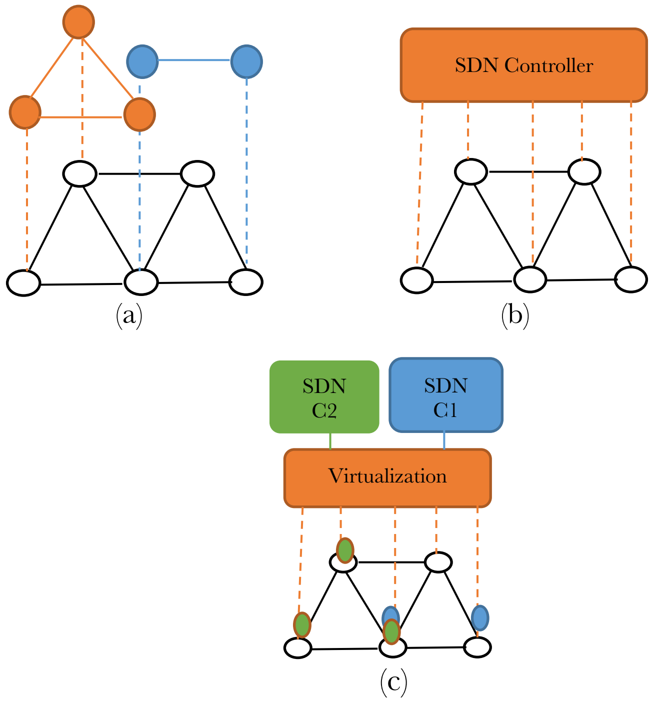

Figure 7.1 Comparison for (a) Virtual network, (b) Software-defined network, and (c) Virtual software-defined network

The Figure 7.1 (a) shows that virtual networks are nested inside a substrate network whereby the dashed lines denote virtual resources. Figure 7.1 (b) shows the SDN controller operating on a physical SDN. With the introduction of virtual software-defined networks (vSDN) which refers to networking architecture in which virtualisation and SDN technologies are used to enable the creation of virtual network functions (VNFs) and virtual network services as depicted in Figure 1.1 7(c), network administrators can create, delete, and manage the various slice dynamically. The virtualisation layer acts as an orchestrator for the tenants' virtual networks (SDN C1 and SDN C2). While SDN C1 controls two network components, while SDN C2 has power over three network elements (Blenk, 2018). The subsequent section will discuss the most notable hypervisors and frameworks that addressed partially or fully the slice management and quality of service (QoS) support while section 1.3 will discuss the significance of each solution in the context of the virtual software-defined network (vSDN).

7.2 RESOURCE MANAGEMENT MECHANISMS

FlowVisor (FV) (Sherwood et al., 2010) is well known to be the first hypervisor for SDN networks. FV enabled the capability of sharing SDN networking resources across several controllers. FV eventually operated as a stand-alone application on commodity hardware (server) (Sherwood et al., 2010) where it ensured that tenants have segregated flow areas for the duration of their lease. If a controller attempts to address a flow switch not in their flow spaces, it modifies the packet headers to reflect this. Otherwise, FV generates an OF error message. It is worth mentioning that the majority of the vSDN hypervisors are considered extensions to the FV hypervisor.

MobileVisor applied the FlowVisor approach in mobile packet core networks as presented by Figure 7.2 (Van Giang & Kim, 2014). In MobileVisor, the FV capability was included in the architectural framework of a virtual mobile packet network, which comprised numerous underlying physical mobile networks such as 3G and 4G networks. The MobileVisor can be considered as a proxy between multivendor data traffic and RAN.

The authors in Koponen et al. (2014) proposed the network virtualisation platform (NVP), focusing on data centre network resource abstraction managed by the cloud tenants for multi-tenant environments. NVP worked as a controller for SDN tenants to operate their SDN controllers via the application protocol interface (API) to control their slices in the data centre. Using NVP logical data paths (tunnels) can be created between the destination and the source open virtual switches (OVS) where the logical path corresponds to the corresponding tenant slice.