ABSTRACT
We demonstrate a new controller platform, called the software-programmed networking (SPN) operating system (OS) that enables tenants to flexibly manage their own virtual network services using the abstraction of a virtual network object (VNO). SPN OS separates distributed control of virtual network services from centralized management of network-wide resources.

The main components in the SPN OS architecture are the Arbiter and VNO pool as shown in Fig. 1. On receiving a new service request from the tenant, the Arbiter creates a new VNO in the pool and allocates resources to the VNO based on global optimization of network resources. A VNO is a complete representation of a virtual network service throughout its lifecycle. It provides topology and address space virtualization along with built-in functions such as traffic engineering, resiliency, etc. Unlike virtualization supported in existing SDN controllers, VNOs are designed to provide fine-grained control over features of the underlying physical network such as bandwidth and backup paths. For example, the Arbiter might provide several ways to map the logical topology to connections on the physical network, and a VNO executing a traffic engineering application might control the mapping used to achieve good performance.

We have built a prototype implementation of SPN OS in Java. Fig.2 depicts our prototype's graphical interface. The VNOs and Arbiter use GraphDB for their internal topology and mapping representation. The Arbiter computes one or more service patterns including virtual-to-physical mapping to meet service constraints specified by the tenant. The Arbiter writes these service patterns into the VNO and returns control to the tenant. The tenant can use one or more valid service patterns in its VNO to activate, reconfigure, or deactivate the service on demand.

A distinctive feature of SPN OS is its use of the high-level language NetKAT for programming the service. In order to provide "universal portability" and "universal serviceability", a VNO defines each its virtual network service as a NetKAT program consisting of (i) the virtual network topology, (ii) the virtual network forwarding behavior, and (iii) the virtual-to-physical mapping. Prior to service instantiation, the NetKAT compiler translates the network program in a VNO to equivalent OpenFlow forwarding rules.

We will demonstrate the key elements of the SPN OS platform through virtual network service provisioning examples (e.g., big switch, ring etc.) in a physical network emulated in MiniNet. The lifecycle of virtual network services including request, activation and deactivation will be shown. Each service is specified as a NetKAT program and the compiled rules are installed on Openflow switches during service activation by the VNO. We will also showcase the capability of the NetKAT compiler to automatically insert necessary tags to achieve isolation among VNOs.

SPN OS enforces a clear separation between control decisions made by tenants through VNOs and those made by network operators through the Arbiter. The Arbiter allocates resources when VNOs are created, and tenants are free to decide how to utilize the resources encapsulated in a VNO, often without having to consult the Arbiter. This separation provides flexibility to tenants as well as ensures scalable performance.