1. Motivation

Virtualization of Software Defined Networks
- Allows multiple tenants to share a physical SDN infrastructure (slicing the SDN infrastructure)
- Each tenant can bring and run its own controller for a flexible control of its virtual SDN network (vSDN)
- Is achieved through a network hypervisor that abstracts vSDNs and translates control plane of each vSDN controller

Limitations of Current SDN Network Hypervisors [1]
- Inflexible architecture as current implementations are either in software hosted on servers or as hardware extensions to SDN switches
- Virtual SDN abstraction focuses on data plane abstraction, less attention to slicing of the control plane, which can be a performance bottleneck
- No dynamic and flexible mechanisms for adaptation available
- Not adapting to hardware platforms
- No SDN-based management available

2. HyperFlex Architecture

Design Goals [2]
- Realizes the virtualization of OpenFlow-based SDN infrastructures
- Achieves the flexibility of hosting hypervisor functions on servers in software or using available processing capabilities of network nodes
- Provides adaptation mechanisms in case of dynamic vSDN demands

Management System
- Central entity for management of virtual SDN networks
- Provides interfaces to state of the art hypervisors: FlowVisor, OpenVirtex
- Provides interfaces to SoA virtual network embedding algorithms (resource allocation between virtual networks)
- Uses python-based SDN controller Ryu to setup the isolation function on the network

Distributed Hypervisor Software [3]
- Provides a lightweight network hypervisor out-of-the-box: DITRA
- DITRA: Distributed hypervisor for elastic operation of SDN network virtualization layer

Monitoring Agents
- Benchmark and monitoring module for hypervisor resources, e.g., CPU

Control Plane Slices [2]
- Guarantees for the control plane slices between vSDNs via software or hardware modules
- Consists of OF switches that are controlled by the hypervisor SDN controller
- Protection of hypervisor resources, e.g., CPU utilization
- Software modules, e.g., control plane translation
- Control plane isolation in software, e.g., in terms of OpenFlow (OF) msg/s
- Control plane traffic isolation in hardware, e.g., policing of control throughput (kbps)

Function Placement Strategies [4]
- According to operation goals, different placement algorithms for hypervisor instances are possible
- Performance guarantees per virtual SDN network: average or maximum latency
- Can be used for planning virtual SDN networks

3. HyperFlex - Graphical User Interfaces

4. Demos

4.1 Admission Control [5]
- Interplay of tenant, management, and monitoring systems
- Tenant request for a virtual SDN data and control slices
- Tenant request for a control slice isolation mechanism
- Embedding of virtual data slices on the physical network
- Update feature of virtual SDN slices on run time

4.2 Control Path Migration Protocol [3]
- Online control path migration between multiple hypervisor instances
- Instantiation of hypervisor instance on run time
- Hypervisor load balancing and improving control slice latency
- Transparent for tenants, i.e., no control loss

References