REFERENCE DESIGN

Trellis: NFV Fabric

ONS TS-101 | Version 1.0 | March 2019

About ONF Reference Designs

Reference Designs (RDs) represent a particular assembly of components that are required to build a deployable platform. They are "blueprints" developed by ONF's Operator members to address specific use cases for the emerging edge cloud.

RDs are the vehicles to describe how a collection of projects can be assembled into a platform to address specific needs of operators. By defining RDs, ONF's operator members are showing the industry the path forward to solutions they plan to procure and deploy.

Each RD is backed by specific Operator partner(s) who plan to deploy these designs into their production networks and will include participation from invited supply chain partners sharing the vision and demonstrating active investment in building open source solutions. The RD thus enables a set of committed partners to work on the specification and a related open source platform.

Assembling the set of selected components defined by the RDs into a platform enables a proof-of-concept to allow the test and trial of the design. These platforms are called Exemplar Platforms and each of them will be based on a Reference Design and will serve as reference implementations. These platforms are designed to make it easy to download, modify, trial and deploy an operational instantiation and thereby speed up adoption and deployment.

About the Open Networking Foundation

The Open Networking Foundation (ONF) is an operator led consortium spearheading disruptive network transformation. Now the recognized leader for open source solutions for operators, the ONF first launched in 2011 as the standard bearer for Software Defined Networking (SDN). Led by its operator partners AT&T, China Unicom, Comcast, Deutsche Telekom, Google, NTT Group and Turk Telekom, the ONF is driving vast transformation across the operator space. For further information visit http://www.opennetworking.org

Disclaimer

Trellis (NFV Fabric)

Reference Design v1.0

Document Number: ONF TS-101
Document Revision Date: March 18, 2019
Document Revision Number: v1.0
Document Release Status: Final Specification

This reference design specification was authored by an operator-led Reference Design Team (RDT) composed of experts from:

Operator Group:

Comcast, AT&T and Deutsche Telekom

ONF Supply-Chain Partners:

Infosys

Write to rdspec@opennetworking.org with comments or questions.

Contributors

Comcast: Robert Howald (RDT Chair), Louis Donofrio, Nagesh Nandiraju, Sebnem Ozer

Infosys: Vignesh Ramamurthy

ONF: Saurav Das, Aseem Parikh

Document Revision History

Date	Revision	Description
20 Nov 2018	Draft 0.1	Initial Draft
5 Mar 2019	Draft 0.2	Updates to make remove exemplar implementation references
12 March 2019	Draft 0.3	Updates to incorporate inputs by AT&T & Infosys
17 March 2019	1.0	Final Specification for public release

TABLE OF CONTENTS

LIST OF FIGURES

1 INTRODUCTION

This Open Networking Foundation (ONF) Reference Design (RD) describes the SDN-native spine-leaf data center fabric optimized for edge services and applications.

The reference design provides a high level architecture to support the need for different edge services and applications. This architecture can be realized by using different technology options for SW and HW components, protocols and interfaces to develop new exemplar implementations. As one possible realization of this RD, Trellis is also the name of an exemplar NFV Fabric platform that uses the ONOS Controller with whitebox switching hardware and open source software.

1.1 PURPOSE & SCOPE

Trellis RD provides an architecture pattern to develop solutions for SDN/NFV based fabrics for service providers. The purpose is to define a common infrastructure component for both service providers and suppliers towards the development, productization and support of open source software and whitebox hardware.

The scope of the NFV Fabric (referred to interchangeably as “Trellis”) reference design is to provide an SDN/NFV based fabric for a broad set of access technologies, such as HFC, PON, Ethernet and wireless. The goal is to use SDN, NFV and Cloud technologies to build agile datacenters for the network edge. Unlike traditional networking approaches, in the classic SDN approach, the fabric itself does not run control protocols embedded in each network switch or router (such as BGP or OSPF), but is instead controlled by applications running on a clustered, scalable, and highly available controller platform. The NFV Fabric embodies two emerging architectural requirements for future carrier networks: slicing, and control-user plane separation (CUPS). Slicing is supported by allowing multiple separable instances of switching and routing to be instantiated in the same switches at the same time. CUPS is supported by using SDN applications to control all or part of the fabric as separate entities that can exhibit different properties. For external connectivity the entire fabric acts as one or more distributed virtual switch or router data-planes. Additionally, SDN controller applications for subscriber and service management, authentication of access components and

establishment of secure connections between components should be supported for access technologies. Service assurance through telemetry, logging and fault management platforms is also within the scope of the reference design. The scope should be flexible for the integration of new technologies in terms of new silicon and devices and new virtualization and software defined networking solutions.

1.2 ASSUMPTIONS, DEPENDENCIES, PROCESS VARIANCES, OUT-OF-SCOPE SUMMARY

This document assumes the following relationships among reference design, target, and time to market solutions.

The reference design is described in terms of one or more solution elements. The particular implementation stream document will define how to deliver the elements. The most important of these steps is the target or end state. Target describes the most desired reference design in the context of the preferred future environment. The reference design may also define a series of well-known intermediate elements that might be needed to go to market sooner or with near term constraints that prevent moving directly to the target. These are defined as Time-to-Market solutions.

Given these relationships, the target reference design describes both a set of assumptions and dependencies in addition to the body of the design itself. Because the design may change over time, it is also expected that the assumptions and dependencies may also change with the Time-to-Market solutions.

The assumptions for the SDN/NFV Fabric reference design are:

1. *Operators are interested in initial commercial deployments starting in 2018*
2. *Both brownfield and greenfield deployments are envisioned.*
3. *The reference design will enable network programmability and automation for cost-effective and simplified operations, self-optimization and self-healing.*
4. *NFV Fabric will be deployed in infrastructure aggregates, often called pods. A point of delivery, or PoD, is a "module of network, compute, storage, and application components that work together to deliver networking services. The PoD is a repeatable design pattern, and its components maximize the modularity, scalability, and manageability of data centers" (reference: Cisco Nexus 2000 Series Fabric Extenders Data Sheet).*
5. *NFV Fabric will include containerized software managed using*

Kubernetes.

Dependencies for NFV Fabric include:

1. *Working Kubernetes Environment*
2. *CI/CD Tools (e.g. Jenkins, etc.) for development as well as deployment instances*

The NFV Fabric exemplar implementation is related to several mature projects at ONF. Trellis includes components such ONOS, ONOS SDN Apps, Hardware Abstraction Drivers, and software components from Open Compute Project such as ONL and ONIE. Because this work interacts with existing released work in active communities at ONF, it is likely that some of the processes defined for normal new reference design work may need to be adjusted to ensure that the existing communities and their work do not become disenfranchised.

1.3 AUDIENCE

The ONF members are the current audience of this pre-release version of the document, per the ONF Reference Design Process, and at this stage this document should not be shared outside of the ONF membership defined at the top of the ONF membership page.

Following the ONF RD process for the timeframe for members to review and comment, and following review of comments by the TLT, the TLT will provide decisions about revision of the document and petition the ONF Board to release the RD as an ONF Final Specification.

1.4 DOCUMENT RELATIONSHIP

The NFV Fabric RD is a standalone document in the NFV Fabric/Trellis process.

The ONF site provides a Trellis wiki that provides references to the designs, exemplar code, workflows, JIRA board, meeting times, meeting recordings, developer meeting list and Slack channel.

1.5 SOFTWARE RELEASES

The NFV Fabric project should define software releases as a solution set for the software components, including but not necessarily limited to ONOS, Hardware abstraction Drivers, ONL, and OCP.

The NFV Fabric software release documentation should provide the solution

set information for these software releases.

The NFV Fabric software release documentation should also provide the lifecycle management of the compatible releases between these components, in order to define flexibility and dependencies for coordinated upgrades of the components.

The hardware from vendors may also include embedded software for controlling, monitoring and abstracting low level functions of the hardware, including BIOS, firmware, board support drivers and baseboard management controllers (BMCs). The vendors shall identify the required versions of these embedded software components, and how to upgrade these embedded software components using open software lifecycle management procedures.

1.6 HARDWARE RELEASES

The carriers define the hardware solution, including vendors, models, and releases. ONF suppliers do provide value to identify hardware for an ONF reference design, and to update the carriers with roadmaps and new product information for enhancements and improved cost.

2 REFERENCE DESIGN TARGET

Trellis is designed as an NFV and SDN native spine-leaf data center fabric optimized for edge services and applications as shown in Figure 1. Different access networks such as DOCSIS, PON, Ethernet and Wireless networks may be supported by the common platform. The fabric is controlled by a scalable, highly-available SDN controller. For the Trellis exemplar, this controller is ONOS and other controller platforms may be used for different exemplar architectures. The fabric consists of Leaf and Spine switches. Distributed Access Aggregation Switch (DAAS) is a special Leaf switch that connects access network components. A vRouter solution is provided for external connectivity. Applications integrated within the controller platform provide functionality for subscriber and service management, authentication of access components and establishment of secure connections between components.

The main HW and SW components in the Trellis platform are white box switching hardware and open source software, as shown in

Figure **2**. SDN apps run in containers and control the fabric. Microservices VNFs are local network functionalities in containers and can be common to all access technologies or specific to a certain technology. The design is modularized, based on micro-services and customized workflows can be added as required. Network orchestration and dynamic load balancing are done programmatically. Compute orchestration is programmatically managed through add-ons on top of Kubernetes. An SDN controller platform is modular with applications that can be removed and added based on the use case and scale to serve service providers' serving group sizes.

The reference design defines the following components:

- SDN controller and apps
- Virtual Router
- Kubernetes controlled container platform
- Integrated Network VNFs
- Leaf-Spine whitebox switches
- Compute servers

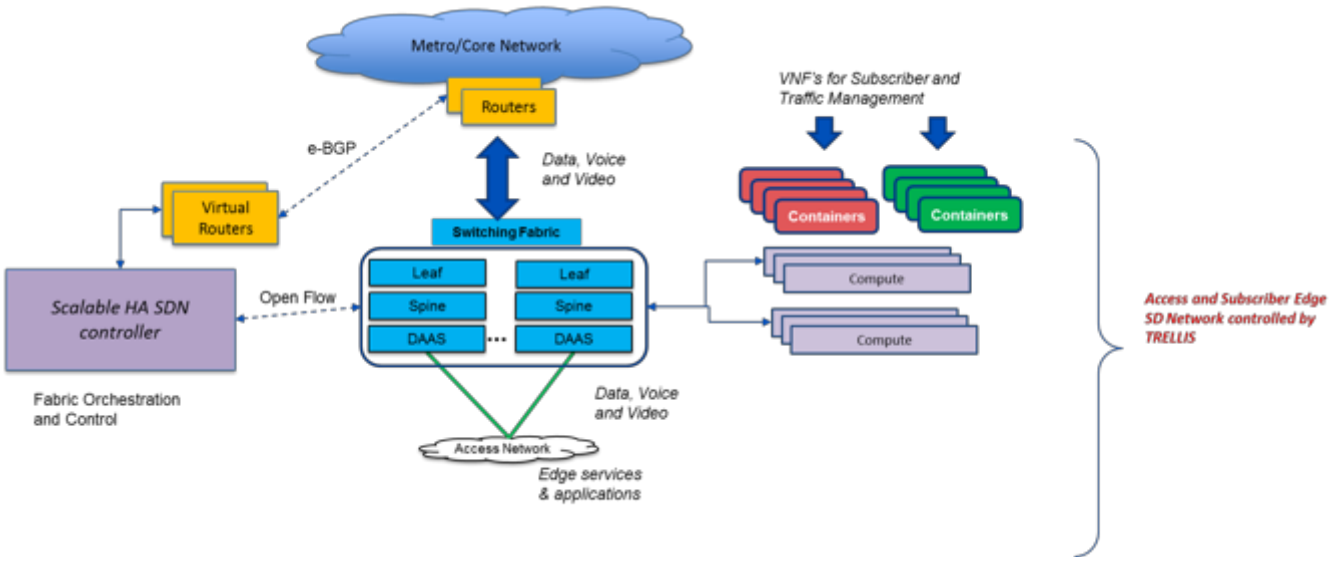

Figure 1 Simplified Project Overview

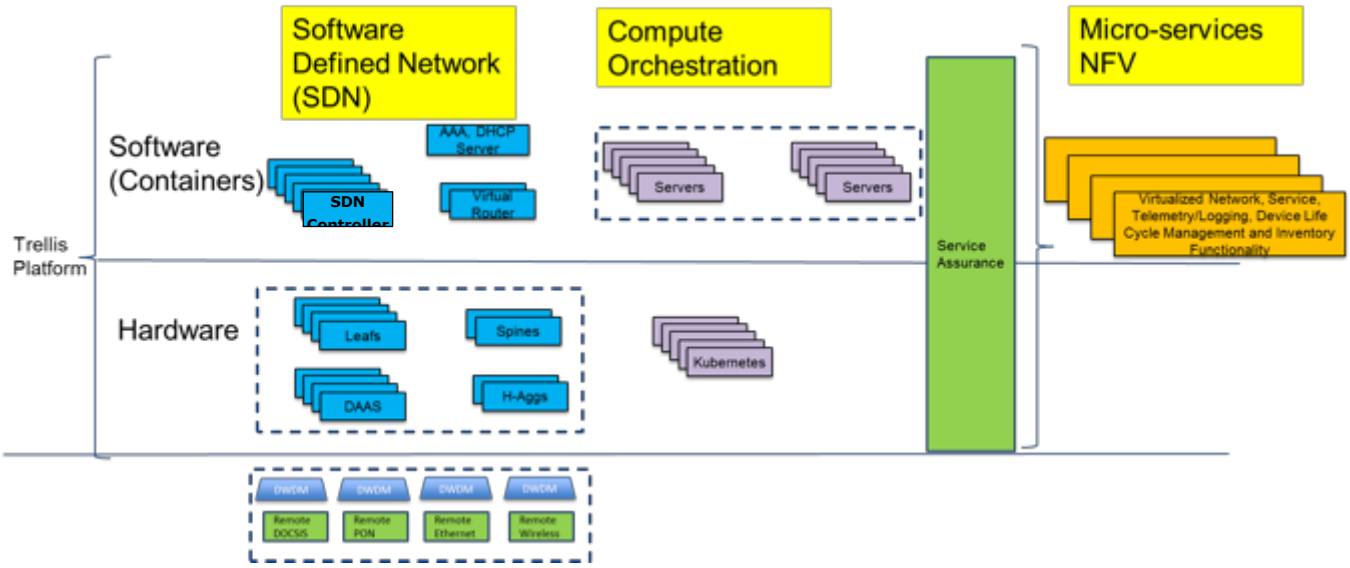

Figure 2 SW and HW Components

2.1 SALIENT CHARACTERISTICS OF THE END STATE

In recent years, many operators set a roadmap to transition to distributed access architectures (DAA) with flexible deployment options and better performance characteristics. The end-to-end system design is driven by the following objectives:

- Network programmability, elasticity, scalability and load balancing
- Automation and self optimization, high availability
- End-to-end service assurance with better Quality of Experience
- Agile development and faster innovation
- Path to service agility, new service integration
- Integration into one management and control platform
- Central Office/Head End consolidation and power/space optimization
- Reduced cost (use of common reusable hardware and open software)
- New revenue opportunities

The NFV Fabric effort will deliver an SDN and NFV based switch architecture to connect edge services to the backend architecture by meeting these objectives. The key elements are:

1. SDN controlled fabric must support End-Hosts (servers) and containers for forwarding traffic seamlessly. Dual-homing support for connectivity to hosts and upstream routers is required. Access devices are single-homed to the leaf-switches.
2. An open interface and protocol (e.g. OpenFlow, P4RunTime) should be used to program forwarding rules in the switching fabric
3. SDN controller must support applications that support YANG model configuration using NETCONF (or gNMI) for additional services in Fabric switches such as IEEE 1588v2, MAC-SEC.
4. IPv4 and IPv6 unicast traffic forwarding in fabric is required. The use of ACLs for blocking or overriding default routing decisions must be supported
5. Network, subscriber and service connectivity applications such as DHCP Relay for directly and indirectly connected hosts, AAA for certificate authentication, RIP and PD stability should be implemented through SDN controller apps in containers
6. Multicast IPv4 and IPv6 data traffic must be supported through the fabric. Multicast control plane protocols (IGMP, PIM) are optional
7. Traffic forwarding based on VLANs and bridging must be supported
8. vRouter solution should be integrated for external connectivity, where the entire fabric acts as a distributed virtual router data-plane, and control-plane E-BGP protocol should be supported for interacting with external routers.
9. Security features for authentication of HW and SW components, secure connections among these components for data and control domains should be supported
10. Service assurance is aimed for all HW and SW components through push based telemetry, logging and debugging functionalities
11. Fabric should be capable of multi-stages, for example leaf-spine-spine-leaf
12. Troubleshooting tools should be natively supported by the fabric

The following key characteristics are defined for an architecture with lowest Total Cost of Ownership for service providers:

1. A modular design with automated management and configuration
2. Secure and reliable infrastructure with self-recovery capabilities
3. Flexible deployment support with up/down scaling options
4. Modular HW and SW design where disaggregated components may be added/removed based on operators' needs
5. High performance with load balancing and self-optimization capabilities
6. A common architecture that can support multiple access technologies and edge services in the southbound and multiple orchestration

platforms in the northbound
7. Virtual identification (instead of physical location based identification)

2.2 MAJOR FUNCTIONAL COMPONENTS

2.2.1 **Trellis Infrastructure: Hardware and Software**

The Infrastructure layer includes the hardware including the devices (switches, compute servers), racks and shelves, powering equipment and connections, and external fibers or electrical cables, and other passive devices.

The Software includes the SDN controller, applications, VNFs, Kubernetes platform, switch software (OS, boot-loaders, drivers etc.), hardware abstraction layer agents and access integration specific SW.

Trellis spine-leaf fabric should be controlled by the SDN controller via interfaces such as OpenFlow, NETCONF and/or P4 RunTime. An example controller and switch software stack used in the Trellis Exemplar platform is displayed in Figure 3 and Figure 4 respectively. Future versions may adopt other interfaces as well. Trellis also provides a vRouter solution making the entire fabric act as a distributed virtual router data-plane for external connectivity.

SDN controller applications include authentication and secure connections and communications among hardware and software components (Figure 5). In this context, certification authentication (e.g. 802.1X authentication), encryption (e.g. IPsec, TLS, MACsec) and command authorization (e.g. Kubernetes audits, Kafka audits, TACACS+) are defined among a combination of PNFs and VNFs. SDN controller applications also include subscriber and service managements functionality such as L3 DHCP Relay, ARP, NDP and RA exchanges, unicast and multicast route updates, RIP and PD stability and host connectivity matrix. The architecture is modular based on containerization of VNFs defining microservices that can be added and removed based on the service provider needs.

Service assurance is provided by push-based telemetry, logging and debugging platform. Monitoring data is streamed over real-time data pipelines and streaming applications platform (e.g. Kafka). Monitoring data is stored, processed and visualized by using platforms with dimensional data modeling, flexible query language, efficient time series database and alerting capabilities (e.g. Prometheus). Visualization can be provided by effective dashboards (e.g.

Grafana). Logging data is queried through an effective search engine (e.g. ElasticSearch) and visualized as dashboards (e.g. Kibana). Artificial intelligence techniques enable effective ways to connect raw data to high level events for proactive and predictive maintenance, troubleshooting, network planning and service agility. The scope of this RD is to define the support of PNFs and VNFs for the monitoring, logging and debugging data. However, the storing and processing platforms are considered out of scope for this Reference Design.

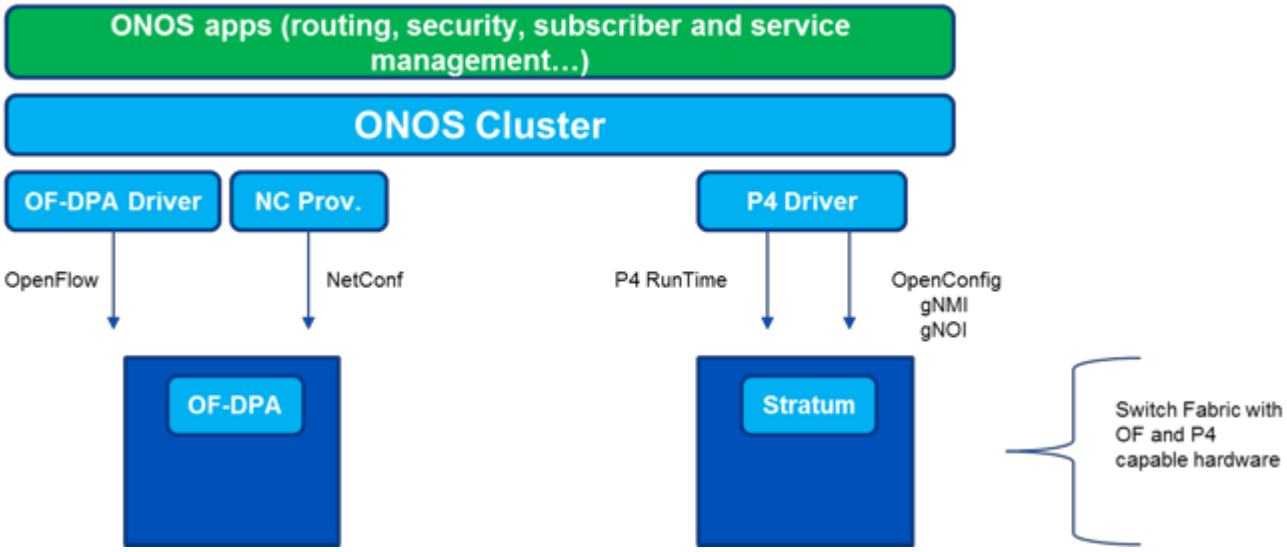

Figure 3 High Level Trellis Exemplar HW and SW Components

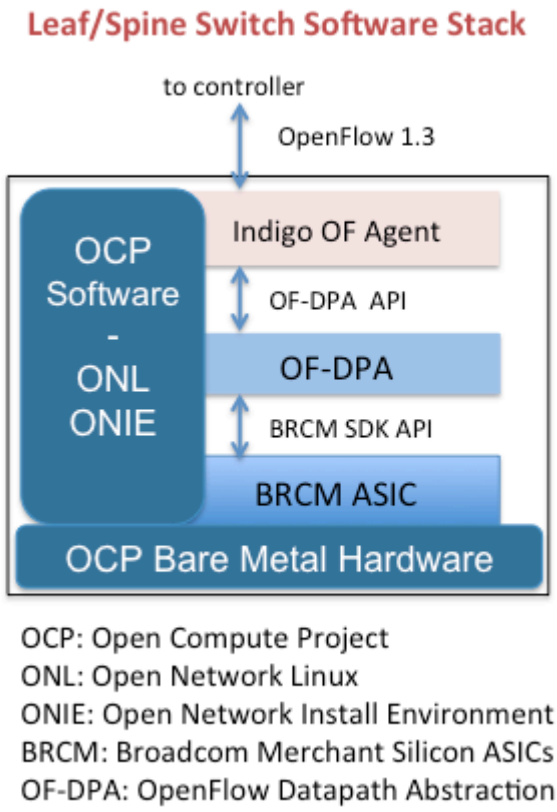

Figure 4 Example Leaf/Spine switch software stack

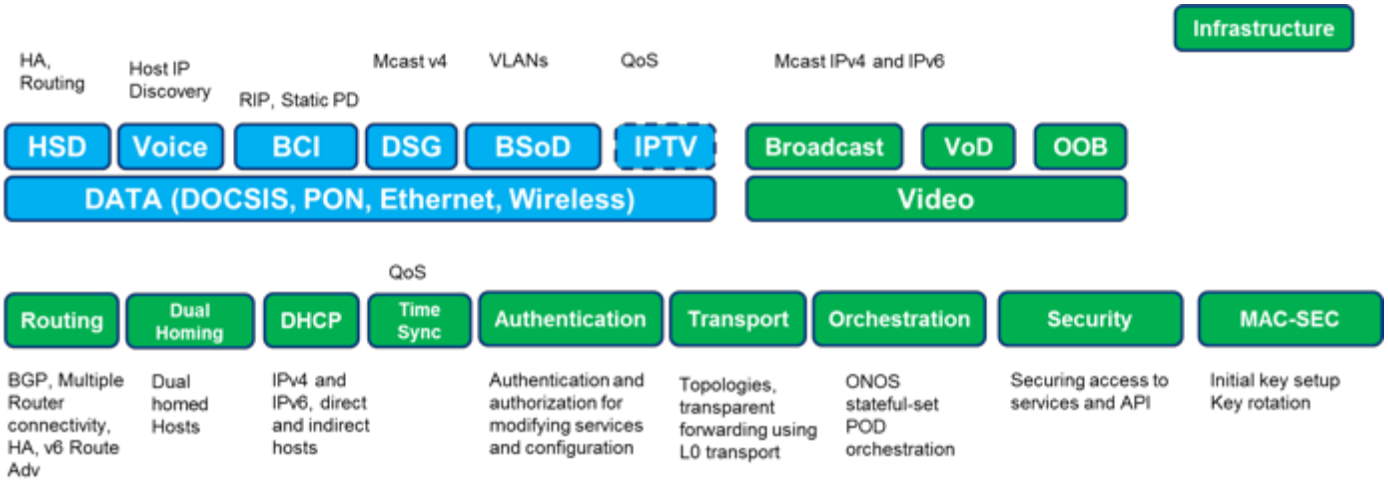

Figure 5 Access Network Services virtualized and underlying protocol stack

2.2.2 **Infrastructure Requirements**

Scaling

The scalability requirement is deployment dependent. It depends heavily on the implementation network architecture and services offered from the POD. The Trellis community shall conduct the scalability study and provide the deployment guidelines which meet the service provider implementations.

The typical edge deployments of service providers at central offices / cable head-ends in a single SDN controller domain is likely to be in the range of:

- *15k – 20k subscribers*
- *About 150k routes (IPv4 and IPv6)*
- *About 400 Remote digital nodes (R-PHY or OLTs)*

Trellis POD must support scaling the POD elements horizontally to accommodate growing from the initial deployment to a larger footprint by adding the access nodes and resources.

Vertical scaling of a POD means adding more resources to VNFs or into operating hardware. Horizontal scaling is preferred in clouds to simplify operations. The application of vertical scaling must be justified as an alternative to horizontal scaling.

Resource management

Resource management includes compute, memory, storage and network connectivity and bandwidth required for the initial POD installation, and the modeling and capacity management plan as additional utilization grows by access nodes and subscribers.

Data Collection

The status and the health of the Trellis POD, including all physical and local components and services, must be monitored and provide on-demand and periodic reporting mechanism to the orchestration platform through the Trellis NBI Client.

Operation configurable Threshold Crossing Alert (TCA) must be utilized for raising the event to inform the operator about the state of the POD elements.

Overlay networking requirements

Trellis POD must provide secure communication channels internally within the POD and externally to the provider's management network. The data path channel must not be compromised and provide unlawful access to the management and control channels into the POD infrastructure and provider's management network.

Speeds/feeds, performance (e.g., compute and I/O)

The physical connectivity between the POD and the access network and backbone must maintain the SLA of the services offered of the POD. The connectivity requirement may vary based on the overall access network architecture design and the type of the services offered. The Trellis exemplar implementation of a provider will determine the requirements. The requirement shall define whether it is Layer 2 and/or Layer 3 connections.

Defined by the implementation, the oversubscription over the physical bandwidth may be required.

The Trellis project shall provide the compute and storage performance based on the provider's implementation and scalability requirements.

POD Assembly

The assembly of a POD includes the definition of the virtualization approach of the software to the hardware resources.

The exemplar platform begins with a set of container elements run in a Kubernetes environment to optimize the utilization of compute resources.

Large-scale orchestration and lifecycle management of PODs from an operator's cloud environment may lead to approaches to standardize on other variants of infrastructure environments.

2.2.3 **Use Cases and Flow**

The Trellis project wiki page hosts the Trellis use cases and requirements from multiple Service Providers and will become the basis for the Trellis implementation streams.

3 TIME TO MARKET SOLUTIONS

The architecture can support options based on different needs of operators. It is highly desired that operator variants change the functional components rather than the interfaces between the components.

3.1 SOLUTION ELEMENTS

SERVICE AND NETWORK ORCHESTRATION PLATFORM

The service and network orchestration platform is external and northbound of the Trellis architecture and out of scope of this Reference Design. Subcomponents may include service and resource modeling for agile service management, service and device life cycle management and inventory and subscriber data management. These components may be hosted in the cloud and may communicate with other components within the POD structure of the Trellis platform.

3.1.1 **Infrastructure Layer**

The Infrastructure layer includes the hardware including the devices (switches, compute servers), racks and shelves, powering equipment and

connections, and external fibers or electrical cables, and other passive devices.

Trellis infrastructure is designed as PODs architectures. PODs can be distributed and managed by a central POD that interfaces service and network orchestration platforms; or each POD can be managed independently. PODs include fabric switches, compute servers and may include devices specific to an access technology. PODs are connected to a central POD or other PODs, to the service and network orchestration platform, to data management platforms and access infrastructures.

The combination of components in the Infrastructure layer compose the physical inventory of the solution that suppliers plan for delivery, fulfillment solution providers aggregate in a supply chain, and that installers place and validate.

Main hardware components are: underlay fabric, compute servers and access infrastructure. Access infrastructure is out of scope for this Reference Design but Trellis architecture is designed to enable its integration. In one architecture, an access HW may be connected to a switch fabric within the POD while in another, it may be connected over a fiber link to a distant location closer to the served customers. Similarly, an access aggregation switch (Leaf switch) may be located in the CO/headend within the POD with other fabric switches, or in the secondary hub or in the field.

The infrastructure scale at the central office / head-end depends on the specific scaling needs and can vary widely across service Provider environments. A typical SDN controller domain at the Access Edge is likely to have approximately 100 compute servers, 12 – 16 edge switches and 2 spine switches. It is also possible that edge switches at the secondary locations are aggregated via an aggregation switching layer. Depending on the density, it is also possible to host multiple SDN controller domains at the same physical site.

An example deployment option is shown in Figure 6.

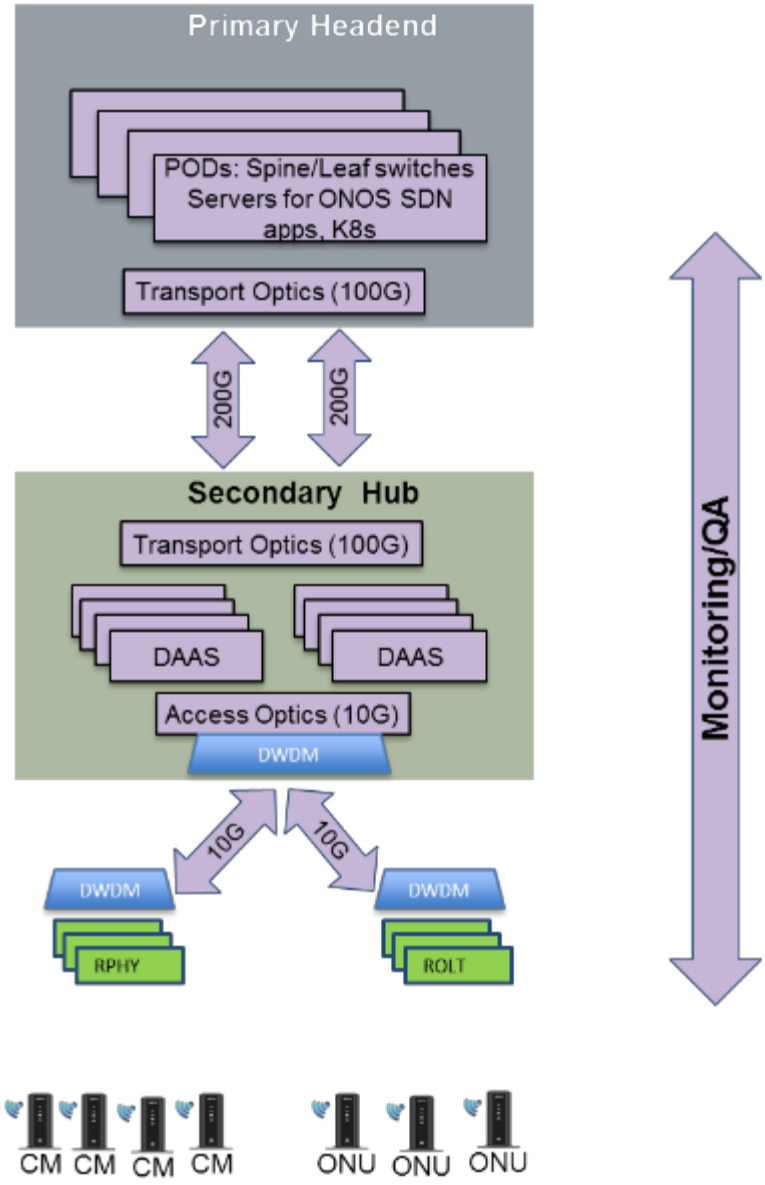

Figure 6 Example Deployment of the Trellis Platform

3.1.1.1 Trellis Underlay Fabric

The SDN-based Leaf-Spine fabric is built with whitebox (OCP Accepted) hardware and open-source switch software. The leaf switches are connected to compute servers, access blades, metro/uplink network equipment and

spines. The spine switches use all of their ports to connect to the leaves. In smaller deployments, more leaf ports can be used to connect servers, and in the extreme case, the spines can be eliminated altogether. Similarly, other deployments can use different switch models.

3.1.1.2 Compute Servers

OCP accepted compute servers should be used for SDN control and applications and access network VNFs.

3.1.1.3 Access Infrastructure

Access infrastructure includes all active and passive devices that are connected to the switch fabric. Although these devices are out of scope of this Reference Design, the requirements for the switches to connect to these devices and for the compute servers that may host SDN functionality to control these devices are defined within the exemplar architectures.

3.1.2 **SDN Control and Application Layer**

Trellis should provide SDN control for the underlay fabric and is responsible for interfacing with conventional upstream routers. It is also responsible for the connectivity between the access infrastructure and related VNFs. It hosts subscriber and service management applications (e.g. DHCP Relay, RA, unicast and multicast routes, host tables, RIP, PD stability). In addition, security features among microservices and between HW and SW components should be supported by applications (e.g. 802.1x authentication, MACsec).

3.1.3 **Kubernetes Platform**

Kubernetes provides a container-centric management environment for microservice platforms. It orchestrates computing, networking, and storage infrastructure on behalf of user workloads. The goal is to have an effective platform to deploy, scale, and manage microservices and applications.

3.1.4 **Installation and Configuration Platform**

Ansible scripts should be used for automated installation and configuration of the Trellis platform. Open industry standards like Redfish may be integrated with Ansible scripts.

3.1.5 **Network VNFs Platform**

The access network VNFs can run in containers on compute servers and may be managed within the same Kubernetes platform. Trellis RD architecture addresses these challenges, and in doing so:

- Simplifies the switching nodes (eliminating the need for complex control and routing protocols in the switches)
- Creates a single routing instance for the entire fabric, simplifying how the fabric connects to the external network
- Provides distributed virtual routing to all tenant traffic in the compute infrastructure, so their traffic can directly reach the external network without having to hairpin to a gateway VM
- Coordinates the underlay and overlay to optimize resource deployment and connectivity
- Uses ECMP routing for multi-pathing, and ACLs for blocking or overriding default routing decisions
- Fully integrates the software switches running on each server, to provide a complete solution for interconnecting servers, applications, VMs and containers with dynamically created tenant domains and service-chains
- Unlike traditional networking approaches, the fabric itself does not run a control protocol (such as BGP, OSPF or RSTP). Instead, all the intelligence is moved into applications running on the clustered SDN controller. In this way the fabric switches can be simplified, the entire fabric can be optimized by using a holistic view of all activity, and new features and functionality can be deployed without upgrading the switches. Trellis will also provide a vRouter solution for external connectivity, where the entire fabric acts as a distributed virtual router data-plane.

3.1.6 **Interior Interfaces**

The functional descriptions of the internal interfaces include:

- NNI interfaces between switches in the fabric
- NNI interfaces between compute servers and switch fabric
- NNI interface between switch fabric POD and transport network
- NNI interface between access aggregation switch and transport network
- NNI interface between transport networks in the secondary/primary hubs/headends
- API interface between SDN controller/apps and switch fabric
- API interface between virtual router and SDN controller/apps
- API interface between switch fabric and monitoring apps
- API interface between compute nodes and monitoring apps

3.1.7 **Exterior Interfaces**

The functional descriptions of the external interfaces include:

- NNI interface between access network and access aggregation switch
- NNI interface between Leaf and Aggregation/Metro Router
- API interface between virtual router and Aggregation Router
- API interface between SDN controller/apps, VNFs and Orchestration layer
- API interface between monitoring apps and telemetry/logging platform

3.1.8 **Security**

As a managed virtualized environment, Trellis should derive and use security requirements, security architecture, security best practices and open source security solutions from other open communities.

The security features include:

- security event monitoring solution
- network access controls & strong authentication on the server BMC interfaces and all other out-of-band management interfaces

- resource quota policies, secured registries and secure APIs on Kubernetes
- mutual authentication between OpenFlow controllers and switches
- mutual authentication between access devices and VNFs
- encryption of data and control path among network segments

3.1.9 **Availability, Reliability and Resiliency**

Availability is percentage of time that the system remains operational under normal circumstances in order to serve its intended purpose. It is typically expressed in terms of number of "9s", such as "five 9s" indicating 99.999% availability. Availability includes planned down time/maintenance windows.

Reliability is the probability of a system or component to function under stated conditions for a period of time. The reliability is also a component of the availability of a system or component and is typically expressed in Mean Time Between Failure (MTBF).

A reliability analysis is recommended for the components of a POD, and of the entire POD, in order to predict the availability of the POD as a system.

Resiliency is the ability of a server, network component, or POD to recover from a failure (such as a power failure, or equipment failure) and quickly resume operations.

Redundancy and/or clustering is employed to help improve the availability of a POD.

3.1.10 **System Performance**

System performance measurements in the internal processing functions of the Pod are important to make and understand and should include system response for control and management transactions, and scalability of the system to provide a determinate number of operations of a certain type while the system is operating under a defined load profile (profile of a variety of defined operations at defined frequencies over a defined period of time).

The implementation of monitoring tools to monitor system performance must be considered and selected in order to minimize their own impact on system performance and resources.

3.1.11 In band Management

SDN controller controls and manages the switches via an Out-Of-Band management channel. This typically works for the switches in the primary location as it is feasible to setup an OOB management network at the primary locations. In remote secondary locations that are typically connected via DWDM optical transport, it is expensive and in-feasible to have an OOB management connectivity. Hence it is desirable to have an in-band management connectivity.

The in-band management typically requires an ability to identify the management traffic in the data plane (typically via VLAN) and move the management traffic between the forwarding plane ASIC and the CPU of the switch. It is also feasible to setup redundant in-band management paths across the fabric to ensure high availability

3.1.12 Capacity Management

Capacity management provides analytics and reporting of the resources needed for a solution. It derives capacity measurements from the Performance Management element and from the resources that define the capacity of a solution element.

The implementation of capacity management will occur in the Service and Network Orchestration layer that receives performance measurements from the Pod(s), and so the Performance Management collection requires forwarding to the CAP for that function.

A provider may determine that Capacity Management functions are implemented in the local pod and provided through a local operator interface.

EMS may be used to transform the faults to a desired format used by other OSS systems.

3.1.13 Configuration

Automated configuration of Trellis components are within the scope of this Reference Design (i.e. using Ansible, Redfish etc,) while configuration of access devices, network VNFs, service and subscriber profiles are out of scope.

The configuration platform must provide the capability to periodically collect configuration information from each of the POD components and export them to another safe location. Then, in the event of a software or equipment failure it must be possible to restore the Trellis POD using the backed-up system configuration information. The restore process shall be able to be monitored for progress and success.

3.1.14 Software Lifecycle Management

Trellis shall provide APIs capable of managing software for the components of the POD. This includes the physical equipment resident in the POD as well as all the applications.

During an upgrade APIs will be provided to track the progress. The APIs shall be able to determine the success of the upgrade activities and identify any fallout activities which are required.

Rollback to previous software components must be possible.

Activities impacting customer service shall be performed in maintenance periods, usually 2 to 4 hours. Expected interruption to subscriber service shall be less than 5 minutes.

3.1.15 Accounting and Status

Access network device and VNFs real-time status are within the scope of this Reference Design while accounting of subscribers is out of scope.

3.1.16 Fault and Performance Management

Different APIs (e.g. Thrift, gRPC, Redfish etc.) may be used to export data to a Kafka topic and provide all performance metrics and faults through a normalized collector in Kafka. A subsystem may be used to transform the metrics, alarms and events to a desired format used by other OSS systems. An example platform is shown in figure 7.

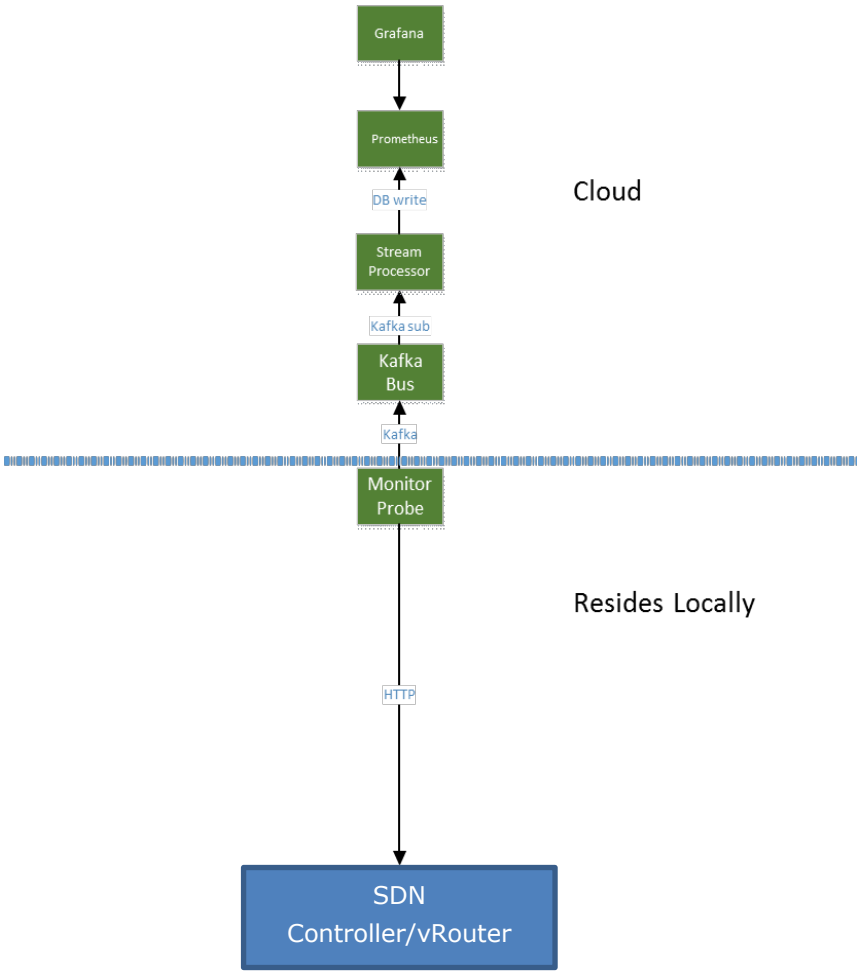

Figure 7 Example platform for fault and performance management

3.1.17 Inventory

Inventory is the system components and interfaces. In a life cycle view, the planned inventory includes the expected components and interfaces that need to be operating in the infrastructure layer to deliver the POD functions. The actual or discovered inventory requires discovery and validation against the planned inventory in order to provide the required infrastructure for services and operations of the POD.

3.1.18 Automation and Management

POD Management and Status Reporting

- Provide inventory information for common hardware components
- Download and manage software upgrades for SDN controller, SDN Apps, and other VNF components
- Monitor common hardware resources
- Provide detail views on CPU utilization, component states, POD status, Container status
- Status Reporting
- Alarm Management
- Performance Monitoring
- Profile Management
- Define and manage technology profiles required for various hardware adapters
- Access device, service and subscriber management integration

3.2 SUPPORTING ACTIVITIES

3.2.1 Operational Plan

3.2.1.1 Physical Environment

The physical environment for a Trellis POD is likely to be mostly in a Central Office (e.g. Headend), but operators may pursue options to

deploy some elements in the secondary hubs or field. Some applications may be hosted in the cloud and integrated with Trellis POD platform.

3.2.1.2 Physical Requirements

The physical requirements for a Trellis POD cover multiple areas such as space, rack placement, power, operating temperature ranges, cooling and heat dissipation. The requirements may change per operator.

3.2.2 **Ecosystem Component Assessment**

3.2.2.1 Open Source SW

Open source software should follow the guidelines of ONF as to the open software licenses that ONF projects can use to incorporate open source.

Operators and members contributing code to the ONF open source are responsible to conform to the guidelines of their companies or organizations to contribute code to ONF.

3.2.2.2 Open HW

The classification of open hardware will follow the definitions within the OCP, such as the “OCP Accepted” and “OCP Inspired” trademark definitions.

3.2.2.3 Functional Decomposition Supplier Consistency

Supplier consistency in the functional decomposition of the Trellis exemplar platform and into specific implementations is desired and encouraged in order to align with suppliers and developers, while not limiting innovation to improve cost, performance or reliability.

The operators, suppliers and developers should proactively collaborate to communicate about implementations and product roadmaps and evolving open software technologies, and to propose the controlled evolution of solutions.

3.2.3 **Operator Specific Addendum (system impacts, etc).**

Operators should provide specific addenda here that require special attention to the Trellis project, as derived out of their experience, requirements, or consequences of their Trellis implementations.

3.2.4 **Key outstanding Questions**

There are currently no open questions.

End of

Trellis Reference Design

Write to rdspec@opennetworking.org with comments or questions.