TRELLIS PLATFORM BRIEF

ONF's Trellis is the leading open-source multi-purpose leaf-spine fabric supporting distributed access networks, NFV and edge cloud applications. Built using white-box switches, merchant silicon and SDN principles, Trellis is scalable, flexible, cost-effective and production-ready. It is deployed in multiple geographies by a Tier-1 US network operator.

BENEFITS

- Production-ready fabric optimized for Service Provider networks
- Flexible, easily extensible and configurable using classic SDN methods
- Capex and Opex savings with white-box hardware and open source software
- Offers horizontal scaling and redundancy
- Future proof: P4 and Stratum integration to unlock advanced capabilities

KEY USE CASES

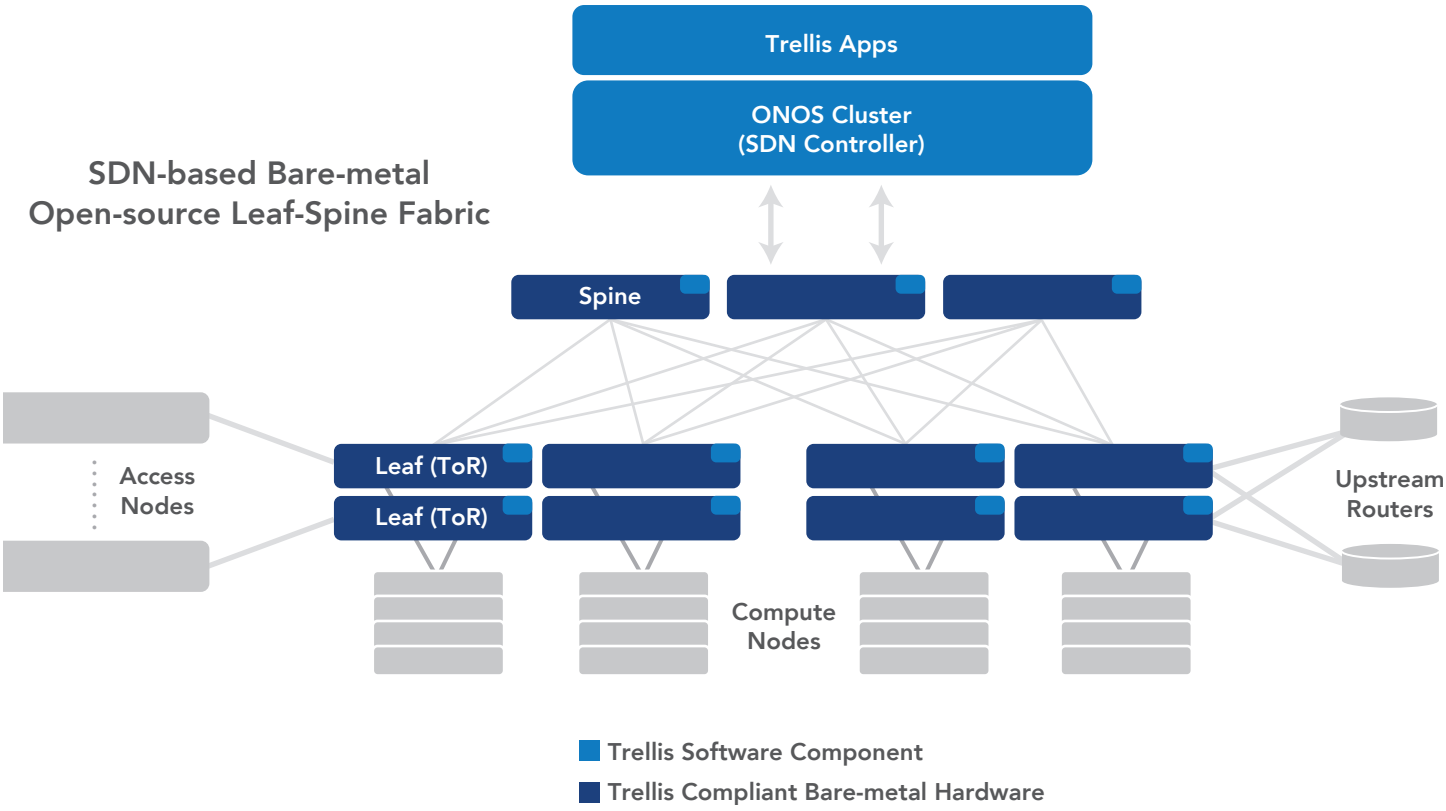
- Distributed Fabric for SP Access/Edge Networking
- Disaggregated BNG in SEBA Using P4
- Enterprise Datacenter Fabrics

Overview

Trellis is an open-source multi-purpose L2/L3 leaf-spine switching fabric. The development of Trellis over the last four years has been influenced by three core trends in the networking industry:

- **First, Trellis is built using bare-metal switches with merchant-silicon ASICs.** Instead of using OEM networking hardware, Trellis uses hardware directly from ODMs. The trend of using bare-metal (white-box) switches is unmistakable in the networking industry today, spurred by the massive bandwidth-density and growing sophistication of merchant silicon ASICs. Production quality Trellis today is based on EdgeCore switches with Broadcom Trident2, Tomahawk and Qumran switch ASICs. The Trellis team continues to work towards including more ODMs and merchant silicon vendors.
- **Second, Trellis is based on SDN principles, to provide simpler, more flexible and easily customizable networks.** By externalizing the network's control, management functions and policy decisions in the ONOS SDN controller, Trellis provides network operators with a number of SDN benefits compared to traditional box-embedded network control. These include centralized configuration, automation, operation and troubleshooting.
- **Third, Trellis is open-source.** The networking industry has seen an explosion of open source projects, and network operators have been eager to embrace open-source solutions. Trellis allows operators unparalleled ability to customize Trellis for their application, integrate with the rest of their systems, add features and APIs themselves and not be beholden to a traditional vendor's timelines and prices. An absence of commercial licenses lowers the bar for anyone to try out Trellis.

Together, all three attributes of Trellis considerably lower the Total Cost of Ownership (TCO) for operators who plan to run it in production.



Architecture

Classic-SDN. Trellis operates as a hybrid L2/L3 fabric. As a pure (or classic) SDN solution, Trellis does not use any of the traditional control protocols typically found in networking, a non-exhaustive list of which includes: STP, MSTP, RSTP, LACP, MLAG, PIM, IGMP, OSPF, IS-IS, Trill, RSVP, LDP and BGP. Instead, Trellis uses an SDN Controller (ONOS) decoupled from the data-plane hardware to directly program ASIC forwarding tables using OpenFlow and with OF-DPA, an open-API from Broadcom running on the switches. In this design, a set of applications running on ONOS implement all the fabric functionality and features, such as Ethernet switching, IP routing, multicast, DHCP Relay, pseudowires and more.

Bridging and Routing. The main fabric-control application running on ONOS programs L2 forwarding (bridging) within a server-rack, and L3 forwarding (routing) across racks. The use of L3 down-to-the Top-of-Rack switches (ToRs) is a well-known concept in leaf-spine fabrics, mainly due to better scalability of L3 over L2. In such cases, the ToRs (leaves) route traffic by hashing IP flows to different spines using ECMP forwarding—essentially every ToR is 2-hops away from every other ToR in a 2-stage Clos, and there are multiple such equal-cost paths. Trellis supports routing both IPv4 and IPv6 traffic using ECMP groups.

The need for supporting bridging and access/trunk VLANs comes out of the use of Trellis in NFV infrastructures, where VMs (VNFs) designed for legacy networks often expect to communicate with each other at L2, and also send and receive multiple VLAN tagged packets. When necessary, such VMs can be allocated on servers in the same bridging domain. In other cases, Trellis can also be configured to operate completely in L3 mode.

MPLS Segment Routing. Trellis architecture internally uses Segment Routing (SR) concepts like globally significant MPLS labels that are assigned to each leaf and spine switch. This minimizes label state in the network compared to traditional MPLS networks where locally-significant labels have to be swapped at each node. In Trellis, the leaf switches push an MPLS label designating the destination ToR (leaf) onto the IPv4 or IPv6 traffic, before hashing the flows to the spines. In turn the spines forward the traffic solely on the basis of the MPLS labels.

This design concept, popular in IP/MPLS WAN networks, has significant advantages. Since the spines only maintain label state, it leads to significantly less programming burden and better scale. For example, in one use case, the leaf switches may each hold 100k+ IPv4/v6 routes, while the spine switches need to be programmed with only 10s of labels! As a result, completely different ASICs can be used for the leaf and spine switches—the leaves can have bigger routing tables and deeper buffers while sacrificing switching capacity (example Broadcom Qumrans), while the spines can have smaller tables with high switching capacity (example Broadcom Tomahawks).

External Connectivity with vRouter. Trellis assumes that the fabric connects to the public Internet (or other networks) via traditional routers. This external connectivity is handled by a feature known as Trellis vRouter. In contrast to a software virtual-router (VNF), Trellis vRouter performs only control-plane functionality in software. It uses a routing protocol stack (like Quagga or FRR) to speak BGP or OSPF with the external router, while the data-plane functionality of the router is performed by the fabric itself. In other words, the entire leaf-spine fabric behaves as a single (distributed) router data-plane and is viewed as such by the external world. The Trellis vRouter is a big multi-terabit-per-second distributed router, where the leaf-switches act like line-cards, and the spine switches act like backplanes of a traditional big-box chassis router, with obvious cost and performance advantages over chassis-based routers or software router VNFs respectively.

Redundancy. Trellis supports redundancy at every level. A leaf-spine fabric is redundant by design in the spine layer, with the use of ECMP hashing and multiple spines. In addition, Trellis supports leaf-pairs, where end-hosts can be dual-homed to two ToRs in an active-active configuration. In the eventuality that a ToR dies, it's pair continues to support traffic flow. In certain configurations, operators may choose to single-home connected equipment to only one leaf/ToR—this is typical in NFV infrastructure where Access nodes (R-PHYs, OLTs, eNodeBs) have a single wired connection to the operator's facility. Trellis supports both single and dual homing, combinations of which can be used in the same infrastructure. Additionally, Trellis also supports dual-homing of the external connectivity—Trellis can appear to external routers as two BGP-speaking routers in well-known redundant configurations.

Many SDN solutions use single-instance controllers, which are single-points of failure. Other solutions use two controllers in active-backup mode, which is redundant, but may lack scale as all the work is still being done by one instance at any time and scale can never exceed the capacity of one server. In contrast, Trellis is based on ONOS, an SDN controller that offers *N-way redundancy and scale*. An ONOS cluster with 3 or 5 instances are all active nodes doing work simultaneously, and failure handling is fully automated and completely handled by the ONOS platform.

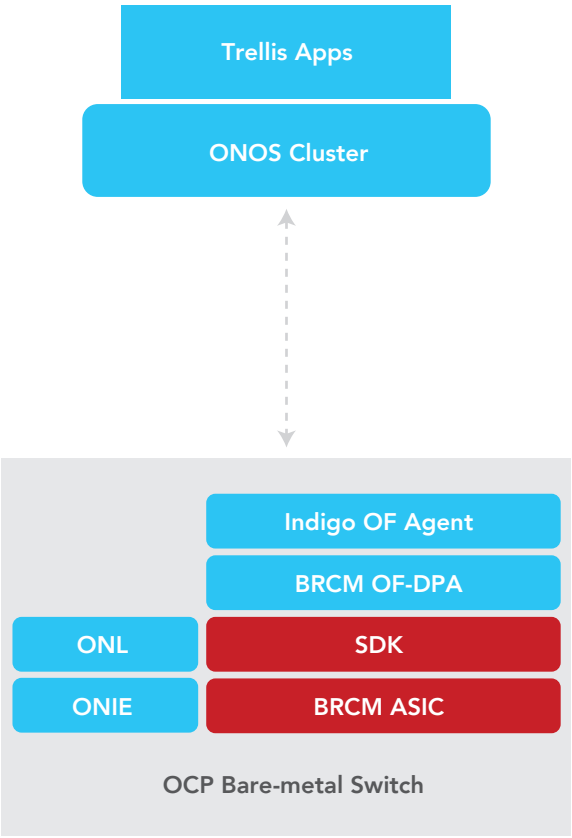
Topologies. Trellis supports a number of different topological variants. In its simplest instantiation, one could use a single leaf or a leaf-pair to connect servers, external routers and other equipment like access nodes or physical appliances (PNFs). Such a deployment can also be scaled horizontally into a leaf-and-spine fabric (2-level folded Clos), by adding 2 or 4 spines and up to 10 leaves in single or paired configurations.

SALIENT FEATURES

- **Classic-SDN Control with ONOS to directly program ASIC forwarding tables in bare metal switches with merchant silicon**
- **L2 forwarding (Bridging) within server-racks and L3 forwarding (Routing) across racks**
- **MPLS Segment routing for better scale and reduced programming**
- **Control plane functionality with Trellis vRouter for external connectivity**
- **N-way redundancy and tier-1 telecom operator scale**

In a special configuration, a Trellis fabric can itself be distributed across geographical regions, with spine switches in a primary central location, connected to other spines in multiple secondary (remote) locations using WDM links. Such 4-level topologies (leaf-spine-spine-leaf) can be used for backhaul in operator networks, where the secondary locations are deeper in the network and closer to the end-user. In these configurations, the spines in the secondary locations serve as aggregation devices that backhaul traffic from the access nodes to the primary location which typically has the facilities for compute and storage for NFV applications.

Network Configuration & Troubleshooting. A final point worth noting about Trellis operation—all Trellis network configuration and troubleshooting happens entirely in ONOS. In contrast to traditional embedded networking, there is no need at all for network operators to go through the error-prone process of configuring individual leaf and spine switches. Similarly, the SDN controller is a single-pane-of-glass for monitoring and troubleshooting network state. Trellis provides in-built tools for statistics gathering and connectivity diagnosis, which can further be integrated with operator's backend systems.



System Components

Open Network Operating System (ONOS). Trellis uses Open Network Operating System (ONOS) as the SDN controller. ONOS is designed as a distributed system, composed of multiple instances operating in a cluster, with all instances actively operating on the network while being functionally identical. This unique capability of ONOS *simultaneously* affords high-availability and horizontal-scaling of the control plane. ONOS interacts with the network devices by means of pluggable southbound interfaces. In particular, Trellis leverages OpenFlow 1.3 protocol for programming, and NETCONF for configuring certain features (like MacSec) in the fabric switches. Like other SDN controllers, ONOS provides several core services like topology discovery, and end-point discovery (hosts, routers etc attached to the fabric). Unlike any other open source SDN controller, ONOS provides these core services in a distributed way over the entire cluster, such that applications running in any instance of the controller have the same view and information.

ONOS Applications. Trellis uses a collection of applications that run on ONOS to provide the fabric features and services. The main application responsible for fabric operation handles connectivity features according to Trellis architecture, while other apps like dhcprelay, aaa, multicast, etc. handle more specialized features. Importantly, Trellis uses the ONOS Flow-Objectives API, which allows applications to program switching devices in a pipeline-agnostic way. By using Flow-Objectives, applications can be written without worrying about low-level pipeline details of various switching chips. The API is implemented by specific device-drivers that are aware of the pipelines they serve, and can thus convert the application's API calls to device specific rules. This way the application can be written once, and adapted to pipelines from different ASIC vendors.

Leaf & Spine Switch Hardware. In a typical configuration, the leaf and spine hardware used in Trellis are Open Compute Project (OCP) certified switches from a selection of different ODM vendors. The port configurations and ASICs used in these switches are dependant on operator needs. For example, one such application uses EdgeCore 5912 switches with Broadcom StrataDNX ASICs (Qumran) as leaf switches with 48 x 10G ports and 6 x 100G ports, and EdgeCore 7712 switches with Broadcom StrataXGS ASICs (Tomahawk) as spine switches with 32 x 100G ports. However, other use cases and operators have used Tomahawks or Barefoot Tofinos as leaf switches, or even Trident2 based 1G or 10G switches.

OF-DPA & Indigo. In production, Trellis leverages software from Broadcom's OpenFlow Data Plane Abstraction (OF-DPA) project. OF-DPA implements a key adaption role between OpenFlow (OF) 1.3 protocol constructs and Broadcom's proprietary ASIC SDKs. In particular OF-DPA presents an abstraction of the switch ASIC pipeline and hides the differences between the actual ASIC pipelines (Broadcom Strata XGS or DNX) underneath, so that an open-source OF protocol agent like Indigo can be layered on top of the API. With this layering, an external SDN controller (like ONOS) can access, program and leverage the full capabilities of today's modern ASICs, a fact that proves crucial in providing dataplane scale and performance. For trials, ONF provides a free downloadable binary for the community-version of OF-DPA. In production, a commercial version of OF-DPA is available from Broadcom.

ONL & ONIE. Trellis switch software stack includes Open Network Linux (ONL) and Open Network Install Environment (ONIE) from OCP. The switches are shipped with ONIE, a boot loader that enables the installation of the target OS as part of the provisioning process. ONL, a Linux distribution for bare metal switches, is used as the base operating system. It ships with a number of additional drivers for bare metal switch hardware elements (eg. LEDs, SFPs), that are typically unavailable in normal Linux distributions for bare-metal servers (eg. Ubuntu).

Stratum. (optional) Trellis also supports switch software from the ONF Stratum project. Stratum is an open source silicon-independent switch operating system. While the integration of Stratum in Trellis is preliminary and not ready for production, Stratum is envisioned as a key software component of SDN solutions of the future. Stratum implements the latest SDN-centric northbound interfaces, including P4, P4Runtime, gNMI/OpenConfig, and gNOI, thereby enabling interchangeability of forwarding devices and programmability of forwarding behaviors. We discuss Trellis leveraging Stratum and P4 in more detail in the Use Cases section.

Kubernetes & Kafka. (optional) While ONOS instances can be deployed natively on bare-metal servers, there are advantages in deploying ONOS instances as containers and using a container management system like Kubernetes (K8s). In particular, K8s can monitor and automatically reboot lost controller instances (container-pods), which then rejoin the ONOS cluster seamlessly. Additionally, a next-gen version of ONOS known as µONOS (pronounced "micro-ONOS") is composed of several micro-services which can be optimally deployed and managed by K8s. Finally, the use of a messaging bus like Kafka can help integration with operator's backend systems, where ONOS and Trellis applications publish statistics and events on to Kafka for post-processing by operator OSS/BSS systems.

Benefits

Optimized for Service Provider Applications. Service provider networks tend to use more complicated header and tunneling formats that the typical enterprise datacenter. For example, residential subscriber traffic in Fiber-to-the-Home (FTTH) networks often is double VLAN tagged (QinQ) with PPPoE encapsulation, whereas mobile subscriber traffic from eNodeBs are GTP tunnel encapsulated. Enterprise subscriber traffic could be VXLAN encapsulated or pure IP, whereas cable network traffic can be L2TP encapsulated. Trellis includes a number of features designed to handle these different types of traffic and encapsulations.

Furthermore, Trellis includes the capability for the same fabric to be geographically distributed where subscriber traffic can be aggregated from multiple secondary locations, and delivered for processing by virtual-applications (VNFs) at the primary location—a backhaul service. And it supports multiple different types of ASICs for different applications—both typical datacenter designed ASICs like Trident/Tomahawk and Tofino, but also ASICs with bigger tables, deeper buffers and many QoS capabilities like Qumran.

Trellis is a hardened and production-tested platform, delivering broadband to thousands of paying subscribers of a large tier-1 operator.

These features, topologies and some ASICs are not typically found in fabric offerings in the industry, which tend to be localized, pure L2 or L3 fabrics meant only for enterprise datacenters. While Trellis can also be used as a pure L3 Clos, these additional capabilities make it ideal in NFV infrastructures, for access and edge network backhaul, and for Service Providers that aim to modernize their infrastructure with edge-cloud capabilities. This domain is the central idea behind ONF's flagship umbrella project—Central Office Re-architected as a Datacenter (CORD). Indeed Trellis is central to the CORD network infrastructure.

Offers All SDN Benefits. As Trellis is based on classic SDN, it comes with a number of benefits purely from this design choice.

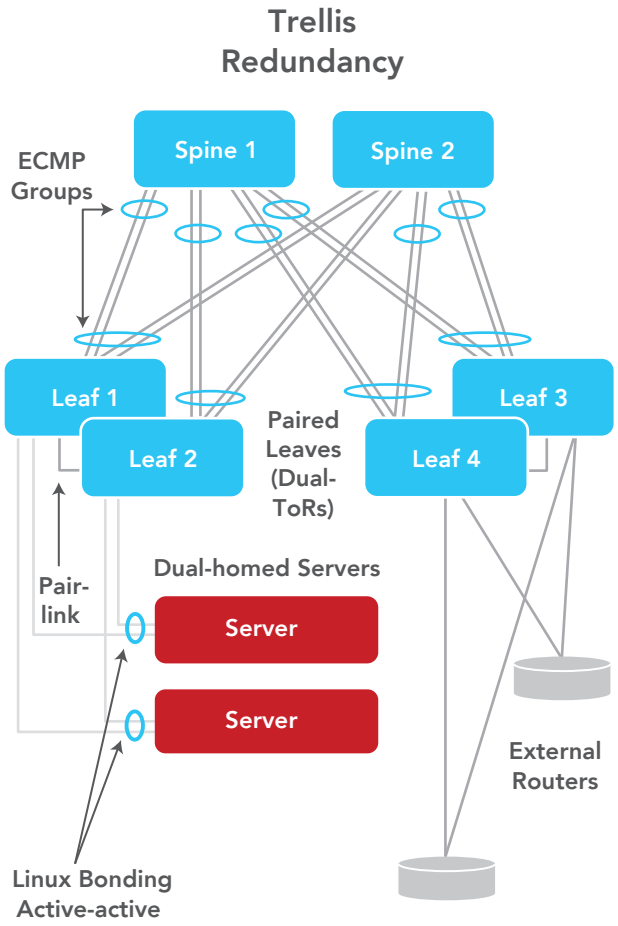
- **SDN simplifies the development** of number of features compared to traditional networking. For example, when Trellis vRouter is integrated with overlay-networks in cloud applications, it allows any VM or container to directly communicate with the outside world, without having their traffic be hair-pinned through some other gateway VM, as is prevalent in many other overlay-only networking solutions. This is an example of Distributed Virtual Routing (DVR). Another example of a feature simplified by SDN is the handling of IPv4 Multicast for IPTV service to residential subscribers. When users flip channels on their TV service, the home set-top box sends an IGMP join message upstream requesting the IP multicast stream corresponding to the channel selected. In response the multicast app directly programs the fabric to replicate and deliver the stream

to the subscribers that have requested the same channel. Such SDN optimized handling of multicast is far simpler operationally than the successive IGMP/PIM filtering and multicast-tree formation performed by independent Ethernet switches and IP routers running embedded multicast protocols in traditional networks.

- **SDN control provides flexibility** in the implementation of features by optimizing for a deployment use case. For example, in one deployment some leaf switches could be programmed with 100k+ routes and others with only a few thousand routes. By contrast if the fabric was using a distributed routing protocol like OSPF, all leaves would get all the routes. In general, SDN allows faster development of features (apps) and greater flexibility to operators to deploy only what they need and optimize the features the way they want.
- **SDN allows the centralized configuration** of all network functionality, and allows network monitoring and troubleshooting to be centralized as well. Both are significant benefits over traditional box-by-box networking enabling faster deployments, simplified operations and empowering tier-1/tier-2 support when debugging production issues.

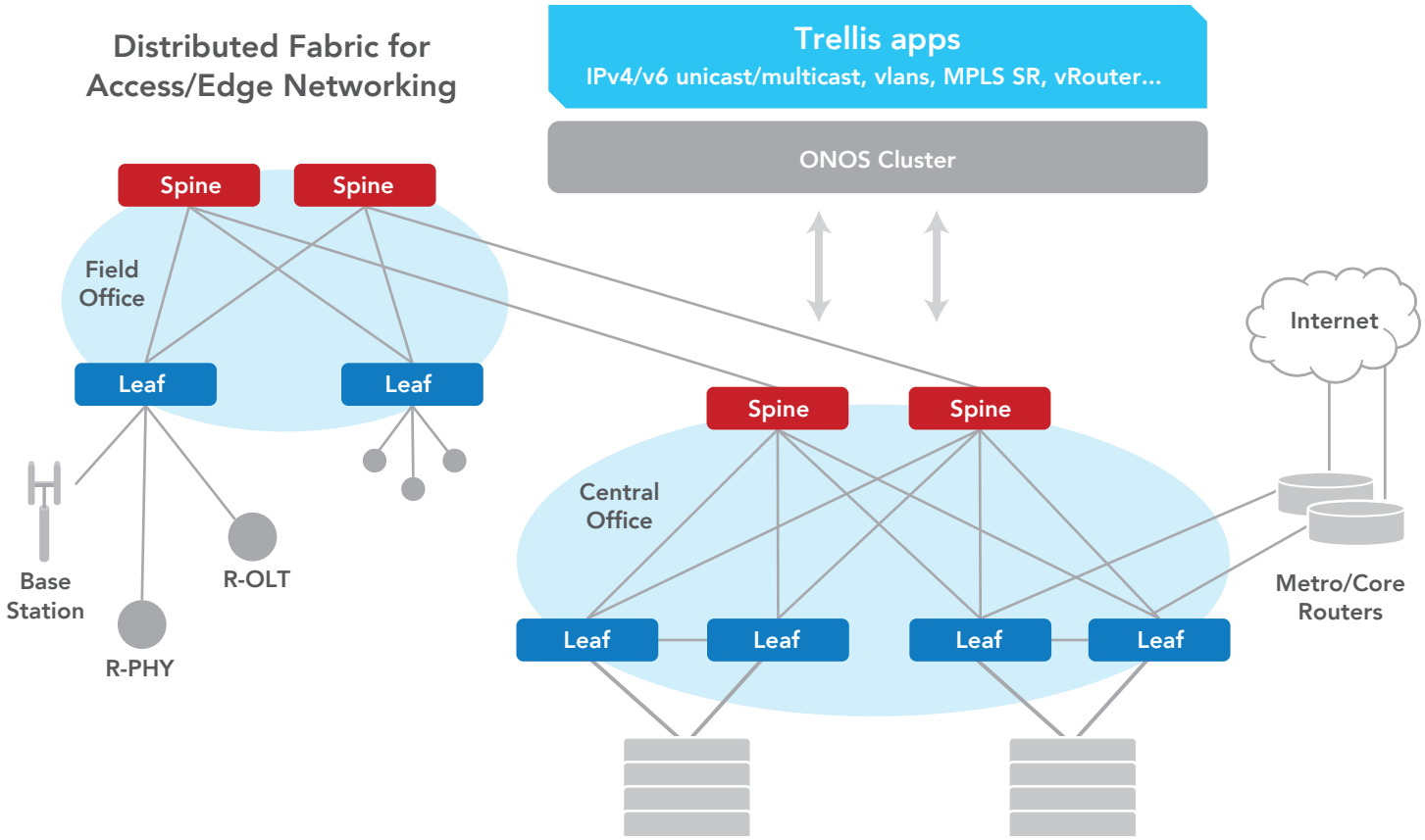
CapEx Reduction. The use of white-box (bare-metal) switching hardware from ODMs significantly reduces CapEx costs for network operators when compared to products from OEM vendors. By some accounts, the cost savings can be as high as 60%. This is typically due to the OEM vendors (70% gross margin) amortizing the cost of developing embedded switch/router software into the price of their hardware.

Benefits of Open Source. Further reduction of CapEx costs can be achieved due to Trellis' use of completely open source, production-ready controller software that is Apache licensed. But more importantly, open-source allows the network operator unparalleled control over their network, the features it includes, various customization and optimizations of those features, and how they integrate with the operator's backend systems. For example, in one deployment the network operator has implemented a full-feature (multi-source multicast) themselves and contributed that work back to open-source. In other instances, they have been able to add rest-apis, and event notifications on kafka for integration with their back-end systems. Such unfettered ability to control their own timelines, features and costs makes open-source Trellis a huge selling point for some network operators.



Horizontal Scaling & Redundancy. Trellis allows horizontal scaling of the fabric. Operators can start off with simply a single switch or two (for redundancy). As needs grow, they can add more spines and more leaves to their infrastructure seamlessly. This feature supports both scale and redundancy built into both spine and leaf layers as well as the control-plane which is based on a high-performance, scalable and resilient ONOS cluster.

Benefits of P4. Trellis' use of P4 and Stratum, while not yet production ready, offers tantalizing possibilities in the near future. In many ways the holy grail of SDN has been not just the ability to program forwarding rules from a decoupled control plane, but to be able to define and create the forwarding pipeline in the dataplane hardware. P4 enabled switches give this power to SDN platforms like Trellis. As a result, instead of operators depending on vendor proprietary custom ASICs for specialized functionality, they could leverage P4 defined functionalities in bare-metal whitebox hardware used in Trellis, an example of which is described in the next section.



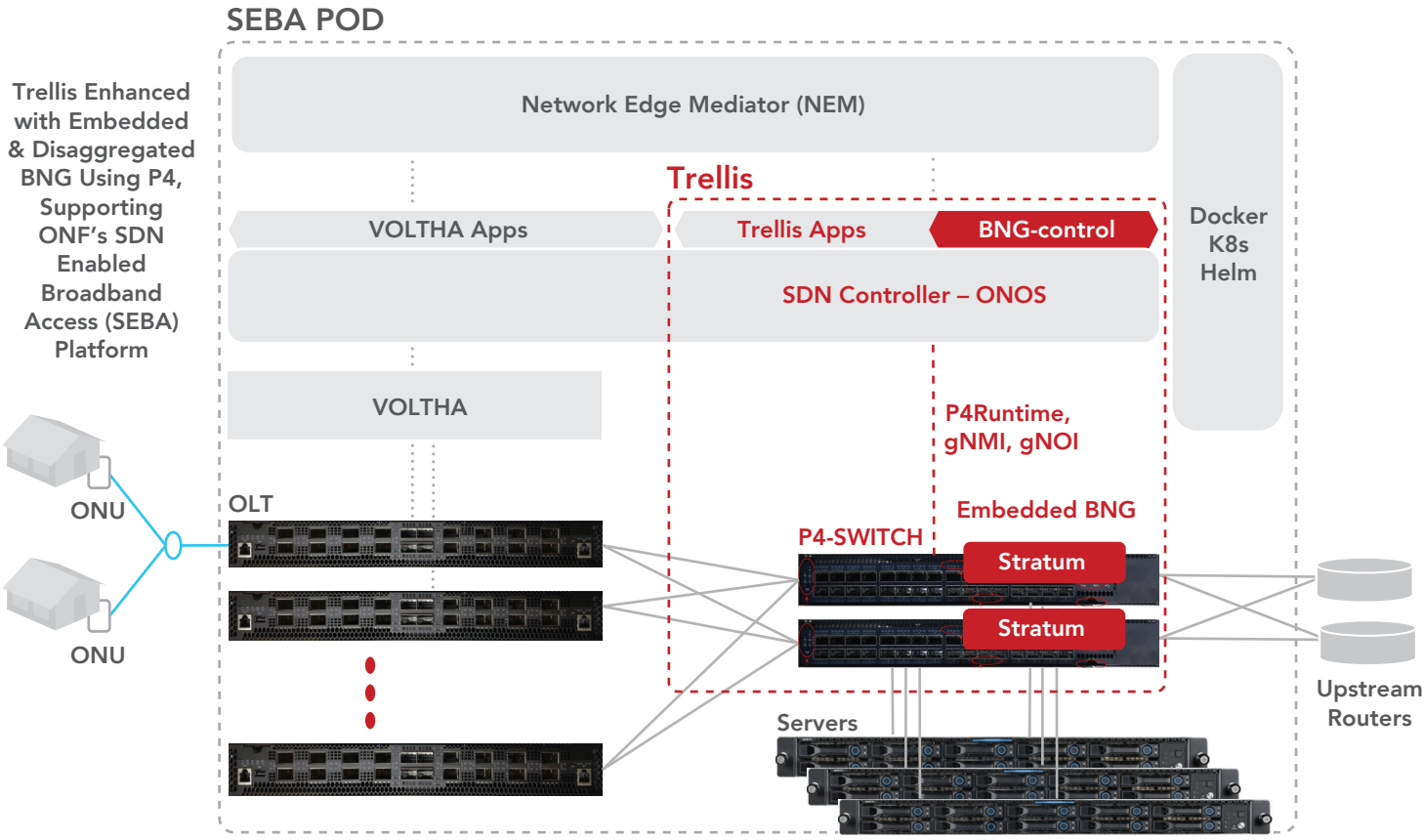
Use Cases

Distributed Fabric for Access/Edge Networking. As detailed in this whitepaper, Trellis is designed for NFV and Access/Edge cloud applications. As such, Trellis is central to the CORD network infrastructure. CORD is an architecture for Telco Central-Offices or Cableco Head-Ends that aims to revolutionize the way they are built and operated. CORD brings together principles of SDN, NFV and Cloud to build cost-effective, common, manageable and agile infrastructure to enable rapid service creation at the network edge. By extending the agility of the cloud to access networks for residential, enterprise and mobile subscribers, CORD aims to enable Edge Computing for next-gen services.

In one instantiation of this idea, Trellis is deployed in production at a Tier-1 US Service Provider's network, in multiple geographies with thousands of live subscribers. In this case, Trellis is deployed as a PPOD (physical pod) with up to 10 switches—2 spines, 4 access-leafs and 2 pairs of server-leafs, that can serve up to 20k subscribers. The 4 access-leafs are single-homed to up to 192 remote access nodes, both optical and RF devices, which further connect to residential and enterprise customers. The server-leafs are dual-homed to compute nodes (servers) that host all the VNFs intended for subscriber traffic tunnel-termination, processing and forwarding. One of the leaf-pairs is also connected to two upstream routers that provide access to the SP's metro/core network.

Trellis apps are responsible for all network connectivity functions. They include authentication of the access nodes via 802.1x and MacSec, DHCP based IP address assignment to subscriber set-top boxes, routing unicast IPv4 and IPv6 traffic, IPTV support via v4 and v6 multicast, advertising subscriber routes to the upstream routers, blackholing routes, DHCPv6 prefix delegation, VLAN cross-connects for enterprise subscribers and more.

As of this writing, Trellis scales in production to roughly 10k subscribers in a single PPOD, with around 50k IPv4 and IPv6 routes resulting in roughly 110k flows in the whole system. In lab settings, nearly double the scale deployed in production has been achieved. We continue to work with the service provider and our ecosystem partners to increase maximum capacity to support even larger production PPOD configurations.



Disaggregated BNG in SEBA Using P4. SDN Enabled Broadband Access (SEBA) is another instantiation of CORD. Indeed, the SEBA exemplar implementation today is built on the CORD software platform, and the project itself is a variant of an earlier version known as R-CORD (or Residential-CORD). SEBA today leverages disaggregation of broadband equipment using SDN techniques, rather than relying on containers or VNFs for processing subscriber data plane traffic. As a result SEBA 'fastpaths' all subscriber traffic from the access node to the external (upstream) backbone straight through hardware, thus achieving much greater scale

In SEBA today, headend equipment in Fiber-to-the-Home (FTTH) networks called OLTs are disaggregated using software from ONF's VOLTHA project. Vendor proprietary chassis-based OLT systems are replaced by simpler white-box OLTs, managed by VOLTHA and a set of applications running on ONOS. In principle, it is similar to how Trellis manages a fabric of whitebox switches with OF-DPA and a set of applications running on ONOS.

In SEBA, dataplane traffic from such white-box OLTs are aggregated upstream by a Trellis leaf switch (or leaf-pair) and VLAN-cross-connected (L2-only) to upstream external Broadband Network Gateways (BNGs). As such, Trellis is used in its simplest form in SEBA.

However, SEBA intends to go further by continuing the disaggregation story by doing the same to the BNG. A vendor proprietary BNG today supports multiple functions like subscriber traffic termination (QinQ, PPPoE), H-QoS, routing, multicast, lawful-intercept, accounting and more. Inspired by prior work from Deutsche Telekom, these functions can also be disaggregated in SEBA in an SDN way by implementing the dataplane features in a P4-enabled switch, and control plane features as applications running on ONOS. In particular, Trellis already supports routing, multicast and QinQ termination functionality. Trellis is then enhanced by additional apps that program a P4-defined pipeline to perform PPPoE tunnel termination and control traffic handling, anti-spoofing, per subscriber accounting, policing, subscriber traffic mirroring and more.

SEBA in its current form, with an external BNG, is being trailed by several operators, and expected to go into production in 2020. SEBA with a disaggregated BNG embedded in Trellis using P4 switches is currently under development by ONF and the SEBA community. We expect this use-case to gain significant momentum next year.

Enterprise Datacenter Fabrics. While Trellis was targeted at Service Provider access & edge applications, there is nothing fundamental in its architecture that precludes it from being used as an Enterprise datacenter fabric. Indeed, Trellis can be used as a pure L3 fabric, or a hybrid L2/L3 Clos fabric without any of the additional features and characteristics meant for carrier access network backends.

Enterprise datacenters, similar to carrier datacenters, typically have some form of virtualization/orchestration system, either open-source (K8s,OpenStack) or commercial (VMWare, RedHat). These VM or container orchestration systems typically have needs from the network expressed as a set of API calls (example OpenStack Neutron or K8s CNI) that are then implemented by plugins from several vendors for their proprietary or open-source *overlay* networking solutions.

When working with overlay networks, Trellis can be treated as an *underlay* fabric. Both underlay and overlay networks/fabrics are independently useful, and can be used independently. However, in certain cases, it is beneficial for underlay and overlay to be aware of each other and work together to deliver the goals of the orchestration-system's network API. Trellis' northbound APIs can be enhanced and integrated with various overlay networking plugins to enable such cooperation. In another instantiation, the overlay network can also be eliminated and Trellis can be the sole provider of the orchestration system's networking needs. To enable the latter, Trellis would need to implement the plugin into the orchestration system's APIs and develop capabilities for network virtualization. We encourage the open-source community to leverage Trellis as a foundation for such explorations.

Chassis Routers. As we described in the Architecture section, Trellis with its vRouter feature behaves just like a traditional chassis based router. Instead of spreading the leaf and spine switches over multiple server-racks in a datacenter, housing the switches and controller-servers in the same rack essentially mimics a chassis-router. The leaf switches are the line-cards, the spine switches are the backplane, and the SDN controller servers are redundant route processor (RP) cards (although RPs are typically active-backup while ONOS instances in each server are all active).

Technically, this approach is similar to what Google did in its B4 work. And if the goal is similar to B4 in its deployment in a private internal network—B4 is a private network connecting Google's datacenters—then Trellis with its current feature set is likely sufficient to replace chassis-based routers. However if the goal is to replace a chassis-router in a public network, Trellis vRouter would need to support more features and protocols like EVPNs or L3VPNs. Depending on the architecture of the public network, there may also be a need to carry the full Internet routing table in addition to VPN routes. Trellis does not currently support hardware that is capable of carrying the global Internet routing table, although techniques exist that propose carrying only a subset of the full table in hardware FIBs, similar to 'working sets' in Operating Systems. We encourage the open-source community to leverage Trellis as a foundation for such explorations.

Specifications

FEATURES	DESCRIPTION
SDN Features	• ONOS cluster of all-active N instances affording N-way redundancy and scale, where N = 3 or N = 5 • Unified operations interface (GUI/REST/CLI) • Centralized configuration – all configuration is done on controller instead of each individual switch • Centralized role-based access control (RBAC) • Automatic host (end-point) discovery – attached hosts, access-devices, appliances (PNFs), routers, etc. based on ARP, DHCP, NDP, etc. • Automatic switch, link and topology discovery and maintenance (keep-alives, failure recovery)

L2 Features	Various L2 connectivity and tunneling support
	• VLAN-based bridging
	  ◦ Access, Trunk and Native VLAN support
	• VLAN cross connect
	  ◦ Forward traffic based on outer VLAN id
	  ◦ Forward traffic based on outer and inner VLAN id (QinQ)
	• Pseudowire
	  ◦ L2 tunneling across the L3 fabric
	  ◦ Support tunneling based on double tagged and single tagged traffic
	  ◦ Support VLAN translation of outer tag

L3 Features	IP connectivity • IPv4 and IPv6 unicast routing (internal use of MPLS Segment Routing) • Subnetting configuration on all non-spine facing leaf ports; no configuration required on any spine port • IPv6 router advertisement • ARP, NDP, IGMP handling • Number of flows in spines greatly simplified by MPLS Segment Routing • Further reduction of per-leaf flows with route optimization logic
DHCP Relay	DHCP L3 relay • DHCPv4 and DHCPv6 • DHCP server either directly attached to fabric leaves, or indirectly connected via upstream router • DHCP client directly either attached to fabric leaves, or indirectly connected via LDRA • Multiple DHCP servers for HA
vRouter	vRouter presents the entire Trellis fabric as a single router (or dual-routers for HA), with disaggregated control/data plane • Uses open-source protocol implementations like Quagga (or FRR) • BGPv4 and BGPv6 • Static routes • Route blackholing • ACLs based on port, L2, L3 and L4 headers
Multicast	Centralized multicast tree computation, programming and management • Support both IPv4 and IPv6 multicast • Dual-homed multicast sinks for HA • Multiple multicast sources for HA
Troubleshooting & Diagnostics	• Troubleshooting tool – T3: Trellis Troubleshooting Tool • Diagnostics one-click collection tool `onos-diags`
Topology	• Single leaf (ToR) or dual-ToR (dual-homing) • Supports typical leaf-spine topology, 2 to 4 spines, up to 10 leaves • Multi-stage leaf-spine fabric (leaf-spine-spine-leaf) • Can start at the smallest scale (single leaf) and grow horizontally

Specifications (continued)

FEATURES	DESCRIPTION
Resiliency	Provides HA in following scenarios • Controller instance failure (requires 3 or 5 node ONOS cluster) • Link failures • Spine failure Further HA support in following failure scenarios with dual-homing enabled • Leaf failure • Upstream router failure • Host NIC failure
Scalability	• (in production) Up to 50k routes, 110k flows, 8 Leaf, 2 Spines, with route optimization enabled • (in pre-production) Up to 120k routes, 250k flows, 8 Leaf, 2 Spines, with route optimization enabled
Security	• TLS-secured connection between controllers and switches (premium feature) • AAA 802.1x authentication • MACSec (L2 encapsulation)

P4-ready	• Support for Stratum, P4Runtime and gNMI and P4 programs
	• Innovative services enabled by programmable pipeline
	  ◦ BNG – PPPoE, anti-spoofing, accounting and more
	  ◦ GTP encap/decap

Overlay Support	Can be used/integrated with 3rd party overlay networks (e.g. OpenStack Neutron, Kubernetes CNI)
Orchestrator Support	Can be integrated with external orchestrator, logging, telemetry and alarm service via REST apis and Kafka events
Controller Server Specs	Recommended (per ONOS instance) • CPU: 32 Cores • RAM: 128GB RAM. 65GB dedicated to ONOS JVM heap (based on 50K routes)

Whitebox Switch Hardware	• Multi-vendor: Edgecore, QCT, Delta, Inventec
	• Multi-chipset
	  ◦ Broadcom Tomahawk, Trident2, Qumran
	  ◦ Barefoot Tofino
	• 1/10G, 25G, 40G to 100G
	• Refer to docs.trellisfabric.org/supported-hardware.html for the most up-to-date hardware list

Whitebox Switch Software	• Open source ONL, ONIE and Indigo OF client • (in production) OF-DPA software commercial version – contact Broadcom • (in labs/trials) OF-DPA software community version available from ONF (for switch models based on Trident and Tomahawk, not Qumran) • (in labs/trails) Stratum available from ONF
Documentation	docs.trellisfabric.org

ONF is an operator-led consortium transforming networks into agile platforms for service delivery.

For more technical information and tutorials: opennetworking.org/trellis

To learn of the Trellis commercial ecosystem: info@opennetworking.org