ODTN, Trellis and Stratum: A Seamless Packet-Optical Multi-Stage Datacenter Solution

Andrea Campanella, Brian O'Connor, Carmelo Cascone, Charles Chan, Pier Luigi Ventre,
Maximilian Pudelko, and Yi Tseng.
Open Networking Foundation, Menlo Park, CA, U.S.

(Invited Paper)

***Abstract*—With the dramatic increase in data traffic envisioned for the next years, operator's network will be requested to sustain a large amount of data traffic, while at the same time being more flexible easily upgradable and having a lower total cost ownership (TCO). The industry at large is shifting towards SDN architectures, with white-box devices and open source software as building blocks. Such trend allows for further optimization across network layers, packet and optical as an example, with shared infrastructure and control. In this paper, we demonstrate a full end-to-end multi-stage data-center solution, combining leaf-spine control from Trellis, dedicated optical link provisioning by ODTN, and switches and packet optical programmable devices managed by Stratum.**

Unifying optical and Ethernet capabilities, the Cassini packet-optical transponder acts as a spine in the intra data-center Trellis leaf spine topology, and as the transponder for the DCI inter data-center links.

We will describe the implemented architecture, the overall topology, the changes required to achieve the end to end solution and the challenges faced. We then outline the benefits of the proposed solution in terms of resource consumption, end to end optimisation, reliability and resiliency based upon data collected in a physical setup at ONF. Finally, we conclude this paper with future work and proposed optimisations.

***Index Terms*—Disaggregated Optical Networks, Software Defined Networking (SDN), Control Plane Resilience, Open and Disaggregated Transport Networks (ODTN), Leaf-Spine Fabric, Data Plane Programmability, Open Networking Foundation (ONF), Open Network Operating System (ONOS).**

I. Introduction

KEY requirements of service provider networks include programmability, resiliency, low cost of deployment and low total cost of ownership (TCO). A flexible, re-configurable and elastic architecture is also critical to fulfill rapid changing requirements and use cases.

Open Networking Foundation (ONF) has developed several software platforms in order to meet these requirements. A key one is the Open Network Operating System (ONOS) [7] [21], a multi-domain SDN controller capable of handling different network deployments, devices and use-cases. In addition to its deployment flexibility, ONOS provides a resilient highly available control plane, through the use of a cluster of instances and distributed stores [3]. Moreover, ONOS is capable of handling service provider's scale requirement, supporting thousands of device, hundreds of thousands of flow rules and groups [1].

978-3-903176-33-1 © 2021 IFIP

A number of applications have been developed on top of ONOS to support various scenarios, including Trellis for data-center leaf-spine fabric control [1] and ODTN for optical Data Center Interconnect (DCI) [2] [4] [3]. ONF also collaborates with Google to develop Stratum [19], an open-source, ASIC-independent thin switch operating system, to provide SDN capabilities on whitebox devices.

The solution described in this paper integrates all of these development efforts into an end-to-end solution for data-center networking, with Stratum deployed across a variety of devices in a leaf-spine topology, including packet optical transponders. Discovery, management and connectivity provision both within and across data-centers are performed by Trellis and ODTN collectively across multiple network layers from L0 to L3.

The paper describes the complete end to end solution and is organized as follows. Firstly, Sec. II introduces Trellis, the open-source multi-purpose leaf-spine fabric. Then, Sec. III details ODTN which adds optical capabilities to ONOS. Sec. IV presents Stratum. Sec. V details the Cassini device. The integration of Trellis, ODTN and Stratum is then analyzed in section VI. VII describes a real topology used for our demonstration. Benefits of the described solution are presented in VIII Finally, Sec. IX draws conclusions and next steps.

II. Trellis

Trellis is an open-source multi-purpose L2/L3 leaf-spine fabric solution. As a classic SDN implementation, Trellis leverages an SDN Controller (ONOS) to program the ASIC forwarding tables. There are a set of ONOS applications being implemented to support fabric functionality and features. The main application, fabric-control, handles L2 forwarding (bridging) within a server-rack, and L3 forwarding (routing) across racks. Trellis internally uses MPLS Segment Routing (MPLS-SR). To route an IP packet across the fabric, the source leaf will first push an MPLS label to IP packets to specify the destination leaf. It then hashes the MPLS packet to the spines using ECMP in order to achieve load balancing and failure recovery. The spines forward the packet solely based on the MPLS label. This design significantly reduced the number of flows required on the spines.

Importantly, Trellis leverages ONOS Flow Objective API [5], which allows applications to program network devices

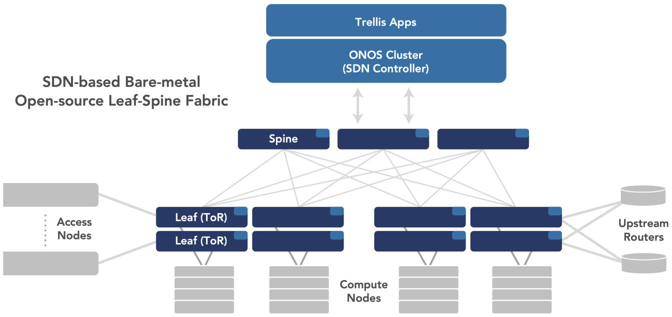


Fig. 1. Trellis Architecture and typical deployment

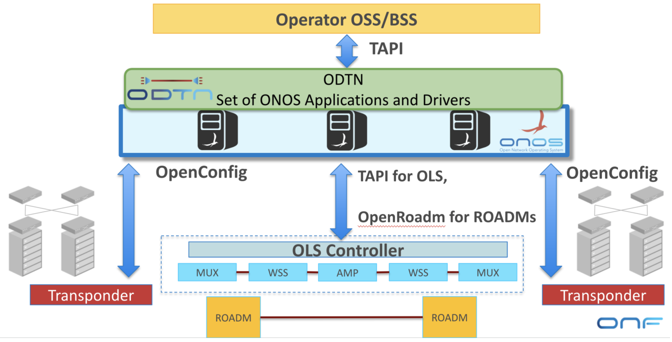


Fig. 2. ODTN architecture

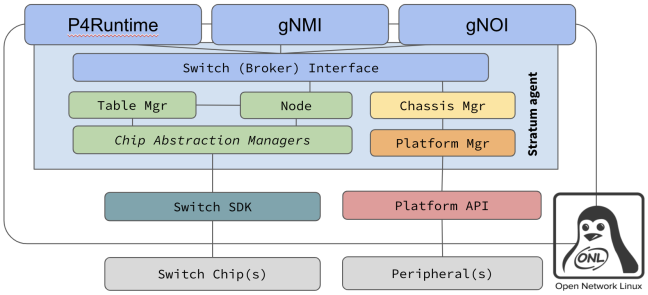


Fig. 3. Stratum overview and internal architecture

in an ASIC pipeline-agnostic way. The pipeline-agnostic API calls are then translated into device-specific entries by corresponding device drivers. Using Flow Objective APIs, the same application can be adapted to various hardware pipelines and support different control protocols such as OpenFlow and P4 Runtime. In 2020, Trellis was deployed in production in the U.S., serving 160k+ subscribers, 420k+ connected devices, 25 pods and 14 geographies [6].

III. ODTN

The Open Disaggregated Transport Network (ODTN) is a set of applications on top of ONOS that provides optical specific APIs and models to both Northbound interface (NBI) and Southbound Interface (SBI). Specifically, the NBI interface is already well-defined adopting T-API (version 2.0) [16]. ODTN manages optical network devices, such as transponders, amplifiers Open Line systems (OLS) and ROADMs (re-configurable add-drop multiplexer) providing the capability to setup optical connectivity between two endpoints (transponders) of an optical line by configuration of wavelengths, cross-connects, power and modulation.

The optical connectivity is created by requesting a high level configuration of the optical layer called intent [9]. ODTN applications perform path computation and spectrum assignment, creating a pair of flow rules for each device in the optical path. Rules are then forwarded to the device specific drivers [18] in the SBI. Similar to the above-mentioned Trellis case, the driver translates the flow rules into model specific messages such as OpenConfig [13], OpenROADM [14] or T-API [16]. The model-specific messages then get forwarded to the device through certain protocols, such as NETCONF, RESTCONF, REST or gNMI, in order to apply the required configurations. Figure 2 shows the ODTN architecture and supported deployment solutions.

IV. Stratum

Stratum [10] is a next generation, thin, ASIC independent network OS for both traditional fixed-pipeline and programmable switching devices. Stratum leverages the following interfaces to achieve this:

- P4 programming language [11] is used to define the forwarding behavior for programmable switching chips as well as model fixed-pipeline ASICs, establishing a contract implemented by the data plane and programmable by the control plane.
- P4Runtime [20] is an SDN control plane interface for controlling forwarding behavior at runtime. It populates forwarding tables and manipulates packet processing behaviour based on a P4 program, in a hardware agnostic way.
- OpenConfig models defines a vendor-agnostic base for device configuration and management.
- gNMI (gRPC Network Management Interface) improves on the existing configuration interfaces by using a binary representation on the wire (protocol buffers - Protobuf), and enabling bi-directional streaming.

V. Cassini

The Edgecore AS7716-24SC (Cassini) [12] is an open and disaggregated packet-optical transponder with modular optical interface design, covering data center interconnect, metro and access backhaul use cases. It offers system throughput of 3.2 Tbps based on Broadcom StrataXGS™ Tomahawk™ Plus switching silicon with 16 fixed 100 Gigabit Ethernet QSFP28 ports and 8 line card slots to incorporate a flexible mix of (a) 100/200 Gbps Analog Coherent Optics (ACO) and Digital Coherent Optics (DCO) Digital Signal Processor (DSP) from Acacia Communications, Fujitsu Optical Components and Lumentum and/or (b) additional 100GbE ports.

VI. Integrating Trellis, ODTN and Stratum

To achieve production grade, P4 programmable, SDN fabric integration efforts had to be made first to extend Stratum with

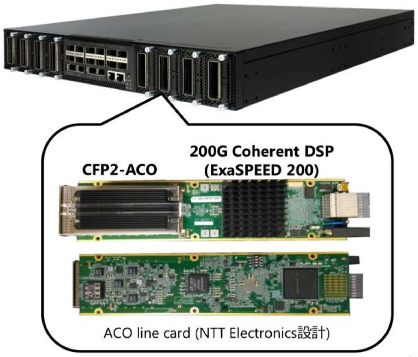


Fig. 4. Cassini Edgecore AS7716-24SC optical transponder equipped with Lumentum CFP2-ACO transceivers.

optical capabilities, then to tie together Trellis, ODTN over Stratum controlled devices.

To support the configuration of optical parameters over ACO/DCO cards both the Stratum northbound and southbound interfaces had to undergo enhancements. In particular, on the northbound the wavelength, power and terminal endpoints of the optical OpenConfig models are now supported and exposed through gNMI. Stratum's platform manager has been extended to include the TAI [15] libraries towards the physical device in order to properly communicate with Optical pluggable cards, both ACO and DCO. TAI provides a vendor-independent mechanism to control optical components and simplifies the integration work between a network operating system and the underlying optical hardware. Through the use of TAI Stratum can now configure both ACO and DCO modules supported by TAI. By linking the OpenConfig models to the TAI API calls, Stratum relays optical configuration from a controller down to the physical device, establishing the optical channel. Stratum also had to be extended to support the Cassini chassis platform and the Tomahawk+ chipset from Broadcom.

On the ONOS side, ODTN drivers have been extended with a new driver that uses OpenConfig models over the gNMI protocol to configure Stratum based Cassini devices. The new driver is capable of configuring and retrieving wavelengths, power and the newly exposed optical ports.

From a Trellis perspective, integrating Cassinis' packet side through Stratum was seamless because extensions were already made to include P4Runtime protocol. Fabric.p4 was already supported by Stratum and by the Broadcom chipset APIs, thus all L2/L3 and resiliency capabilities of Trellis were readily available through the ONOS drivers, namely the FabricPipeliner. To achieve the end-to-end solution, Trellis, ODTN and their respective drivers have been deployed all together on top of ONOS, making sure no conflict was present.

Through this integration the Cassini device slotted in as a spine in a typical multi-stage Trellis deployment, acting both as the spine for intra data-center communication and as the optical transponder for inter data-center dedicated optical link.

The integration is possible thanks to proper level of abstractions in both ONOS and Stratum. As an example, Trellis and ODTN being written on top of Intent and Flow-Objectives APIs in ONOS allowed to integrate the Cassini device through just a set of drivers, with no change to the applications.

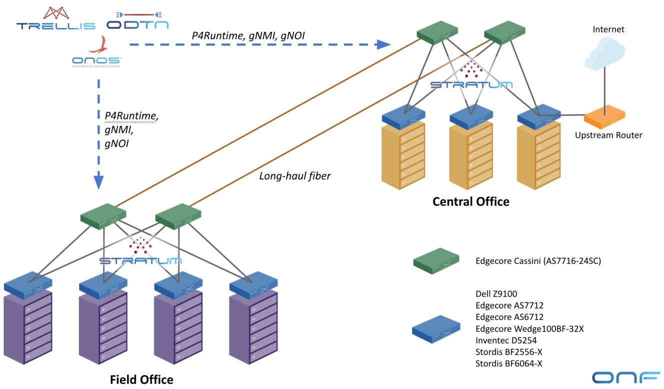


Fig. 5. Deployment Topology

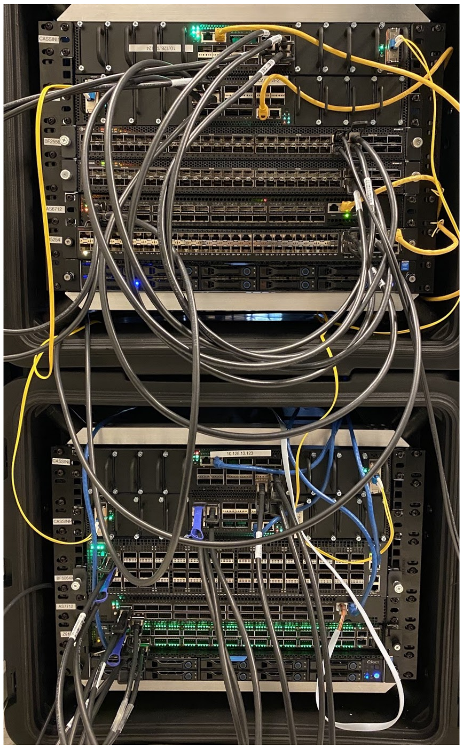


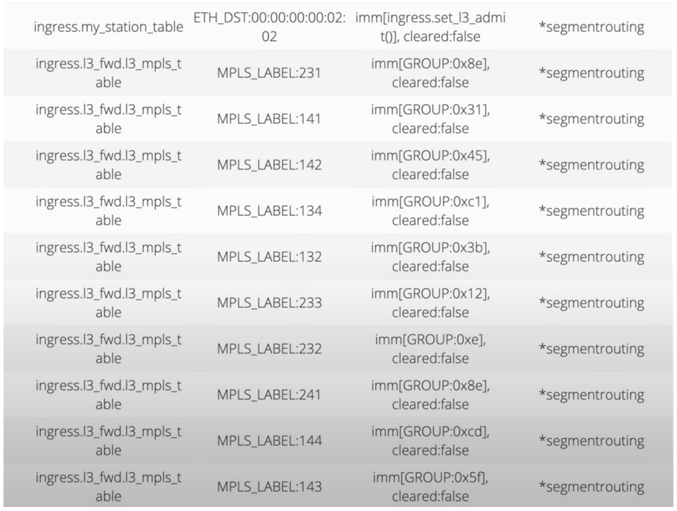
Fig. 6. Deployment Rack

The integration has, although, presented its challenges, especially on the Stratum side for the integration of both TAI and the Tomahawk+ chip, requiring adjustments and extensions in the Stratum components to cater for the small but significant changes in optical ports.

Figure 5 showcases in detail a deployment of Trellis and ODTN over a combination of Cassini devices as spines and transponders and different switches for leafs.

VII. Demonstration

The integrated solution has been deployed and demonstrated on a physical setup in ONF's Menlo park office. Figure 6 visualizes two racks, the top one acting as the field office, while the bottom one being the central office. To provision an end-to-end path across data centers, Trellis will program the



ingress.my_station_table	ETH_DST:00:00:00:00:02:02	imm[ingress.set_l3_admit()], cleared:false	*segmentrouting
ingress.l3_fwd.l3_mpls_table	MPLS_LABEL:231	imm[GROUP:0x8e], cleared:false	*segmentrouting
ingress.l3_fwd.l3_mpls_table	MPLS_LABEL:141	imm[GROUP:0x31], cleared:false	*segmentrouting
ingress.l3_fwd.l3_mpls_table	MPLS_LABEL:142	imm[GROUP:0x45], cleared:false	*segmentrouting
ingress.l3_fwd.l3_mpls_table	MPLS_LABEL:134	imm[GROUP:0xc1], cleared:false	*segmentrouting
ingress.l3_fwd.l3_mpls_table	MPLS_LABEL:132	imm[GROUP:0x3b], cleared:false	*segmentrouting
ingress.l3_fwd.l3_mpls_table	MPLS_LABEL:233	imm[GROUP:0x12], cleared:false	*segmentrouting
ingress.l3_fwd.l3_mpls_table	MPLS_LABEL:232	imm[GROUP:0xe], cleared:false	*segmentrouting
ingress.l3_fwd.l3_mpls_table	MPLS_LABEL:241	imm[GROUP:0x8e], cleared:false	*segmentrouting
ingress.l3_fwd.l3_mpls_table	MPLS_LABEL:144	imm[GROUP:0xcd], cleared:false	*segmentrouting
ingress.l3_fwd.l3_mpls_table	MPLS_LABEL:143	imm[GROUP:0x5f], cleared:false	*segmentrouting

Fig. 7. Trellis flows

paths between leaf switches and the packet side of Cassini devices.ODTN then programs the optical links between Cassini devices across the data centers.

We also provisions redundant paths thanks to dual homed hosts, leafs and spines, with resiliency mechanisms built in both Trellis [1] and ODTN [3]. Following is an example of wavelength configuration with selected OchSignal -35×50 GHz, which is translated into the 191.35 THz wavelength.

```
18:15:25.518 INFO [ClientLineTerminalDeviceFlowRuleProgrammable] OpenConfig added
        flowrule TerminalDeviceFlowRule{id=bf000012c33c7d, deviceId=general
        :10.100.100.3:830, priority=100, selector=[IN_PORT:10101, OCH_SIGID:OchSignal
        {-35 x 50.00GHz +/- 25.00GHz}, OCH_SIGTYPE:FIXED_GRID], treatment=
        DefaultTrafficTreatment{immediate=[OUTPUT:10101], deferred=[], transition=
        None, meter=[], cleared=false, StatTrigger=null, metadata=null}, tableId=0,
        created=1572967515459}
18:15:25.557 INFO [ClientLineTerminalDeviceFlowRuleProgrammable] Optical Channel
        Frequency <components xmlns=thttp://openconfig.net/yang/platform’><component
        nc:operation=’merge’><name>oc1/0</name><oc-opt-term:optical-channel <xmlns:oc
        -opt-term=’http://openconfig.net/yang/terminal-device’></oc-opt-term:config><
        oc-opt-term:frequency> 191650000 </oc-opt-term:frequency> </oc-opt-term:
        config> </oc-opt-term:optical-channel></component></components>
```

The following trace shows ONOS configuring a power at -6 dBs on the line port of the Cassini through Openconfig.

```
18:28:18,650 INFO [CassiniTerminalDevicePowerConfig] 246 - org.onosproject.onos-
        drivers-odtn-driver - 2.3.0.SNAPSHOT I Setting per <rpc xmlns--urn:ietf:
        params:xml:ns:netconf:base:1.0"><edit-confip<target><candidate/></target><
        config><components xmlns-"http://openconfig.net/yang/platform"><component><
        none>oc1/0</name><optical-channel xmlns-"http://openconfig.net/yang/terminal-
        device"><config><target-output-power>-6<target-output-power></config></
        optical-channel></component></components></config></ edit-config></rpc>
```

Figure 7 shows some of the Trellis configured flow rules, in particular MPLS label related forwarding rules.

A live version of this demonstration has been done at the OCP 2020 global summit.

VIII. Benefits of a combined solution

The combined and integrated solution described in VI and VII inherits all the leaf-spine fabric capabilities that Trellis possesses including redundancy, fail-over and scalability. It also encompasses the optical capabilities implemented in the ODTN apps and drivers. Finally it leverages all the stability, programmability and performance of Stratum.

As shown in the demo deployment and configuration, the same automatic provisioning of end to end paths is achieved, thus no change is seen by the operator and user of the network.

The operator can benefit from such integrated deployment in multiple ways. Trellis allows horizontal scaling of the fabric. Operators can start off with simply a single switch and as needs grow, they can add more spines and more leaves seamlessly. Operators also have unparalleled control over their network, the features it includes, various customization and optimisations of those features, and how they integrate with the operator’s back-end systems, giving them the ability to control their own timelines, features and costs.

The APIs provided by ONOS also allow to seamlessly enhance the network as new technologies come in by swapping in new devices and drivers while re-using the hardened version of the applications.

Having the whole deployment manged by one entity also allows for great end to end optimisation of configuration and bandwidth allocation, not to mention having a single point of inquiry for metrics, statistics and alarms, greatly simplifying FCAPS processes and troubleshooting. Great optimisation comes also in terms of cost. The use of white-box (bare-metal) switching hardware from ODMs significantly reduces CapEx 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. Furthermore with the introduction of the Cassini as a spine in the fabric a device gets removed from the network entirely, with cost saving in terms of equipment, power and AC consumption and simplification of configuration and maintenance of the deployment.

IX. Conclusion

This demonstration has shown the feasibility of the first full blown open source leaf-spine fabric solution, including data center interconnect capability through the use of the ONOS SDN controller to provision data connectivity services across the disaggregated topology with real hardware exposing common and open data models through the use of Stratum. Such deployment provides a seamless end to end path provisioning for hosts connected to the network, while ensuring resiliency and fail over in case of errors. Proving to be a cost effective, scalable and deployable solution for operators. The demo showcases the first open source optical device stack with switch OS, Stratum, open line card APIs TAI and deployed over the Cassini white box transponder.

Overall, this work showed the feasibility of the selected approach while further work is required to enhance the system with more optical capabilities and more device support.

Acknowledgment

This work has been done under the ONF framework with an open source community and the source code is available under Apache 2.0 license.

References

[1] *Open Networking Foundation (ONF), Trellis platfrom brief* https://opennetworking.org/wp-content/uploads/2019/09/TrellisPlatformBrief.pdf Accessed: 12 May 2021.

[2] A. Campanella, Boyuan Yan, R. Casellas, A. Giorgetti, V.r Lopez, and A. Mayoral, *Reliable Optical Networks With ODTN, Resiliency and Failover In Data Plane And Control Plane",* in Proc. European Conference on Optical Communication (ECOC) - Demo Session, Dublin, Sept. 2019.

[3] A. Campanella, Boyuan Yan, R. Casellas, A. Giorgetti, V.r Lopez, Yongli Zhao and and A. Mayoral, *"Reliable Optical Networks With ODTN: Resiliency and Fail-Over in Data and Control Planes"* in J. Lightwave Technol., vol. 38, no. 10, pp 2755–2764 (2020).

[4] A. Giorgetti, A. Sgambelluri, R. Casellas, R. Morro, A. Campanella, and P. Castoldi, *Control of Open and Disaggregated Transport Networks using Open Network Operating System (ONOS),* J. Opt. Commun. Netw., vol. 12, no. 2, pp. A171-A181 (2020).

[5] *ONOS Flow Objectives (ONF), SDN A systems approach (2020)* https://sdn.systemsapproach.org/onos.html#northbound-interface Accessed: 13 May. 2021.

[6] *Trellis in production with COMCAST.* https://www.prnewswire.com/news-releases/comcast-has-achieved-production-roll-out-of-trellis-open-source-networking-fabric-300917819.html Accessed: 13 May. 2021.

[7] *Open Network Operating System*, https://github.com/opennetworkinglab/onos accessed 27 Nov. 2019.

[8] R. Enns, ed. *Network Configuration Protocol NETCONF*, IETF Request for Comments 6241, June 2011.

[9] A. Campanella, *Intent Based Network Operations*, in Tech. Dig. Optical Fiber Communication Conference (OFC), San Diego, 2019.

[10] B. O'Connor, Y. Tseng, M. Pudelko, C. Cascone, A. Endurthi, Y. Wang, A. Ghaffarkhah, D. Gopalpur, T. Everman, T. Madejski, J. Wanderer, A. Vahdat *Using P4 on Fixed-Pipeline and Programmable Stratum Switches* ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS) (2019).

[11] P. Bosshart, D. Daly, G. Gibb, M. Izzard, N. McKeown, J. Rexford, C. Schlesinger, D. Talayco, A. Vahdat, G. Varghese, D. Walker *P4: Programming Protocol-Independent Packet Processors*, Association for Computing Machinery, vol. 44, no. 3, pp. 87-95 (2014).

[12] *Edgecore Cassini* https://www.edge-core.com/solution-inquiry.php?cls=5&id=58, Accessed: 12 Apr. 2021.

[13] *OpenConfig Project*, http://www.openconfig.net, Accessed: 13 May. 2021.

[14] *OpenROADM Project*, http://openroadm.org, Accessed: 13 May. 2021.

[15] *Disaggregated Transponder Chip Transport Abstraction Interface* https://github.com/Telecominfraproject/oopt-tai, Accessed: 13 May. 2021.

[16] *Open Networking Foundation (ONF), Transport API project* https://wiki.opennetworking.org/display/OTCC/TAPI Accessed: 12 May. 2021.

[17] *RFC8040: 'RESTCONF Protocol'*, 2017.

[18] *ODTN Drivers* https://github.com/opennetworkinglab/onos/tree/master/drivers/odtn-driver/src/main/java/org/onosproject/drivers/odtn.

[19] *Stratum white box operating system* https://www.opennetworking.org/stratum/.

[20] *P4Runtime* https://p4.org/p4runtime/spec/main/P4Runtime-Spec.html Accessed: 12 May 2021.

[21] P. Berde, M. Gerola, J. Hart, Y. Higuchi, M. Kobayashi, T. Koide, B. Lantz, B. O'Connor, P. Radoslavov, W. Snow and others, *ONOS: towards an open, distributed SDN OS*, in Proceedings of the third workshop on Hot topics in software defined networking (HotSDN), 2014.