TeraFlow: Secured Autonomic Traffic Management for a Tera of SDN flows

Ricard Vilalta[*], Raul Muñoz[*], Ramon Casellas[*], Ricardo Martínez[*], Victor López[†], Oscar González de Dios[†],
Antonio Pastor[†], Georgios P. Katsikas[‡], Felix Klaedtke[§], Paolo Monti[¶], Alberto Mozo[||], Thomas Zinner[**],
Harald Øverby[**], Sergio Gonzalez-Diaz[††], Håkon Lønsethagen[‡‡], José-Miguel Pulido[x], Daniel King[xi]

[*]Centre Tecnològic de Telecomunicacions de Catalunya (CTTC/CERCA), Castelldefels, Spain
[†]Telefónica I+D, Madrid, Spain
[‡]UBITECH, Athens, Greece
[§]NEC Labs Europe, Heidelberg, Germany
[¶]Chalmers University of Technology, Göteborg, Sweden
[||]Universidad Politécnica de Madrid, Madrid, Spain
[**]Norwegian University of Science and Technology, Norway
[††]Atos, Madrid, Spain
[‡‡]Telenor, Fornebu, Norway
[x]Volta Networks, Barcelona, Spain
[xi]Old Dog Consulting, Llangollen, UK

***Abstract*—TeraFlow proposes a new type of secure, cloud-native Software Defined Networking (SDN) controller that will radically advance the state-of-the-art in beyond 5G networks by introducing novel micro-services architecture, and provide revolutionary features for both flow management (service layer) and optical/microwave network equipment integration (infrastructure layer) by adapting new data models. TeraFlow will also incorporate security using Machine Learning (ML) and forensic evidence for multi-tenancy based on Distributed Ledgers. Finally, this new SDN controller shall be able to integrate with the current Network Function Virtualization (NFV) and Multi-access Edge Computing (MEC) frameworks as well as to other networks. The target pool of TeraFlow stakeholders expands beyond the traditional telecom operators towards edge and hyperscale cloud providers.**

I. Introduction

Software Defined Networks (SDN) have been in the market for more than 10 years with great technological success, allowing network softwarization and becoming a common tool. Since the introduction of the OpenFlow protocol in 2008 [1], new and more advanced generations of SDN protocols have emerged and have been introduced into transport networks, cross-haul networks, Data Centre (DC) or campus networks. Despite this optimistic market picture, the crude reality is that network operators are only slowly adopting basic SDN deployments. Facing this reality, it can be claimed that there is no clear path to introduce SDN in operator networks.

The need for network automation clear and will allow network operators to fully benefit from SDN adoption. Enabling such automation will facilitate the use cases and workflows across standards-based interfaces that can operate in greenfield and brownfield deployments. During the last few years, we observed the confluence of Artificial Intelligence/Machine Learning (ML) and 5G networks, with ML adoption is slowly evolving, but without a clear adoption path. Automation and ML are two flips of the same coin, and the applicability of automated decisions in this context is uncertain when dealing with challenges in network management, security, optimization, and scalability.

As 5G networks are deployed, 3GPP release 17 and upcoming Beyond 5G (B5G) networks will require massive scale flow management. Automation is essential in these scenarios, in which humans will not be able to manage and operate these networks. SDN has to provide the capabilities to fulfil these requirements. For example, flow aggregation at the network core is not efficient enough. A massive number of flows, realized as network intents, cannot be consistently handled by current SDN controller solutions, such as ONOS or OpenDayLight [2]. These solutions consist of a monolithic software core, that can synchronize with other deployed SDN controllers through specific protocols. Some limitations to this current software architecture have been raised, and SDN organizations are slowly looking at possible solutions, by completely redesigning SDN controllers. For example, μONOS promises to provide a cloud-native SDN controller.

Cloud-native architectures consist of stateless microservices which interact with each other to fulfil network management tasks. But, only considering a microservice-based software architecture (even at the edge) is not enough to achieve this goal, as there is also a clear need for hardware-specific offloading in support for B5G scenarios. This can be done with the introduction of P4/OpenFlow-based programmable switches, as well as for example (FPGA-based) Smart Network Interface Cards or Graphics Processing Units.

In this context, network automation requires the introduction of new software components that are able to detect and eliminate security attacks in a timely manner. This leads to a novel SDN controller re-design with a security-centric design to reduce exposure to attacks and enhance the diagnostic potential of the network. In order to achieve such network automation, ML components integrated by design in SDN controllers (e.g., ML applications running on top of the controller) has emerged as an encouraging approach. Moreover, Permissioned Distributed Ledgers (PDL) are expected to bring novel use cases to evolve security in B5G networks, such as smart-contracts, to enforce resource allocation or real time weaknesses analysis of network applications.

This also leads to the need for proper integration of SDN controllers in Network Functions Virtualization (NFV) and Mobile Edge Computing (MEC) orchestration paradigms to be applied to B5G networks. Finally, an operator domain should supports smart or value added connectivity services on-demand as well as assured service quality (ASQ) path services at various traffic aggregate levels [3].

This paper proposes a complete architecture in order to provide a cloud-scale number of SDN flows, while preserving security and autonomicity of network traffic management. In the first section, we present a state of the art (SoA) review on SDN controllers, security in SDN networks and emerging smart connectivity services. Later, we present the proposed TeraFlow architecture. Then, we focus on life-cycle automation and high performance SDN components. Later, we discuss some network security and interworking across B5G networks. Then, we compare the proposed solution against SoA SDN controllers and finally we provide a few concluding remarks.

II. State of the Art

In this section, we review the SoA of SDN controllers, as well as related security and integration support for beyond 5G networks.

A. SDN Controllers

The increasing interest of the community in the SDN paradigm lead to the development of numerous open-source SDN controllers [4], having the maximum exponents in Open Networking Operating System (ONOS) [5] and OpenDayLight (ODL) [6].

Open Network Operating System (ONOS) is an open source SDN network operating system originally created by ON.lab in 2014 [7]; it is currently supported by the Open Networking Foundation (ONF) and freely distributed under the Apache 2.0 license. ONOS is architected as a distributed system, employing clustering mechanisms to provide a scalable and robust SDN control plane solution. The ONOS components and applications are written in Java as bundles, loaded into Apache Karaf, powered by OSGi. Apache Karaf is an application container that enables the development of ONOS modules that can be installed and run dynamically in the form of OSGi bundles in a single Java Virtual Machine. ONOS supports a wide variety of SouthBound Interface (SBI) protocols, including OpenFlow, NETCONF, RESTCONF, BGP, MPLS, OSPF, OVSBD or P4Runtime among others, also supporting telemetry using Influx database and Grafana plugins. ONOS also provides Northbound Application Programming Interfaces (API) that simplify the administration with REST API systems and extensible user interfaces.

OpenDaylight (ODL) is a modular open-source SDN platform that was born as a collaborative project in 2014 [8]; it is currently hosted by the Linux Foundation and distributed under the Eclipse Public License. The ODL platform core is a Model-Driven Service Abstraction Layer (MD-SAL), a message-bus extensible middleware component that represents the underlying devices as objects or models and provides messaging and data storage functionality using YANG as modeling language. In ODL SBI protocols and control plane services, anchored by the MD-SAL, can be individually selected and developed, supporting a wide variety of state of the art protocols such as OpenFlow, OVSDB, NETCONF or BGP among others. All the ODL components are developed in Java, packaged and loaded using Apache Karaf, offering a console interface and a graphical interface through the OpenDaylight User Experience application for the network management.

micro-ONOS. The next generation SDN architecture, titled micro-ONOS or simply μONOS [9], transforms the monolithic ONOS architecture into a highly-distributed ecosystem of cloud native microservices. The main objective of μONOS is to drive the transformation of network infrastructures through an industrial-grade platform that addresses key requirements of modern networked systems, such as network control, configuration, packet-level network telemetry, run-time verification, diagnostics, and a first-class support for 5G functions at the network edge. Among the key differences between ONOS and its successor μONOS, the most prominent ones are noted: (i) support for next generation SDN interfaces, such as gNMI and gNOI as well as native P4Runtime; and (ii) zero-touch provisioning using established tool-chains, such as Kubernetes, helm charts, ansible, etc. This allows μONOS along with auxiliary applications to be seamlessly orchestrated as parts of a larger solution, in a modular and flexible manner.

OpenDayLight Micro is the first project inside OpenDayLight to explore the use of microservices for the development of a new generation of SDN controller with the main goals of simplifying the ODL deployment; reducing the time it takes to develop and debug ODL features; reducing the runtime memory footprint; and reducing the start up time [10].

B. Security in SDN

Many different security issues have been identified in SDN architectures [4], including, among others, unauthorized accesses and disclosure of information, modification or destruction of network information, service disruption and misconfigurations. It is worth noting that all SDN layers and interfaces might become the aim of different types of attack vectors: i) Application layer (e.g. application termination, service neutralization, attacks to northbound API), ii) Control layer (e.g. dynamic flow rule tunneling, controller poisoning, network operating system misuse, forced switch disconnection, packet-in and controller's switch table flooding), iii) Control channel (e.g. eavesdropping and man in the middle), and iv) Infrastructure layer (e.g. denial of service, flow-rule modification and flooding, malformed control packet injection and side-channel attacks).

It is only recently that ML techniques have been applied to detect threats and attacks in SDN architecture. As an isolated ML component or being part of a toolbox or an

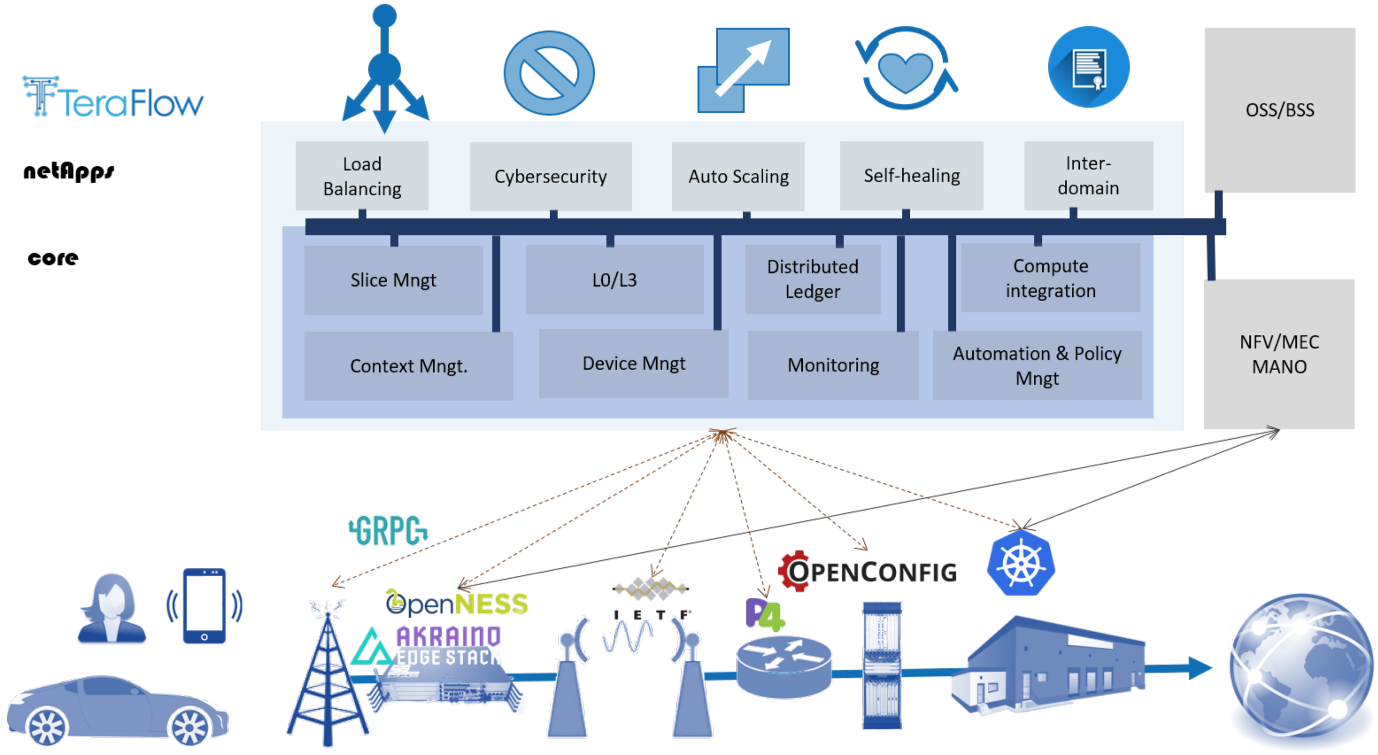

Fig. 1: TeraFlow architecture

Intrusion Detection System, ML components are deployed as applications in the SDN controller which significantly increases the controller's attack surface. In this context, ML components, and in particular deep neural networks, have been found vulnerable against malicious and well-designed examples than can easily fool a ML model with little perturbations imperceptible to humans. Some recent works describe the so-called adversarial attacks in the context of Cybersecurity [11] and more specifically in SDN networks [12]. Although these sophisticated adversarial attacks are in their infancy, the main conclusion is that testing in training processes is insufficient because it provides a lower bound on the failure rate of the system, and therefore in order to provide security guarantees, an upper bound is necessary.

The key features of blockchains, namely, decentralization, immutability, and transparency make the use of blockchains also appealing for managing resources and services in multi-tenant networks. The use of blockchains replaces centralized network management with conventional database management systems. Currently no SDN controller has proposed this solution for multi-domain scenarios.

C. Integration support of SDN controllers

B5G network comprise a heterogeneous set of network and service providers including traditional telecom providers, edge providers and hyperscale cloud providers, utilizing different cloud, network and service management paradigms. Accordingly, a SDN controller needs to be integrated to work with the respective OSS/BSS [13] and MANO systems for NFV and MEC [14].

Besides this integration with the internal management, an improved integration in the interdomain eco-system is needed to enable managed and ASQ concepts beyond single operator domains. A central concept considering these interconnection services, as formulated by the 5GEX project [3], is the so-called Point of Interconnection (PoI) to Region (PoI2Reg). It allows network service providers (NSPs) to hide topology details while allowing a simplified view on the ASQ path infrastructure into a region where the traffic is terminated or delivered to a range or a region of end-points. On top of this abstracted topology of managed or ASQ path infrastructure paths the NSP is able to receive requests for and deliver smart and value added connectivity session services in an on-demand fashion. Thus, the needs of specific applications in edge or core data-centers or in enterprise premises can be matched. The connectivity service concept relies on separating the abstracted infrastructure layer (the traffic / forwarding aggregates) and the layer supporting the on-demand connectivity. A recursive and hierarchical composition of this separated instances further contributes to operational simplicity and scalability of the outlined approach. These state-of-the-art connectivity service concepts need to be evolved in the B5G network context for a variety of 5G services and use cases. This includes an analysis of corresponding business concepts paying attention to both the innovation potential they can unleash as well as regulatory aspects and net neutrality [15].

III. Proposed TeraFlow architecture

Cloud-native software architecture is based on container-based services (containers are a lightweight virtualization technique), which are deployed as microservices and managed on elastic infrastructure through agile DevOps processes and continuous delivery workflows. These microservices are a software development technique that structures an application as a collection of interconnected and related services. In a microservices architecture, services are simple and detailed and the protocols are lightweight.

Figure 1 provides an overview of the proposed TeraFlow OS architecture. The TeraFlow OS is a cloud-native SDN controller that is composed of multiple microservices. Microservices interact with each other using a common integration fabric. Moreover, in the context of B5G networks, the TeraFlow OS is able to interact with other network elements, such as NFV and MEC orchestrators, as well as Operations/Business Support Systems (OSS/BSS). The TeraFlow OS controls and manages the underlying network infrastructure, including transport network elements (optical and microwave links), IP routers, as well as compute nodes at edge or public cloud infrastructures.

The TeraFlow OS cloud-native architecture provides multiple benefits which have already been clearly demonstrated in other cloud computing applications. The most important benefit is application resiliency, where microservices are monitored and restarted in case of misbehaviour. Another benefit is application scalability, which accommodates increasing number of requests (i.e., load), with deployment of new instances of necessary microservices. In order to detail the different TeraFlow OS functionalities (each based on one or multiple microservices), they have been divided into two categories: core and NetApps functionalities. This classification is based on the degree of inter-relationship of these microservices as explained below.

A. *TeraFlow core services*

TeraFlow core microservices are tightly inter-related and collaborate with each other in order to provide a complete smart connectivity service. Once a Transport Network Slice request is received, the Slice Manager translates this request to an L0/L3 microservice. Moreover, the slice request is recorded by the DLT component in the blockchain. The L0/L3 microservice computes the necessary connectivity services and requests the necessary network element configuration (e.g., NETCONF, P4, OpenFlow), or interacts with underlying SDN controllers through the Device Management component. These configurations are also recorded using Distributed Ledger component. Policies per flow are verified, and network elements are monitored for anomalous behaviour in the Automation and Policy Management components. The Context Manager is responsible for handling the distributed non-relational database that contains all necessary information.

B. *TeraFlow netApps services*

TeraFlow netApps consume TeraFlow core microservices. The TeraFlow NetApps provide the necessary carrier-grade features with dedicated focus on: load-balancing, cybersecurity, auto-scaling, self-healing, and inter-domain smart connectivity services. Load-balancing allows the distribution of flow and slice requests among the microservices component replicas. The Cybersecurity component provides artificial intelligence/ML based mechanisms to detect network intrusions and harmful connections, and it provides countermeasures to security incidents. Moreover, the Cybersecurity component will be able to protect itself against adversarial attacks that try to spoof the detector's ML components. The Auto-Scaling component focuses on the autonomous replication of microservices to support high amount of load in terms of incoming requests. The Self-Healing component monitors microservices and per-flow status in order to apply healing mechanisms (e.g., component restart, flow redirection) both from a control and a data plane perspective. Finally, the Inter-Domain micro-service allows the interaction of a TeraFlow OS instance with peer TeraFlow OS instances which manage different domains.

IV. Life-cycle automation and high performance SDN components

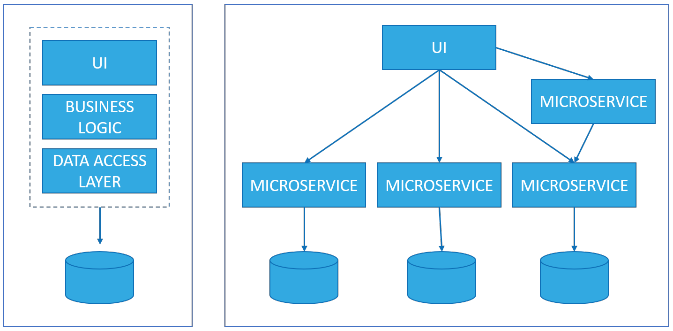

Fig. 2: Comparison between monolythic applications and micro-services

Cloud-native microservices are centred around APIs for interaction and collaboration, such as REST or Google's open source remote procedure call (gRPC). They are architected with a clean separation of stateless and stateful services. Cloud-native applications are deployed on elastic, multi-tenant, and virtualized infrastructures (see Figure 2, which compares both monolythical applications and micro-service based applications). These applications are able to auto-scale by dynamically growing and shrinking, thus adjusting themselves to the varying load. TeraFlow's cloud-native architecture has several key benefits with respect to network automation: a) self-healing properties, due to the constant monitoring of microservices and their restart in case of failure; b) auto-scaling, which allows to monitor microservices' resource consumption and scale these microservices horizontally in case of overload (path computation is a resource-consuming process which easily scales horizontally); c) load balancing, related to auto-scaling, allows balancing the load between replicated microservices; and d) automated roll-backs, which allow a declarative network status description, thus benefiting network operators with additional network programmability. TeraFlow will deliver a new generation open source cloud-native SDN controller to provide smart connectivity services to B5G networks. The TeraFlow OS architecture will consist of inter-related microservices, which are able to scale up/down according to the requested load. TeraFlow OS is expected to reap all the benefits of such a cloud-native architecture.

Operations Support Systems/Business Support Systems (OSS/BSS) may request smart connectivity services with extended constraints, such as isolation from other services. This might ensure that changes in network load or events, such as congestion or outages have no effect on the throughput or latency of the smart connectivity service. In B5G networks, a transport network [16] will provide the required connectivity to different entities in RAN and Core Network segments of an end-to-end network slice, with specific performance guarantees. In this regard the concept of a transport network slice is defined as a virtual network with a particular network topology and a set of shared or dedicated network resources, which are used to provide the network slice consumer with the required connectivity, appropriate isolation, and specific Service Level Agreement (SLA). A transport network slice could span across multiple technologies (e.g., IP or optical/microwave) and multiple administrative domains. The life cycle management of these transport network slices requires integration of IP and optical/microwave transport infrastructure. A novel B5G SDN controller should be able to deploy and control extended L2/L3 VPN services that will realize the defined transport network slices taking into consideration B5G network requirements (e.g., isolation). Multi-layer coordination is also required in order to guarantee the strict requirements as well as to provide the best resource efficiency. Thus, a novel B5G SDN controller shall be able to control a heterogeneous pool of network elements, such as disaggregated optical equipment, microwave network elements, programmable switches (e.g. P4), IP routers (virtual or physical).

TeraFlow will produce the first SDN controller to provide transport network integration, offering a feature-rich northbound API based on IETF's transport network slicing as well as providing multi-layer control of IP routers, P4 switches, and transport network elements (microwave/optical). This integration will allow the benefits from novel lower layer features to be exploited by upper layer management and control components in order to improve B5G networks.

V. Network security and integration support

TeraFlow OS will need to face against different threats at different planes. Management and control plane will expose capacities that could be impacted by multiple vector attacks: from insiders (e.g., unauthorized configurations), to exposed interfaces (northbound, southbound or east-west bound), to SDN applications. At the data plane, the TeraFlow OS will be exposed to both classical and advanced network attacks (e.g., DDoS, malware, traffic manipulation, physical-layer intrusions, etc.). Moreover, the inclusion of ML components as applications of the SDN controller will add a new threat surface that can be utilised by the so-called adversarial techniques that try to fool ML components by introducing small perturbations in the input that cannot be perceived by humans.

Both the massive amounts of information flowing through the network infrastructure and the very short latencies in threat detection impose limitations to current Intrusion Detection Systems that perform ML-based network management. To cope with this problem, TeraFlow proposes a two-layer ML-based architecture with a central ML engine and ML-based threat detectors placed at the edge nodes. The use of this advanced distributed architecture supported by gRPC telemetry data and network flows, will help to detect and mitigate the above mentioned arising threats. In some specific data plane attacks, it will be necessary to deploy distributed detection engines at the edge, with specific inference ML models that should be developed, to solve the attacks close to origin. The use of simulated environment based on real NFV/SDN Telecom infrastructures [17] will allow to generate traffic, train models and deliver accurate inference ML engines to the edge and to the Cybersecurity net application, for early mitigation.

To ensure the resilience of TeraFlow ML models against adversarial attacks, two recently released open source libraries (Cleverhans and Foolbox) will be used for designing defences in ML-based components and to test them against sophisticated adversarial attacks. In addition, TeraFlow will study the design of ML models to be deployed in resource-limited edge nodes using Automated Machine Learning techniques (e.g. Neural Architecture Search and AutoML-Zero) that allow the automated construction of a machine-learning pipeline on a limited computational budget following a "Green AI" approach. Furthermore, and regarding the lack of publicly available network and attack data for training and testing ML algorithms, TeraFlow will apply Generative Adversarial Networks (GANs) for the generation of large amounts of high-quality synthetic network and attack data.

Finally, TeraFlow will develop un-, semi- and supervised learning approaches for multi-layer network security monitoring, with additional embedded intelligence, using standard interfaces, containerization and load balancing, while paving the way towards carrier-grade deployment of ML-based security monitoring [18].

TeraFlow will deliver a permissioned distributed ledger that utilizes blockchains for network management. Furthermore, the network entities, their services, and the components of the TeraFlow OS will interact with the ledger through dedicated smart contracts. TeraFlow will provide a decentralized, robust, and trustworthy solution for storing, querying, and processing critical data for network resources and services. TeraFlow will contribute to blockchain technologies by providing research results on consensus algorithms and research on tools for analysing the security of smart contracts [19].

Current 5G networks are orchestrated using templates and configuration parameters which are hand-tailored and do not provide specifics of the underlying topology of resources (both network and cloud). Edge computing resources are introduced to provide NFV infrastructure, but exploiting the possible benefits of capillarity and lower latencies that bringing computation to edge provides. In this sense, for B5G networks it is necessary to improve the tight relationship between the management and orchestration (MANO) layer and the underlying network layer (physical and virtual), and to provide MANO the full potential of the capillarity of smart connectivity infrastructure. This can be achieved by providing more intelligence to the SDN controller responsible for L2/L3 connectivity services, thus providing more visibility over the edge hosts (such as Akraino or OpenNess) and cloud networking (e.g., Kubernetes, OpenStack) configuration. Later, interfaces from MANO to the SDN controller need to be extended in order to consider novel resource allocation techniques that might benefit virtual networks, such as deterministic networking resource allocation or dynamic location-aware resource placement.

The adoption of the TeraFlow architecture will allow service operators to exploit the benefits of joint orchestration of

TABLE I: Comparison of proposed features

	ONOS	ODL	μONOS	ODL Simple	TeraFlow
Micro-service Architecture			++	+	+++
Supported data models	+++	+++	+	+	++
Security mechanisms	+	+			+++
Integration support	+++	+++	++		+++

network, computing, and storage resources, which are believed to be fundamental requirements for such services. TeraFlow commits to fully accommodate all the stringent quality of service requirements imposed by these services, by jointly addressing the concomitant challenges regarding low latency, high bandwidth, and decentralized processing.

VI. Evaluation of proposed features against state of the art

Notwithstanding the increasing interest in the SDN paradigm, the adoption by network operators is being limited due to some restrictions of the current controller solutions, despite being modular, current controllers are still based on monolithic architectures. Operators require an extended flow processing capacity and management capabilities to cope with the requirements of nation-wide operator networks. For that reason, initiatives such as ONOS or ODL are evolving into cloud-native architectures.

Table I sumarizes the main addressed topics by each SDN controller: micro-service architecture, suported data models, security mechanisms and NFV MANO support. Regarding micro-service architecture, μONOS and ODL simple provide basic disagregation support of the SDN controller in microservices. They propose basic network device configuration microservices, and all network control complexity to be implemented in micro-service applications. TeraFlow provides internal support for network control, thus providing multiple microservices to support all network control and management features. As ONOS and ODL are mature SDN controllers, the variety that they support is completely wide in comparison of the presented data models of μONOS and ODL simple. TeraFlow will provide support for L2/L3VPN services, ONF Transport API, P4 and others, which is a significant variety of data models. Security mechanisms are very low in current SDN controllers and they are offered as external applications. μONOS and ODL simple have not considered yet the deployment of security measures. TeraFlow will provide a Cybersecurity solution for protecting TeraFlow infrastructure against attacks at optical/packet layers. Moreover, TeraFlow will also provide a trustworthy, privacy-aware, and resilient platform for storing, querying, and processing data about network resources and services. Finally, integration support for beyond 5G networks is also a significant feature for an SDN controller. ONOS, ODL and μONOS have been demonstrated to support network integration. TeraFlow offers an NBI to NFV/MEC orchestrator to provide connectivity services. Moreover, the provisioning of smart connectivity services for inter-domain scenarios is also included.

VII. Conclusion

This paper has presented the revolutionary features and architecture proposed by TeraFlow. We have compared with current SoA the proposed features. TeraFlow will foster and drive a new wave of innovation in SDN controllers.

Acknowledgment

Work partially supported by the EC H2020 TeraFlow (101015857) and Spanish AURORAS (RTI2018-099178-I00).

References

[1] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: enabling innovation in campus networks," *ACM SIGCOMM computer communication review*, vol. 38, no. 2, pp. 69–74, 2008.

[2] "Functional requirements for transport api," *ONF technical recommendation ONF TR-527*, 2016.

[3] EU 5GPPP Project 5G Exchange, "5gex business and economic layer. deliverable 2.3," 2018.

[4] S. Secci, A. Diamanti, J. Sanchez, M. Vilchez, M. T. Bah, P. Vizarreta, C. Machuca, S. Scott-Hayward, and D. Smith, "Security and performance comparison of onos and odl controllers," 2019.

[5] Open Networking Foundation (ONF), "Open Network Operating System (ONOS)," 2021. [Online]. Available: http://onosproject.org/

[6] "OpenDayLight (ODL)," 2021. [Online]. Available: https://www.opendaylight.org/

[7] P. Berde *et al.*, "Onos: towards an open, distributed sdn os," in *Proceedings of the third workshop on Hot topics in software defined networking*, 2014, pp. 1–6.

[8] J. Medved *et al.*, "Opendaylight: Towards a model-driven sdn controller architecture," in *Proc. of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks 2014*. IEEE, 2014, pp. 1–6.

[9] Open Networking Foundation (ONF), "Micro ONOS (uONOS)," 2020. [Online]. Available: https://opennetworking.org/news-and-events/blog/%C2%B5onos-and-ng-sdn/

[10] Linux Foundation, "ODL micro project," 2020. [Online]. Available: https://wiki.opendaylight.org/display/ODL/ODL+Micro+Project

[11] I. Rosenberg *et al.*, "Adversarial learning in the cyber security domain," *arXiv preprint arXiv:2007.02407*, 2020.

[12] C.-H. Huang, T.-H. Lee, L.-h. Chang, J.-R. Lin, and G. Horng, "Adversarial attacks on sdn-based deep learning ids system," in *International Conference on Mobile and Wireless Technology*. Springer, 2018, pp. 181–191.

[13] Open Networking Foundation, "Sdn architecture overview," 2013. [Online]. Available: https://opennetworking.org/wp-content/uploads/2013/02/SDN-architecture-overview-1.0.pdf

[14] ETSI, "Network transformation; (orchestration, network and service management framework)," 2020.

[15] NetWorld2020, "Whitepaper on service level awareness and open multi-service internetworking," 2016.

[16] R. Vilalta *et al.*, "Transport api extensions for the interconnection of multiple nfv infrastructure points of presence," in *Optical Fiber Communication Conference*. Optical Society of America, 2019, pp. W1G–2.

[17] A. Pastor *et al.*, "The mouseworld, a security traffic analysis lab based on nfv/sdn," in *Proceedings of the 13th International Conference on Availability, Reliability and Security*, ser. ARES 2018. New York, NY, USA: Association for Computing Machinery, 2018.

[18] M. Furdek *et al.*, "Machine learning for optical network security monitoring: A practical perspective," *Journal of Lightwave Technology*, vol. 38, no. 11, pp. 2860–2871, 2020.

[19] M. Rodler, W. Li, G. O. Karame, and L. Davi, "Sereum: Protecting existing smart contracts against re-entrancy attacks," *arXiv preprint arXiv:1812.05934*, 2018.