ACCEPTED FROM OPEN CALL

End-to-End Intent-Based Networking

Luis Velasco, Marco Signorelli, Oscar González de Dios, Chrysa Papagianni, Roberto Bifulco, Juan Jose Vegas Olmos, Simon Pryor, Gino Carrozzo, Julius Schulz-Zander, Mehdi Bennis, Ricardo Martinez, Filippo Cugini, Claudio Salvadori, Vincent Lefebvre, Luca Valcarenghi, and Marc Ruiz.

To reap its full benefits, 5G must evolve into a scalable decentralized architecture by exploiting intelligence ubiquitously and securely across different technologies, network layers, and segments. The authors propose end-to-end and ubiquitous secure machine learning (ML)-powered intent-based networking (IBN).

Abstract

To reap its full benefits, 5G must evolve into a scalable decentralized architecture by exploiting intelligence ubiquitously and securely across different technologies, network layers, and segments. In this article, we propose end-to-end and ubiquitous secure machine learning (ML)-powered intent-based networking (IBN). The IBN framework is aware of its state and context to autonomously take proactive actions for service assurance. It is integrated in a zero-touch control and orchestration framework featuring an ML function orchestrator to manage ML pipelines. The objective is to create an elastic and dynamic infrastructure supporting per-domain and end-to-end network and services operation. The solution is supported by a radio access network and forwarding plane, and a cloud/edge virtualization infrastructure with ML acceleration. The resulting framework supports application-level resilience and intelligence through replication and elasticity. An illustrative intelligent application use case is presented.

Introduction

The new generation of real-time (RT) mission-critical applications require high-resilience and low-latency coordinated actions. To that end, 5G and beyond (B5G) infrastructures must make extra decisions at the network edge [1], faster and more reliably. In that regard, the proliferation of autonomous devices sensing, communicating, and acting within their environments is posing unprecedented challenges in terms of the generated data at the network edge. This massive amount of data cannot be conveyed to the cloud without incurring large delay and high capacity. To solve this scalability challenge while addressing privacy, latency, reliability, and bandwidth efficiency, intelligence needs to be pushed to the network edge, while exhibiting tight coordination among radio access network (RAN), transport, and computation resources.

However, the relation between applications and the infrastructure is currently limited and mainly focused on provisioning aspects, which is managed independently at every network segment. Current efforts are focused on defining a control and orchestration architecture for the RAN (Open RAN Alliance, O-RAN [2]). The architecture consists of a hierarchy of systems, where RAN intelligent controllers (RICs) are close to the network and provide near-RT operation (i.e., times from 10 ms to 1 s) and abstraction to the service management and orchestration, which involves non-RT operation (i.e., times above 1 s) [2]. In parallel, other initiatives have proposed orchestration solutions considering end-to-end (E2E) service creation and operations for the RAN, transport, and computing from edge to cloud [3].

Smart orchestration requires collecting and analyzing large amounts of data, not only related to RAN, transport, and computation key performance indicators (KPI) [4], but also from the applications to deal with the committed quality of experience (QoE). A recent study anticipates that application and infrastructure monitoring tools will be augmented with machine learning (ML) capabilities over the next five years [5]. Such capabilities facilitate the adoption of the intent-based networking (IBN) concept to data center (DC) operation to simplify operation and reduce overprovisioning for service assurance [6]. Hence, IBN is receiving great attention for its application in the operators' networks context [7].

In addition, relying on ML for network and service operation requires security measures to mitigate weaknesses. Note that ML models can be subject to attacks, such as injecting malicious data to produce ML model bias when used for training, tampering telemetry data to alter ML model inference, or embedding backdoors in the ML models [8].

In this article, we propose a secure smart E2E platform targeting network and computing self-optimization to provide committed QoE to intelligent applications.

E2E Solution

For illustrative purposes, Fig. 1a shows a control and E2E service orchestration solution based on [3], allowing the deployment of E2E services. At every domain (i.e., RAN, transport, and computing from the edge to the metro/core), a technology-dependent orchestrator provides an abstracted view of the domain resources and coordinates a set of underlying software defined networking (SDN) controllers and virtual infrastructure managers (VIMs) in charge of data plane programmability.

Luis Velasco and Marco Ruiz are with Universitat Politècnica de Catalunya; Marco Signorelli is with Telecom Italia; Oscar González de Dios is with Tèlefonica; Chrysa Papagianni is with the University of Amsterdam; Roberto Bifulco is with NEC; Juan Jose Vegas Olmos is with NVIDIA; Simon Pryor is with Accelleran; Gino Carrozzo is with Nextworks, Italy; Julius Schulz-Zander is with HHI, Germany. Mehdi Bennis is with the University of Oulu; Ricardo Martinez is with CTTC/CERCA; Filippo Cugini is with CNIT; Claudio Salvadori is with NGS; Vincent Lefebvre is with Tages Solidshield; Luca Valcarenghi is with Scuola Superiore Sant'Anna.

Digital Object Identifier:
10.1109/MCOM.101.2100141

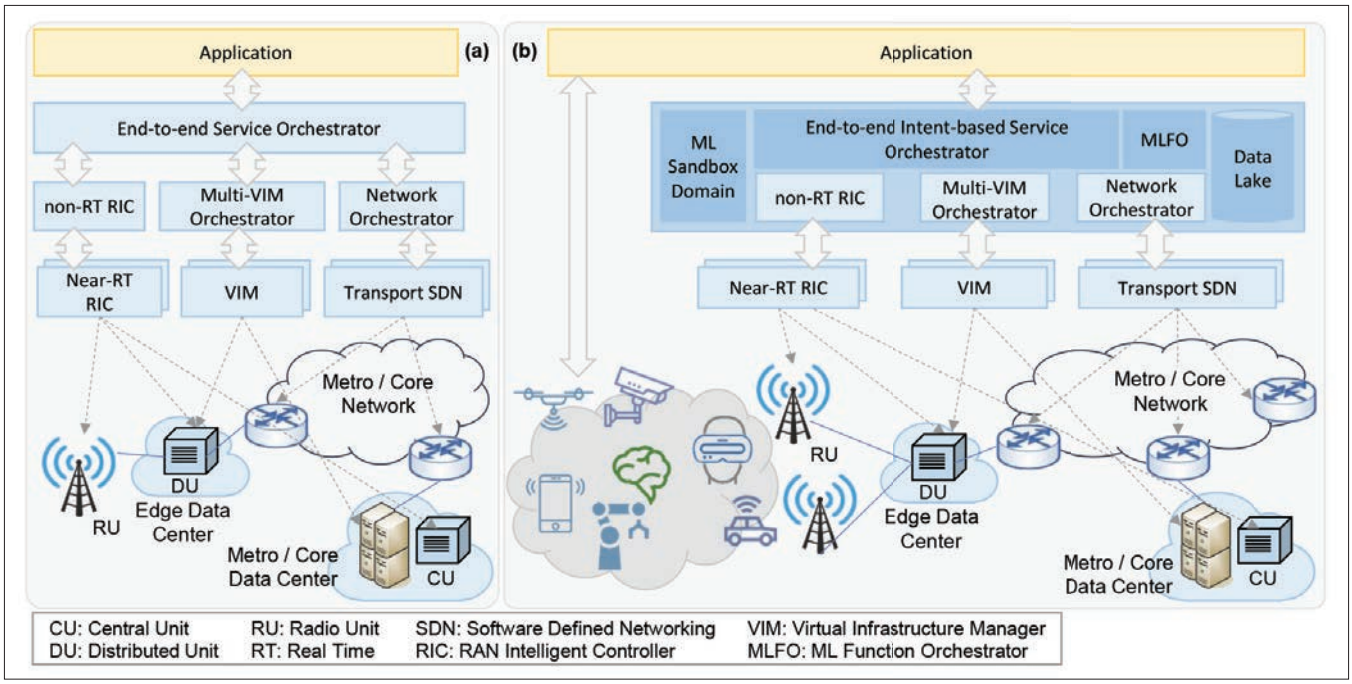

FIGURE 1. a) End-to-end orchestration; b) end-to-end IBN framework.

The proposed solution augments such architecture with an E2E intent-based service orchestration layer to deploy E2E services with tighter coordination among domains, as well as for E2E service assurance and automation (Fig. 1b). The solution is data-driven, that is, measurements are continuously collected from the data plane by the domain controllers, processed and analyzed, and an abstracted representation is exported to the E2E orchestrator. The received data are then correlated with context, analyzed, abstracted, and made available for the upper-layer applications [9]. The logical functions that implement collection, data processing and analysis, and so on are part of the intents and are connected to create an *ML pipeline* to provide policy-based network/service automation. All this is part of the IBN and closed loop automation solution.

The adopted technologies support near-RT resource allocation to adapt services to changing network conditions from both the per-domain and E2E perspectives. ML algorithms are in charge of, among other factors, predicting changes in traffic patterns and cell loads, anticipating service degradation, and detecting anomalies at early stages. With such information, optimization algorithms, forming part of the defined ML pipelines, can make proactive decisions finding the best resource configuration to deal with the future network conditions focused on service assurance. Note that changes in future conditions impacting KPIs (e.g., increasing latency) detected in one domain (e.g., in the RAN) might require reconfiguration of resources in a different domain (e.g., more capacity in the transport) or network-wide (e.g., E2E recovery).

An important aspect is the programmable data plane, tailored to meet multiple objectives like measurability and observability, elastic networking and reliability, and embedded security. Quality of service (QoS) telemetry needs to be E2E, from terminals to the cloud, with high accuracy and sub-millisecond granularity. In this context, telemetry data feeds ML algorithms for training, inference, and rapid detection of anomalies and performance degradations, which makes the data plane highly predictable and reliable, and includes embedded security to create a distributed barrier to mitigate distributed attacks. In the computing and virtualization platform, software and hardware ML accelerators and high-capacity inter-data-center interconnects create a cloud-to-edge continuum to support the ML sandbox domains and the applications.

Intelligent applications can benefit from the devised platform, as it enables dynamic resource adaptation, including the placement of virtual functions and connectivity services for perceived zero latency and application-level resilience to achieve superior QoE.

The next sections tackle the key components of the proposed solution.

Smart Control and Orchestration

Secure Intent and Closed Loop Automation

IBN complements orchestration functions by abstracting operational processes and focusing on behavior. The proposed solution starts from the design tool in the orchestrator; service definition uses templates specifying the intent in terms of policy rules that guide the service behavior, analytics, and closed loop events needed for elastic service management. The solution includes the translation and validation of the intent into a network configuration. The ML pipeline associated with the service is also created [10]; it consists of a set of ML logical functions that are combined to form an analytics function, which is managed by an ML function orchestrator (MLFO) and hosted in a variety of network functions. The optimal network configuration for the services and the related ML pipeline are computed before the deployment.

During the service life cycle, the service assurance system enforces that the network continues to deliver that intent based on the specified design,

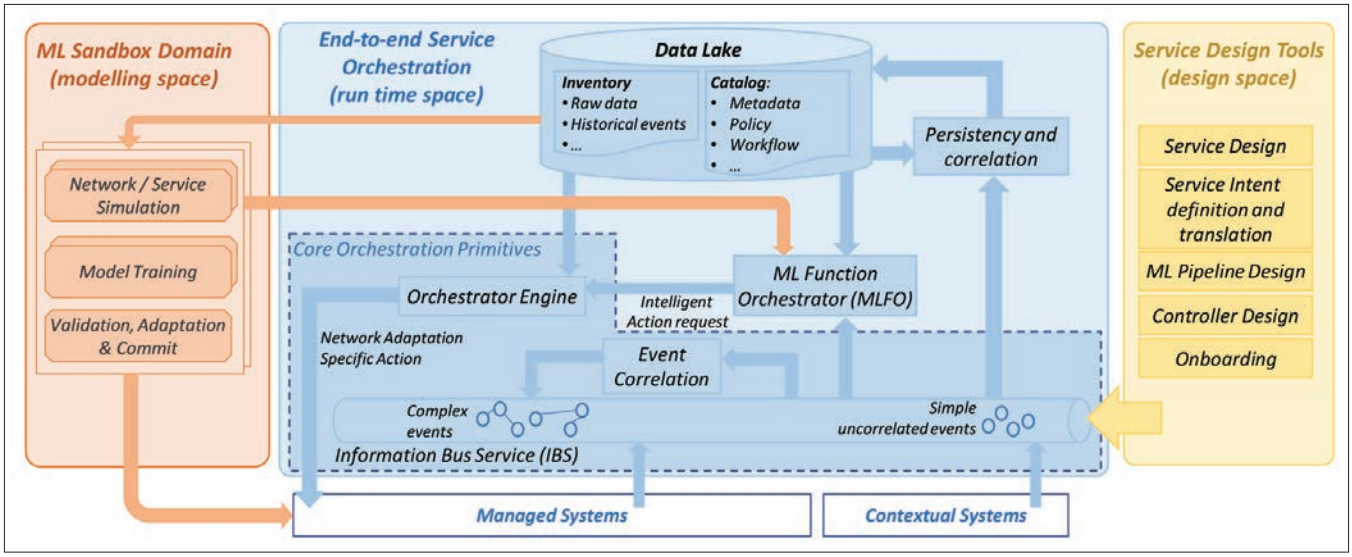

FIGURE 2. End-to-end service orchestration framework architecture.

analytics, and policies, and with the help of ML algorithms; training can be carried in an ML sandbox domain based on data from the network and simulation [11]. The target is to deal with situations ranging from those that require elastic resource scaling or reallocation to fulfill eventual demand variations, to those that require healing and recovery. Actionable insights and rich context together with policy-driven closed loops can take automated actions whenever the network deviates from the intent. The next scenarios are subject to special attention and addressed from an E2E perspective: the automated provisioning phase; service assurance, anomaly and degradation detection, and resource reconfiguration; and resources assignment under competition for resources.

The solution implements IBN at both levels: E2E orchestration and per-domain. Intents are propagated from the service orchestrator to the domains, and thus, coordination of the different domain intents is needed. The functionalities for multi-domain ML pipelines are part of those of the interface between the E2E service orchestrator and the domains. The solution goes beyond standalone ML algorithms by considering both vertical (customer and infrastructure) and horizontal relations among intents and leveraging on transfer knowledge techniques [12].

Data, ML models, and inference require confidentiality and integrity. For this reason, a centralized ML security enforcer, running in a trusted execution environment, generates and delivers execution tokens. Tokens are generated according to trust received from various sources with trust metrics and attack alerts to enforce different security policies, and considers the interaction between the distributed ML nodes. Tokens are delivered to ML nodes and used on the inference process or on specific kernel modules.

E2E Service Orchestration

The orchestration framework is designed to enable *automatic* and *zero-touch* network configuration. Containers can be used to deploy applications and allow for self-healing and horizontal scaling leveraging lightweight virtualization and orchestration. In addition, a new level of flexibility is provided through the serverless computing paradigm, where resources are allocated on demand only while the function is running. Both computing models can be used together, and this flexibility in virtualization enables the dynamic resource allocation based on application requirements.

Different layers/spaces coexist (Fig. 2). The layer including B5G *managed systems* includes orchestrators and functions for heterogeneous computing and networking resources from edge to cloud; and systems supporting context-dependent non-network activities that might help in the decision making process.

Simple events generated by the managed systems are published as topics to the information bus service (IBS). The *Event Correlation engine* is devoted to analyzing and discovering correlation between events, highlighting possible common patterns that might be indicators of a network/service degradation, guaranteeing fast anomaly detection. The output is an additional set of *complex events* that are consumed by the *MLFO* and the *Persistency and Correlation* engine. The Orchestration engine receives network reconfiguration requests from the MLFO, translates them into specific configurations for the network and the virtualization systems, and then executes the proper set of actions for the requested configurations.

The Data Lake system represents the logically centralized repository where data are stored, aggregated, and transformed; it includes structured data from relational databases, semi-structured data, unstructured data, and binary data. Finally, the MLFO manages and monitors all the elements building an ML pipeline. The MLFO makes its orchestration decisions based on the data/events retrieved by the IBS and the Data Lake and on the current performance status of the ML Modules trained in the ML Sandbox domain. When needed, the MLFO will send complex re-configuration requests to the Orchestration Service for service assurance.

Adaptive Network Operations

To deal with the complex B5G manageability, the industry is increasingly adopting an open, software-defined, virtualized, and disaggregated RAN

(vRAN) and is pushing for automated ML control plane functionality [2]. The latter includes requirements for dynamic reconfiguration of the B5G vRAN, like service creation time in the sub-second timescale. In addition, the O-RAN architecture enables the application of ML techniques for RT (below 10 ms), as well as near-RT and non-RT control.

The proposed solution is O-RAN-aligned, with an ML-extensible Open vRAN control plane based on a containerized cloud-native architecture, supporting zero-touch operation and reconfiguration. Telemetry and control plane data from location/positioning subsystems are ingested into the near-RT RIC, which provides the extensible framework for advanced B5G network operation at the edge. vRAN network function placement is also service-aware to fulfill service performance via intents, for example, for low-delay ultra-reliable low-latency communications (URLLC) services. As in the O-RAN architecture, the non-RT RIC enables non-RT control and optimization of RAN elements and resources, ML pipeline, and policy-based guidance in the near-RT RIC via O1/A1 interfaces [2]. Non-RT and near-RT RICs fine-tune RAN behavior to assure specific KPIs dynamically; the non-RT RIC monitors long-term trends and patterns and train models to be deployed at the near-RT RICs, and it exposes an intent-based application programming interface (API) providing high-level abstraction as a northbound interface to the E2E service orchestrator [13].

Regarding the transport network, it encompasses both metro and core network segments supporting heterogenous technologies (e.g., packet and optical) to provide the required transport capacity and connectivity from edge to core. Such network services (NSs) are requested by the E2E service orchestrator and processed by the network orchestrator. As for the RAN, an intent-based API providing high-level abstraction is used. Several SDN controllers handle the actual programmability of a set of network devices within a defined area. The definition of the areas can follow different criteria such as geographical, aiming to reduce control latency, technology/vendor, and so on. Thereby, the network orchestrator coordinates operations with the involved SDN controllers when a network service is deployed or reconfigured.

The considered network control and orchestration architecture is devised to provide fast and effective network automation to permanently ensure and preserve the performance requirements of the network services. To this end, closed loop automation at different levels is adopted within a single area or E2E. Depending on each level and the complexity, the goal is to complete the provisioning and re-configuration processes in a sub-second timescale. Closed-loop automation entails gathering performance monitoring data from different sources, including in-band network gelemetry (INT) [14], active and passive probes, and so on, via a gelemetry API.

Programmable Data Plane

The proposed solution includes a programmable RAN and transport a forwarding plane based on P4 [14] for more granular control of the forwarding plane, as well as a computing platform with ML acceleration that extends from edge to cloud.

Radio Access Network

To make RAN topology configuration more flexible, the RAN must be highly programmable. ML techniques can be adopted to implement radio resource scheduling and multihop path selection and RAN configuration in tight coordination with the control and orchestration architecture. The O-RAN architecture [2] is built to offer an open functional architecture that can integrate and implement these functionalities (Fig. 1b).

On a per-service basis, customized RT control and protocol support can be the market differentiator enabling novel application-network interactions. Moreover, there is a need for flexible cross-service resource sharing, depending on the level of isolation and RT performance requirements. Scheduling schemes and new protocols are instantiated on demand at the RAN: i) on a per-service manner to control the behavior of the service traffic, or ii) across services to, for example, ensure fairness and performance requirements. Toward that end, appropriate low-level hardware scheduling primitives need to be defined and used by the disaggregated cell site, programmed in high-level languages such as P4.

Scheduling performed together with RT control loops still needs policies received from the near-RT RIC. The near-RT needs to be updated periodically based on the state of the network, as well as on contextual data, which is also provided by the non-RT RIC and, ultimately, the E2E service orchestrator. The scenario is even more complex in ultra-dense deployments with overlapping coverage areas and multihop self-backhauled networks with dynamic cell-less-based topology reconfiguration.

Forwarding Plane

Extending the current forwarding plane solutions with programmable traffic management and cross-layer interactions provides significant benefits for meeting the diverse requirements of different services. The application of customizable and programmable traffic management is supported E2E, focusing on bottlenecks along the traffic forwarding paths. In addition, a highly programmable network forwarding plane provides:

- *Measurability.* Telemetry streams are activated among network nodes and controllers to provide continuous and accurate monitoring of node performance with sub-second granularity. INT is also extensively exploited to retrieve accurate statistics of selected services on a per-packet basis and enforced at the terminal side, enabling the collection of precise geo-localization data, while providing accurate E2E connectivity monitoring.
- *Elastic networking.* Novel elastic network technologies are being designed to support changing network topologies at runtime. Stateful extensions to P4 programmability are exploited to provide finite state machines directly in the data plane. This way, elastic E2E adaptations predicted at the orchestrator can be pre-enforced at the node, enabling the dynamic evaluation of complex conditions implemented at wire speed.
- *Reliability.* Selected application-aware packet replication can be performed at the ingress/egress sections. In addition, applica-

To make RAN topology configuration more flexible, the RAN must be highly programmable. ML techniques can be adopted to implement radio resource scheduling and multihop path selection and RAN configuration in tight coordination with the control and orchestration architecture. The O-RAN architecture is built to offer an open functional architecture that can integrate and implement these functionalities.

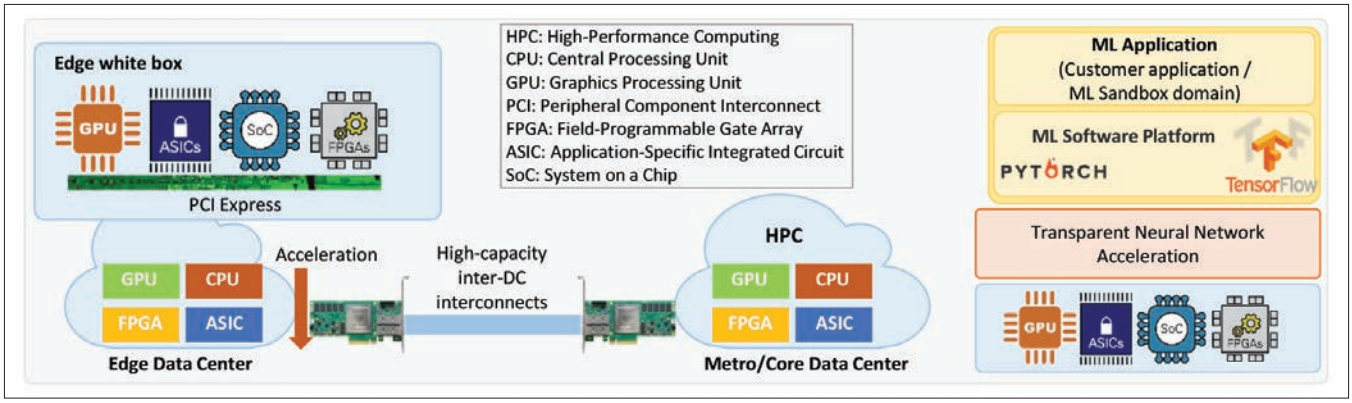

FIGURE 3. Cloud-to-edge continuum with ML acceleration.

tion-aware failure/congestion detection can be implemented.

- *In-network operations.* By leveraging on P4-based stateful features, pre-planned and dynamic countable hardware resources can be dynamically allocated to implement aggregation functionalities.
- *Embedded security.* Stateful and inspection functionalities are exploited to perform cyber-security firewalling at wire-speed (e.g., against distributed attacks). ML feature extraction and telemetry-enabled correlation of network events among different nodes can be exploited to build a distributed barrier for smart anomaly detection.

Edge-to-Cloud Platform with ML Acceleration

As applications become more intelligent, more decisions are made at the edge. To support foreseen B5G application scenarios, the computing platform (Fig. 3) needs to be deployed as a one-stop white box at the edge; this is flexibly and fully interconnected to the cloud for conducting extensive workloads (e.g., those related to ML) by distributing the various processing, learning, and inference functions in a seamless and efficient way. The edge white box is based on a high-performance adapter card with internal processors, GPU units, and reprogrammable units for ad hoc processing. This platform allows the creation of a cloud-to-edge continuum, enabling workloads to be processed locally, as well as E2E network acceleration to interconnect with cloud services through high-capacity transport links when large resources need to be accessed.

Unfortunately, hardware specialization introduces significant barriers for algorithm portability, while ML software platforms are heterogeneous and in continuous evolution. Given such heterogeneity, there is a need to face the challenge of supporting ML applications with seamless portability. Such portability can be supported by the introduction of a novel compiler technology that translates an ML algorithm into a device-specific runnable binary, which is ready to be deployed and optimized for the given device. This would enable scalable function placement over the computing platform from edge to cloud, independent of the specific hardware resources.

Finally, the platform relies on containers orchestration to provide a portable, extensible, open platform for managing containerized workloads and services to facilitate declarative configuration and automation.

Intelligence at the Edge

Intelligent Applications

The possibility to move intelligence to the edge has sparked a groundswell of interest in distributed on-device ML, in which training data is stored across many geographically dispersed nodes [1]. Training is done locally, and aggregated updates are shared with other nodes directly or through a federating server. However, a learning model may have many parameters, and hence a model update can be bandwidth consuming. Moreover, since devices have limited resources, on-device ML must minimize the size of the model running on the device and power usage, while also considering prediction accuracy and privacy constraints. Note that the applications of federated ML enabled by URLLC are instrumental in verticals such as vehicle communications among others.

The proposed orchestration, network, and computing technologies enable intelligent applications with distributed ML by providing computation and connectivity resources that provide enough QoS to support the required QoE. Furthermore, the provided infrastructure-level resiliency can dynamically and coordinatedly self-reconfigure to adapt the resources to anomalies and degradation before they impact the QoE of many applications. However, some services, such as those relying on URLLC and/or massive mobility, might still be impacted by a failure. In that regard, geo-replication can be used to provide geographical redundancy, which results in increased reliability and availability of applications against failures. Indeed, geo-replication is one of the enablers for perceived zero latency. The proposed solution provides the applications with resilience capabilities that go beyond the infrastructure level. Such functionalities include E2E QoS performance monitoring and context metadata; scaling in/out resources and extending topologies from edge to cloud to meet application requirements; and container and serverless function placement with topology adaptation that enables seamless replication and redundancy. Applications' autonomous operations are performed based on the defined customer intent and are closely followed at the infrastructure level.

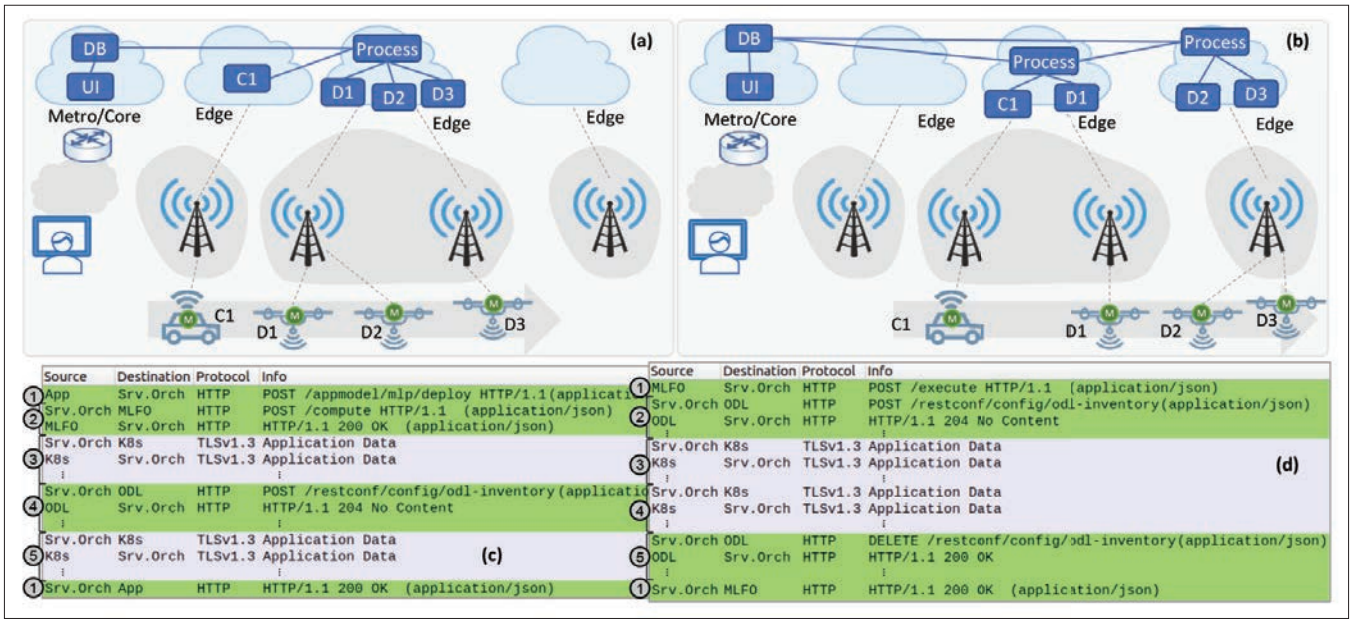

FIGURE 4. a) NS and ML pipeline deployment; b) reconfiguration; c-d) messages exchanged.

Infrastructure intents not only collect performance monitoring data, but also receive data from customer intents and use them to reconfigure resources in advance.

Illustrative Use Case

To illustrate the benefits of the proposed solution, Figs. 4a and 4b present a VR/AR application that takes advantage of application-level resilience based on federated learning. Three drones (labeled D1–D3) and one car (C1) are capturing images, and each one learns an ML model before sending it to the edge, where the model is combined with others coming from multiple sources and sent back to the terminals for better accuracy. An ML pipeline with a connected set of containers/serverless functions is deployed [15]. Nodes in the ML pipeline collect individual ML models from and distribute combined ML models to the devices (labeled with the devices' names), combine ML models (process), store data (DB), and provide user interface (UI). Model combination requires strict delay from the local ML model computation until the reception of the combined ML model. To meet the desired performance, collection and processing containers should be placed at the edge in nodes with ML acceleration, whereas other functions can be deployed at the metro/core.

The E2E latency is constantly monitored by the terminals (M) and is combined with other data sources, like the received RF signal and geolocalization-based estimated trajectory, to predict near-future QoE degradations that would impact the accuracy of the model predictions at the terminals. The ML pipeline is reconfigured accordingly.

Experimental Implementation and Results

We have prototyped a service orchestrator and an MLFO to demonstrate the use case described in the previous subsection. Both modules have been developed in Python 3.8; Kubernetes (K8s) was used as the Multi-VIM orchestrator and Docker as the container technology. The service orchestrator uses the Kubernetes API through a Python client library, and it communicates with the MLFO through a RESTFul-based interface. A private image repository was hosted in Docker Hub. OpenDayLight (ODL) was used as the SDN controller.

Figure 4c shows the messages exchanged during the deployment of the NS in Fig. 4a and the related ML pipeline. An NS descriptor is received from the application with a template for the ML pipeline and some constraints (message 1). The service orchestrator then computes the placement and connectivity for the NS and requests the MLFO to compute the ML pipeline, which computes the ML pipeline to be deployed based on the NS and the received constraints (2). Next, the service orchestrator coordinates with Kubernetes (3) and ODL (4) for the deployment of the NS and the ML pipeline. Containers' deployment is carried out through Kubernetes (5), and information includes the DC identifier, virtual LAN and IP address, the image, and other configurations. The total deployment time was below 30 s.

Figure 4d shows the exchanged messages during ML pipeline reconfiguration (as in Fig. 4b). The workflow is triggered when one or more terminals report E2E latency exceeding some threshold. This entails solving an optimization problem considering the current state. It is worth mentioning that the workflow uses replication and a make-before-break approach for seamless transitions; new connectivity is created (2) and containers are deployed (3), before removing those unused (4–5). Total reconfiguration time was 8.5 s, and we verified that no data were lost.

Summary

A framework based on IBN with ubiquitous and secure ML has been presented. Specifically, the operation of customer and infrastructure services are automated, where ML plays an important role in making predictions of future network and service conditions, detecting anomalies and degradations, and supporting coordinated decisions

Plane	Key component	Description
Control and orchestration	ML-based IBN solution	Intents are based on accurate ML models and can exchange knowledge among them to provide network services with tight coordination and assurance automation.
	Secure ML	Different heterogenous solutions to enforce data and ML model integrity and confidentiality are federated into one fail-safe overarching centralized ML security enforcer delivering tokens, itself protected by trusted execution.
	Zero-touch control and orchestration	Hierarchical architecture with an E2E service orchestrator coordinating RAN, transport network, computing specific orchestrators. Each orchestrator coordinates underlying near-RT controllers. The MLFO manages ML pipelines. Model training performed in sandbox domains with data from a Data Lake.
	Adaptive network operations and service assurance	Per-domain and E2E network and services proactive adaptation and reconfiguration based on ML and near-optimal E2E resource allocation for service assurance. Decisions are made autonomously or include the operator in the loop.
Data/ forwarding	Programmable RAN	Highly programmable ML-backed RAN that extends from scheduling to multi-hop path selection and RAN topology configuration. The E2 interface is key to cover the required functionalities [2].
	Programmable forwarding plane	Network programmability can be exploited to provide measurability, elastic networking, reliability, in-network operations, and embedded security.
	End-to-end platform from edge to cloud	Based on a one-stop white box operating in the edge with full interconnection to cloud solutions and local capability to conduct extensive workloads (e.g., ML training).
	ML acceleration	A compiler technology translates ML into a device-specific runnable binary. This enables scalable function placement from edge to cloud.

TABLE 1. Summary of the proposed solution.

for service assurance. A specific feature is the coordination between customer and infrastructure ML-based intents, aiming to simultaneously meet the requirements of all the services. All this requires RT decision making, flexible placement of functions in the computing platform, RT reconfiguration, and so on from the E2E perspective. Table 1 summarizes the key components of the proposed solution.

Acknowledgments

This work was partially supported by the AEI IBON (PID2020-114135RB-I00) project and by the ICREA Institution.

References

[1] J. Park *et al.*, "Wireless Network Intelligence at the Edge," *IEEE Proc.*, 2019.
[2] O-RAN Alliance; https://www.o-ran.org, accessed Aug. 2021.
[3] 5G PPP Architecture Working Group, "View on 5G Architecture," v. 3.0, Feb. 2020.
[4] A. Bernal *et al.*, "Near Real-Time Estimation of End-to-End Performance in Converged Fixed-Mobile Networks," *Elsevier Computer Commun.*, vol. 150, 2020, pp. 393–404.
[5] Gartner, "Market Guide for AIOps Platforms," Nov. 2019.
[6] ACG Research, "Intent-Based Networking with Apstra AOS: Market Impact Report," May 2018.
[7] "Intent-Based Networking — Concepts and Definitions," IRTF draft work in progress, Feb. 2021.
[8] "AI Security White Paper," Huawei, 2018.
[9] ITU-T Y.3174, "Framework for Data Handling to Enable ML in Future Networks Including IMT-2020," 2020.
[10] ITU-T, "Unified Architecture for Machine Learning in 5G and Future Networks," 2019.
[11] M. Ruiz *et al.*, "Modeling and Assessing Connectivity Services Performance in a Sandbox Domain," *IEEE/OSA JLT*, vol. 38, 2020, pp. 3180–89.
[12] M. Ruiz *et al.*, "Knowledge Management in Optical Networks: Architecture, Methods and Use Cases," *IEEE/OSA JOCN*, vol. 12, 2020, pp. A70–A81.
[13] ONF, "Intent NBI -Definition and Principles," ONF TR-523, 2016.
[14] F. Cugini *et al.*, "P4 In-Band Telemetry (INT) for Latency-Aware VNF in Metro Networks," *Proc. OFC*, 2019.
[15] L. Toka *et al.*, "Machine Learning-Based Scaling Management for Kubernetes Edge Clusters," *IEEE Trans. Network and Service Mnmgt.*, vol. 18, 2021, pp. 958–72.

Biographies

Luis Velasco received his Ph.D. degree from UPC in 2009. His interests include ML for service and network automation.

Marco Signorelli received his M.Sc. science from Universita degli Studi di Torino in 1990 and joined Telecom Italia.

Oscar Gonzlez de Dios received his Ph.D. with honors from the University of Valladolid. He is with Tèlefonica I+D.

Chrysa Papagianni obtained her Ph.D. in electrical and computer engineering from NTUA. She is with the University of Amsterdam, Netherlands.

Roberto Bifulco received his Ph.D. from the University of Napoli "Federico II," Italy. He is with NEC Laboratories Europe.

Juan Jose Vegas Olmos received his Ph.D. from Eindhoven University of Technology. He is a principal engineer at NVIDIA.

Simon Pryor received his M.Sc. in engineering from the University of Florida. He is the R&I strategy director of Accelleran.

Gino Carrozzo received his Ph.D. in telecommunications from the University of Pisa. His interests include SDN/NFV for 5G networks.

Julius Schulz-Zander received his PhD in computer science from Technische Universitt Berlin in 2011. He is with HHI.

Mehdi Bennis is an associate professor at the University of Oulu, Finland. His interests include ML in beyond 5G networks.

Ricardo Martinez received his Ph.D. in telecommunications engineering from UPC in 2007. His research interests include SDN/NFV.

Filippo Cugini received his MSc degree from the University of Parma. His interests include communications and networking.

Claudio Salvadori received his Ph.D. degree from Scuola Superiore Sant'Anna in 2013. He is the CEO of NGS.

Vincent Lefebvre received his M.Sc. degree from ISEN, France. He is the CEO of TAGES SOLIDSHIELD.

Luca Valcarenghi received his Ph.D. from the University of Texas at Dallas in 2001. He is with Scuola Superiore Sant'Anna.

Marc Ruiz received his Ph.D. degree in 2012 from UPC. His interests include ML for service and network automation.