Journal of Network and Systems Management (2022) 30:46
https://doi.org/10.1007/s10922-022-09651-x

Machine Learning-Assisted Closed-Control Loops for Beyond 5G Multi-Domain Zero-Touch Networks

Nathan Franklin Saraiva de Sousa[1] · **Md Tariqul Islam**[1] · **Raza Ul Mustafa**[1] · **Danny Alex Lachos Perez**[1] · **Christian Esteve Rothenberg**[1] · **Pedro Henrique Gomes**[2]

Received: 30 September 2021 / Revised: 29 January 2022 / Accepted: 25 February 2022

Abstract

End-to-End (E2E) services in beyond 5G (B5G) networks are expected to be built upon resources and services distributed in multi-domain, multi-technology environments. In such scenarios, key challenges around multi-domain management and collaboration need to be tackled. ETSI Zero-touch network and Service Management (ZSM) architectural framework provides the structure and methods for effectively delivering E2E network services. ZSM pursues cross-domain automation with minimum human intervention through two main enablers: Closed Control Loop (CCL) and Artificial Intelligence (AI). In this work, we propose a multi-domain ZSM-based architecture aiming at B5G scenarios where several per-domain CCLs leverage Machine Learning (ML) methods to collaborate in E2E service management tasks. We instantiate the architecture in the use case scenario of multi-domain automated healing of Dynamic Adaptive Streaming over HTTP (DASH) video services. We present two ML-assisted techniques, first to estimate a Service Level Agreement (SLA) violation through a Edge-based Quality of Experience (QoE) Probe, and second to identify the root cause at the core transport network. Results from the experimental evaluation in an emulation environment using real mobile network traces point to the potential benefits of applying ML techniques for QoS-to-QoE estimation at Multi-Access Edge Computing facilities and correlation to faulty transport network links. Altogether, the work contributes towards a vision of ML-based sandbox environments in the spirit of E2E service and network digital twins towards the realization of automated, multi-domain CCLs for B5G.

Keywords Beyond 5G Networks · End-to-End Network Services · Network and Service Management · Artificial Intelligence · Self-healing

✉ Nathan Franklin Saraiva de Sousa
nsaraiva@dca.fee.unicamp.br

Extended author information available on the last page of the article

Published online: 22 April 2022

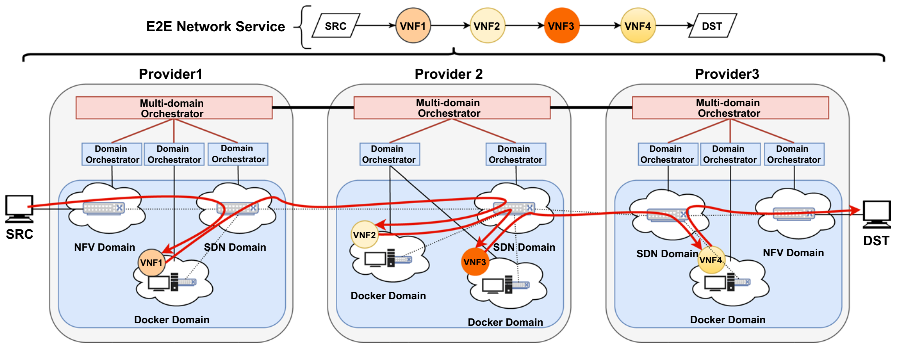


Fig. 1 E2E network service example across multiple (multi-technology and multi-administration) Domains [7]

1 Introduction

Beyond 5G (B5G) networks are expected to take advantage of technological advances such as Software Defined Networking (SDN), Network Function Virtualization (NFV), Multi-access Edge Computing (MEC), and Network Slicing (NS) to provide end-to-end (E2E) services to vertical customers. E2E services are composed of distributed resources and functions which are expected to be available across multiple network domains (e.g., radio/transport/core networks and edge clouds), each combining different technologies (e.g., SDN/legacy networks, optical or packet-switched) and administrative roles (e.g., mobile network operators and cloud service providers) [1]. In the resulting multi-domain, multi-technology environment (see Fig. 1), manually managing the E2E services becomes highly challenging and complex due to the lack of flexibility and programmatic interfaces featured by the diverse enabling components [2]. In this context, cross-domain automation of E2E service management is paramount for efficient service delivery and cost savings [3].

Multi-domain closed-loop automation aims to deliver E2E service and network management tasks (e.g., monitor, analyze, plan, and execute) spanning multiple domains ideally with no human intervention at all [4]. This emerging concept is being referred to as Zero-Touch Network and Service Management (ZSM). Towards its realization, ETSI established the ZSM Industry Specification Group (ZSM ISG) in 2017 [5]. The ETSI ZSM ISG targets to specify a scalable, extensible, and resilient reference architecture for automating complex E2E services spanning across multiple domains with different technologies and administrations. Cross-domain Closed-Control Loop (CCL) and Artificial Intelligence (AI) techniques are two key pillars of the ETSI ZSM framework reference architecture towards the full-automation of E2E management operations [6] of B5G.

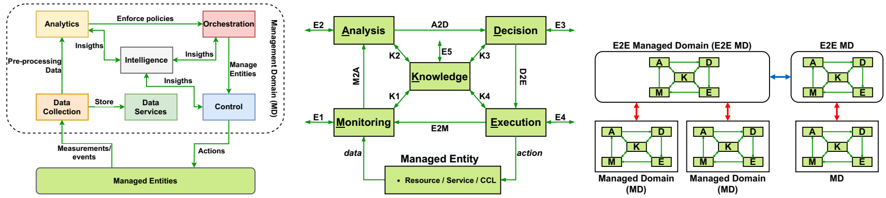
Multi-domain CCL involves the management of E2E services across multiple administrative domains. Each domain has one or more CCLs with specific goals, e.g., self-configuration, self-healing, or self-optimization. In a hierarchical architecture, each domain-specific CCL exposes services and resources to upper CCLs, named E2E CCLs. The E2E CCLs, in turn, have a holistic view of the E2E service execution. The proper collaboration between different CCLs involved in the service execution should satisfy the fulfillment of E2E service requirements [8].

At the crossroads, Machine Learning (ML) and Artificial Intelligence are gaining attraction as an enabling technology for networking [9, 10] to allow self-managing capabilities, reduce operational costs, time-to-market of services, and risk human errors. The applications of ML/AI for network and service automation include (*i*) extracting features/information from a large amount of network data, (*ii*) analyzing patterns, fault diagnosis, and predicting future situations, and (*iii*) proposing optimal actions based on the predicted metrics [11]. In terms of fault management, ML, together with the cross-CL collaboration, can contribute to Root Cause Analysis (RCA) processes in cases where of Quality of Experience (QoE) or Service Level Agreement (SLA) violations.

In this work, we present a ZSM-based architecture with self-healing functionalities in multi-domain zero-touch B5G networks. To this end, ML-assisted CCLs are used at different ZSM reference architecture points. Network-level data monitoring is used as features for ML-based analysis to estimate end service QoE Key Performance Indicators (KPIs) and run RCA to identify under-performing network links. The ML experiments serve as proof of concepts of intelligent CCL mechanisms, best-of-breed emulation-based environments with real 3G, 4G, and 5G network traces were used to provide insights on the applicability and performance of different supervised ML algorithms for QoS-to-QoE correlations and faulty network link localization. The main contributions of this paper can be summarized as follows:

- Design of a multi-domain B5G CCL architecture based on ETSI ZSM framework and illustrated in a use case scenario of ML-assisted automated healing of a Dynamic Adaptive Streaming over HTTP (DASH) video service.
- Design, implementation, and proof of concept evaluation of non-invasive network-level encrypted traffic monitoring approach at edge premises to estimate QoE from bidirectional network-level QoS features. We show a lightweight QoS features extraction technique with 0.5-second granularity to estimate the QoE of DASH video service through ML-based techniques.
- Design, implementation, and proof of concept evaluation of a RCA method where QoS metrics (Round Trip Time (RTT), throughput, and packets) and QoE KPIs are trained in an emulated version of a transport network to support the identification and localization of network link issues responsible for DASH video QoE intent violations.

This article is organized as follows. Section 2 introduces the ETSI ZSM framework regarding its functional view and multi-domain collaboration. We present a ZSM-based multi-domain CCL architecture and its applicability in an automated healing use case in Sects. 3 and 4, respectively. Section 5 presents the design of Edge-based ML-Assisted QoE Probe to run CCL. Section 6 describes a method for a Transport CCL to localize faulty links leveraging ML in a sandbox approach. Section 7 focuses on the experimental evaluation of the proposed ML methods. We point out relevant related works in Section 8 before concluding the paper in Sect. 9.



(a) Architectural Overview. Adapted from: [4] **(b)** CCL Functional View **(c)** Multi-domain CCL

Fig. 2 ETSI ZSM framework

2 ETSI ZSM Framework

This section introduces the relevant background on the ETSI ZSM reference architecture, including CCL automation within the ZSM framework, from functional and multi-domain perspectives.

2.1 ETSI ZSM Reference Architecture

The ZSM initiative under development by ETSI targets E2E service and network management automation in multi-domain environments. ETSI ZSM defines a reference architecture to provide zero-touch automated network and service management in 5G networks and beyond [6]. To enable a fully automated network and service management, the ZSM architecture is based on a set of tenets (e.g., CCL management automation, AI-assisted techniques) and orchestrates Managed Entities (MEs) to achieve a specific target.

Figure 2a depicts the ZSM Management Domain (MD), presenting a set of management services for zero-touch network and service management at a high level. It also shows the means of these services integration, communication, inter-operation, and organization. Each MD contains several management functions organized and integrated towards providing domain-specific capabilities. Some management functions are restricted to be supplied and consumed inside the management domain only, whereas other functions can expose some services and resources to authorized entities outside the domain.

2.2 ETSI ZSM CCL Functional View

ETSI ZSM CCLs can be composed of the architectural building blocks (Fig. 2a) and are represented by a functional view, as shown in Figure 2b. The functional view shows the split of the CCL functionalities, i.e., its stages and knowledge, and the data and control flows between different stages of the CCL and between the CCL and outside entities [8].

Specific models have been developed to represent the ZSM-based CCLs and gather the requirements for managing the entire lifecycle of CCLs. Each CCL should have at least one goal, at least one managed entity, one or more CL stages,

Table 1 Closed control loop models

Attribute	Description
(a) Goal model	
Id	It indicates the identifier of the CL instance goal
Description	Describes the closed-loop goal in a human-readable form
Statement	The goal statement can be a declarative or an imperative statement
(b) Managed entity model	
Id	It indicates the identifier of the managed entity
Type	It indicates a type of managed entity: resource, service, or CCL
(c) Component model	
Description	Describes the component functionality in a human-readable form
Input data list	Lists the information the component can receive from other entities
Output data list	Lists the information the component can provide to other entities
Produced Mngm. list	Lists the capabilities offered for consumption by authorized entities
Consumed Mngm. list	Lists the capabilities consumed by the component for its functioning

Adapted from: Ref. [12]

and at most one knowledge base. CCLs act on the target managed entities to meet the desired objective supported by insights from the knowledge base.

The CCL goal steers its operation towards the fulfillment of the expectations in terms of service quality level. The goals can be specified through declarative statements represented by intents or imperative statements defined by, for instance, a Service Level Specification (SLS). At design time, the goal is set up with reference values; however, these parameters can be changed in run time to adapt to the environmental changes. Table 1(a) presents the parameters of the CCL goal model.

The types of managed entities that ZSM-based CCL can actuate upon include managed resources, services, or other closed loops. A CCL manages the resource and services of a management domain that it has direct access to. However, when the service spans different domains is required the collaboration of multiple distributed CCLs for the proper execution of service. The main attributes of managed entities are presented in Table 1(b). CCLs are composed of management components responsible for performing CCL operations following a sequential flow. Stages and knowledge base, presented in Figure 2b, are examples of CCL components. Table table:ccl-models(c) describes the main attributes of CCL components.

The functional view also includes five stages that compose the ETSI ZSM CCL chain:

Monitoring Collect/pre-process the performance of MEs.

Analysis Derives insights from monitored data (obtained in the monitoring stage) as well as historical data.

Decision From the analysis stage's insights, this stage makes a plan/workflow on the reactive, proactive, or predictive actions.

Execution Carries out the workflows towards MEs when the decision stage determines the required actions. The execution stage can also steer the state of another CCL when the automation of the E2E service management is required.

Knowledge It is a logical center for storing and retrieving data that can be shared between the stages within a CCL and/or between different CCLs. Also, intelligence mechanisms (e.g., rules, policies, ML/AI) can be used between the monitoring and execution stages to translate data into knowledge.

Multiple data and control flows may also run concurrently in a CCL chain:

Primary flow Executed right after receiving data from the underlying MEs and involves the main 4 CCL stages. M2A, A2D, D2E, and E2M in Fig. 2b represent the transitions between the primary flow stages.

Knowledge-enabled flow An augmented version of the primary flow that uses data (real-time or historical) stored/retrieved from a knowledge stage. K1, K2, K3, and K4 in Fig. 2b represent data retrieved from and stored to the knowledge.

Customization flow Used for interactions between different CCLs and between CCLs and external entities, such as human operators or other management systems. E1, E2, E3, E4, and E5 in Fig. 2b represent data and control inputs and outputs, such as start/stop process, change the stage's settings, retrieve the current status and/or historical data, and/or real-time data of a stage, provide the resulting data of the stage to other CCL and/or authorized external entities.

2.3 ETSI ZSM Multi-domain CCL

ETSI ZSM multi-domain collaborative CCLs can be arranged from their functional views. Figure 2c shows an example of E2E CCL management, where each MD features its own CCL functional components. In turn, an E2E MD communicates with CCLs from individual MDs to inter-work and collaborate in managing one or more E2E services. The relationship between two or more CCLs within the ZSM framework can be classified as hierarchical (red arrows) or federated/peer (blue arrows) CCLs [12]. The hierarchical relationship is the case when one CCL implements intra-provider CCL collaboration. In contrast, federated relationships demand inter-provider CCL collaboration.

Hierarchical model Each domain-specific CCL is responsible for self-managing (self-healing, optimization) its domain. However, local actions do not always result in a global optimum or repair of a global fault event – they can even generate conflicts with other CCLs. Hence, E2E CCL coordinates the interaction

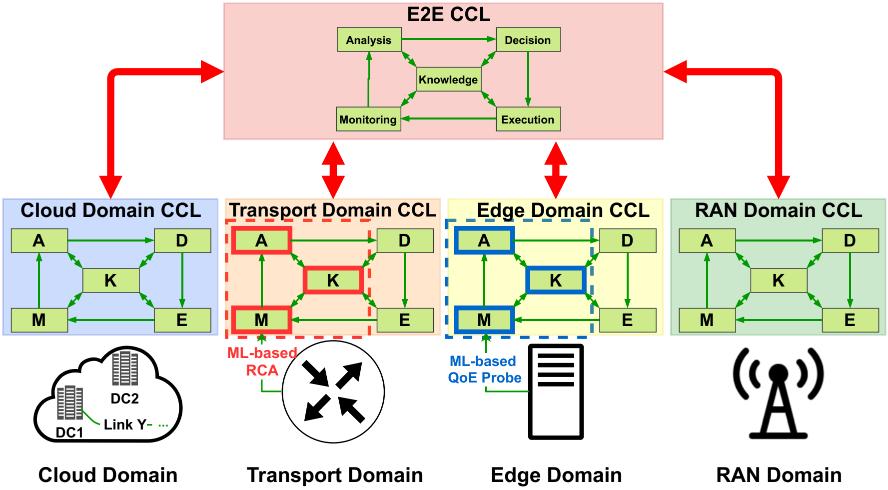


Fig. 3 Multi-domain CCL architecture

between different CCLs to mitigate such issues. This process occurs through determined actions including delegation and escalation. In the case of delegation, the E2E CL delegates policies to the underlying CCLs to act autonomously, whereas in the escalation, the process occurs inversely, the domain-specific CCL cannot achieve the desired goal and escalates the issue to the E2E CCL.

Federated model Consists of integrating two CCLs, enabling exchanging information and sharing resources and services. However, one CCL is not responsible for the other, and both exist independently.

3 Multi-domain CCL Architecture Design

Following the ETSI ZSM architectural framework principles, we now present the design of a multi-domain CCL system architecture shown in Figure 3. Six fundamental characteristics have been identified to implement cross-CCL collaboration: (*i*) Multi-CCL operation, (*ii*) Hierarchical collaboration model, (*iii*) Intent and Policy Management, (*iv*) Modularity, (*v*) Smart CCL, and (*vi*) Workflow.

Multi-CCL operation Multiple CCLs are created on the individual MDs and the E2E MD. More specifically, CCLs in Radio Access Network (RAN), Edge, transport, and cloud domains are considered to automate individual MDs' local operations. The corresponding E2E CCL domain aims at

	interconnecting CCLs from individual MDs to manage multi-domain E2E services in a fully autonomous way collaboratively.
Hierarchical collaboration model	The higher E2E CCL domain is responsible for and has administration towards the lower-level CCL domains (i.e., RAN, Edge, transport, and cloud domain CCLs). Using a hierarchical collaboration approach, cross-CCL collaboration provides CCLs inter-working executed at multiple abstraction levels. Lower-level CCLs provide abstracted information to the higher-level E2E CCL, which aggregate such inputs and generate a broader abstract view. The E2E CCL may also manage the subordinate CCLs, e.g., by sending actions/recommendations to the decision stage, such as migration or reroute, for domain-specific execution.
Intent and Policy Management	Once the collaboration model decides the participating CCLs and their scope, cross-CCL collaboration is ready to get started by policies or intents. A policy [13] defines a set of actions that an external entity requires a CCL to perform when a given set of conditions are met (formulated as: IF condition *A* is met, THEN take action *B*). An intent [14] is a customer-facing and technology-agnostic trigger that delegates to the management component the exploration of different alternatives to find an optimal solution.
Modularity	To provide cross-domain CCL collaboration and interoperability, industry standardized Application Programming Interfaces (APIs) can be used. E2E CCL, for instance, can interact with CCLs from RAN/Edge/transport/cloud domains via RESTful HTTP-based APIs (e.g., ETSI SOL005 APIs [15]) or using IETF-defined APIs from the Application-Layer Traffic Optimization (ALTO) WG [16] and/or Abstraction and Control of Traffic Engineered Networks (ACTN) [17]. Eventually, for communication across different administrative domains, Service Activation & Configuration [18] and Interlude [19] APIs, defined in TM Forum and MEF, respectively, are potential interface candidates. Other

	existing interfaces developed in open-source research projects (e.g., 5G-Transformer [20] and 5GEx [21]) can be also considered.
ML-assisted CCL	This allows autonomous management using machine intelligence techniques to unveil knowledge/insights of network services. More specifically, ML methods are leveraged to output the parameters used by the CCL system to adjust the network configuration continuously, as needed or when scheduled. For example, ML models can generate complex metrics on edge domain (QoE metrics) or in-depth analysis on transport domain (Root Cause Analysis).
CCL Workflow	Domain-specific workflows are used to specify the sequence of activities performed by each CCL to complete the collaboration and reach the desired target (e.g., automated healing, service assurance, resource allocation, or service/network performance optimization).

4 Use Case: ML-Assisted Multi-domain Automated Healing

Automated healing (or self-healing) for E2E services relies on that each MD executes a CCL for self-healing locally, as well as proper cross-domain collaboration. Without adequate cooperation, the self-healing achieved by local CCLs may be hazarded in cross-CL operations. Such collaborative approach together with advanced in-built ML capabilities contribute to the realization of the E2E self-healing task.

For the case study at hand, we consider the architecture shown in Fig. 3 composed of four management domains: Cloud, Transport, Edge, and RAN. Each domain executes a CCL for local self-healing tasks autonomously, whereas E2E CCL interconnects with other CCLs for target service assurance based on the E2E holistic view spanning multiple domains. Interactions among CCLs can take place at any of the four main stages: Monitoring, Analysis, Decision, or Execution. Communications can be bidirectional, either bottom-up driven by the lower-level CCLs or top-down triggered by the E2E CCL.

We focus on cross-domain collaboration and mainly on two key aspects related to the use of AI/ML to deliver self-healing functionality: *(i)* data collection and pre-processing at the Edge MD for service QoE estimation, and *(ii)* flow-level feature analysis for RCA in the Transport MD. These functions are detailed next in Sects. 5 and 6 , respectively.

As the target E2E service, we consider DASH video streaming spanning from a cloud domain to mobile network customers. For next-generation, carrier-grade levels, service assurance requires constant monitoring and rapid corrective actions from

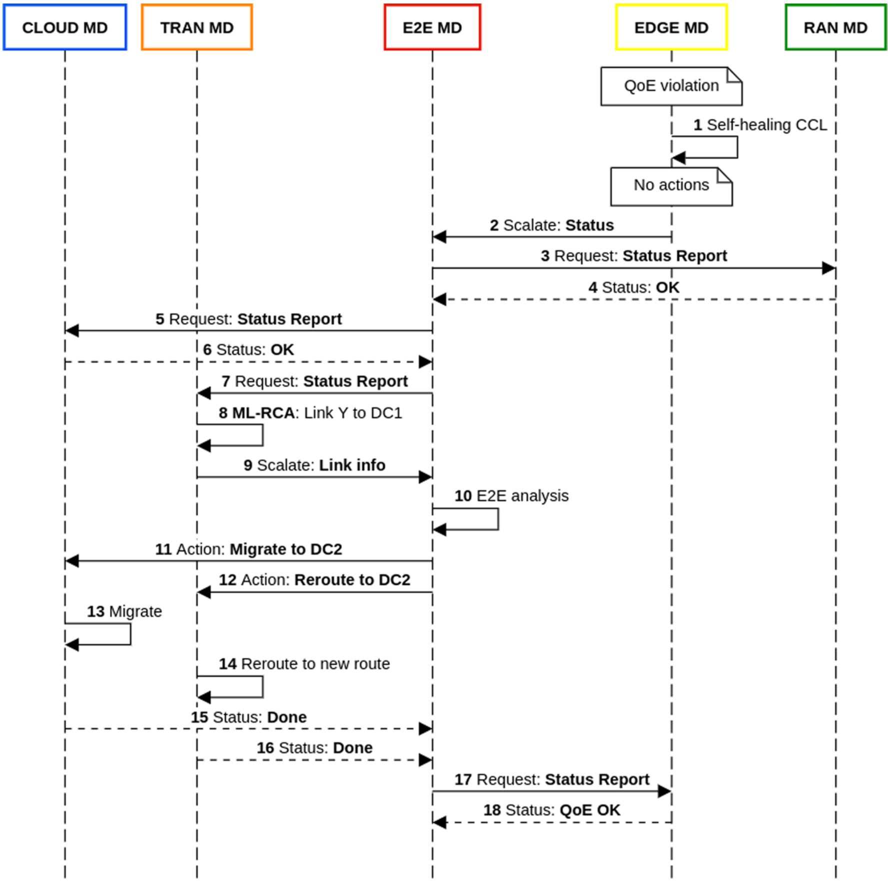


Fig. 4 Workflow of cross-CCL collaborative self-healing

the management systems when the service faces a loss of performance as perceived by the end-users, i.e. in terms of video QoE degradation.

A sequence diagram of the study case is presented in Fig 4 which illustrates how cross-CL collaboration and AI/ML are used to support the self-healing objectives. As preconditions, we define that the E2E service runs according to the predefined agreements (e.g., throughput, latency, coverage, video QoE) and KPIs are being monitored per each CCL. It is worth highlighting architecture is not limited to the workflow presented in Fig 4.

Edge monitoring is based on a video QoE probe agent that periodically collects network-level information used as inputs to estimate users' QoE KPIs. We consider a scenario that starts with increased packet loss in the transport domain and the Edge MD detecting a QoE violation (trigger point). The packet loss rate did not exceed the threshold specified in Transport MD policy; soon, its CCL was not triggered. Edge MD attempts a self-healing CCL to fix the issue, but no action is executed since the root cause does not originate in the Edge MD. Then, the issue is escalated to the

E2E MD that in turn requests from other MDs a status report to realize a complete diagnosis. RAN and Cloud MD return no performance degradation or faulty symptoms in their domains, i.e., status OK. Transport MD performs a root cause analysis based on ML techniques to search for potential issues not being detected through standard QoS monitoring routines. The outcome of the analysis identifies that link Y is undergoing performance issues that are likely causing the observed QoE degradation. The result is escalated to E2E MD which performs an E2E Analysis and identifies two actions: *(i)* migrate the video server to an alternative cloud domain (DC2), and *(ii)* create a new route to DC2,[1] to be executed by Cloud and Transport MD, respectively. Once executed, the E2E MD receives a notification from both domains. Finally, the E2E MD checks service status with Edge MD that reports that the E2E service works properly.

5 Edge-Based ML-Assisted QoE Probe CCL

This section details a Edge MD implementing stages of ZSM CCL assisted by ML methods. We refer to the implemented solution as *Edge QoE Probe* tailored to specific network region (e.g., edge premises), network topology, network conditions (e.g., congestion/bottleneck in specific backhaul links), end-user (e.g., individual user or group), and service (e.g., DASH video). The edge of the network (i.e., MEC domain) is an appealing location to analyze target end-users perceived video service quality (i.e., QoE) due to the proximity [22] and therefore smaller error-prone domain for QoE estimation subject to cross traffic interferences.

Figure 5 presents a high-level overview of the solution and target scenario. An Edge domain is shown alongside the R(AN) to represent the QoE probe scheme's execution as a Virtual Network Function (VNF) instance hosted by a candidate MEC server for the CCL functional stages.

For the instance of the MEC, the proposed *Edge QoE Probe* is stimulated by the "Network performance and QoE improvements" category use case [23]. Under this category, to optimize the backhaul/fronthaul network performance, a monitoring network function is responsible for providing the real-time network-level traffic information to the analytic network function, which role is to compute the traffic requirements if any degradation occurs at the backhaul/fronthaul end. Later, an optimization function will optimize the network according to traffic requirements calculated by the analytic function. Such a use case can also benefit by coordinating the Transport, Edge, and RAN domains. However, here we stated the proposed QoE probe scheme by keeping the harmony with CCL functional stages.

The *Edge QoE Probe* provides non-invasive, network-level encrypted traffic measurement methods based on passive probing running on the edge premises. The probe scheme periodically collects the bidirectional real-time network traffic information to predict users' QoE metrics. In short, such predicted QoE triggers the Edge

[1] Alternatively, the migration task could include the video DASH server at its new location DC2 sending HTTP redirect messages to active users among other well-known Service/VM migration techniques

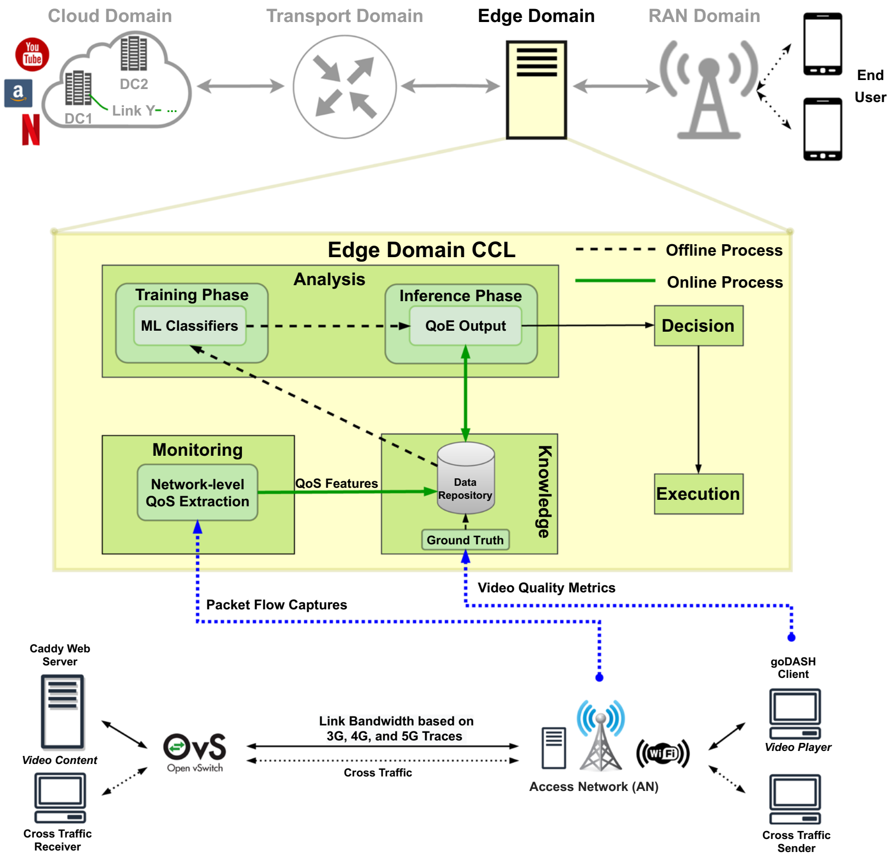


Fig. 5 Edge-based ML-assisted QoE probe CCL

domain's self-healing CCL hierarchy to identify the quality degradation and take proper action to overcome it.

The QoE probe scheme is responsible for (`Monitoring`) collecting and pre-processing data from bidirectional real-time encrypted DASH video traffic to yield the required features (`Knowledge`) that allow ML methods to infer (`Analysis`) the end-user perceived QoE. After evaluation of the intended QoE (e.g., violated or not), subsequent CCL stages (`Decision` and `Execution`) complete the work-flow through feasible actions (e.g., no action, reactive, proactive, predictive).

Supervised ML techniques have achieved enough maturity in recent years [24] to derive the complex relationship between encrypted network-level QoS features and user-perceived QoE metrics. Therefore, we follow a supervised ML technique to implement the `Analysis` and `Knowledge` stages of the CCL flows.

The *Edge QoE Probe* CCL can be divided into offline and online workflows to derive models and estimate the QoE.

Offline For network-level QoS features, we follow a lightweight, fine-grained network-level temporal QoS features extraction technique based on three-time windows, i.e., *current*, *trend*, and *session*, by observing bidirectional IP header information without computationally expensive (encrypted) video segment identification. We set a fine-grained 0.5-second length time window to compute different statistics of temporal QoS features. As far as we are aware, this is the shortest granularity by now for in-network QoE inferring of encrypted video streaming, allowing quick reactive performance diagnosis and resource allocation actions. However, as needed, an operator may opt for different QoE inferring and QoS extraction granularity (e.g., 1-, 5- and 10-second). As per the CCL chain, extracted QoS features stores in the `Knowledge` data repository. Moreover, for ground truth, we rely on specific users who can contribute using a particular device or player by continuously reporting the video streaming performance in objective QoE metrics format. The `Analysis` is responsible for training a supervised correlation model and (re-)evaluating the model performance by retrieving features and ground truth information from the data repository. Currently, the correlation model is based on a classification approach to predict a category/discrete value as QoE output. More details are presented in the evaluation Sect. 7.

Online Also known as *run-time phase*, the `Monitoring` functions fetch and extract the QoS features and stores them in the `Knowledge` data repository. The `Analysis` stage's inference functions periodically retrieve these QoS features and feed them into the trained model (classifier) to get end-user estimated QoE output.

6 ML-Based Transport Root Cause Analysis

This section presents the Transport MD design featuring ZSM CCL stages where ML methods assist the root cause analysis of under-performing (i.e., faulty) links.

Recalling the use case scenario presented in Sect. 4 (see sequence diagram in Fig. 4), after receiving the QoE violation alarm by the MEC MD, the E2E CCL is attempting to restore the service quality by interacting with the CCL in each MD. When requested by the E2E CCL, the Transport MD carries out a status check routines that include a RCA (a.k.a root cause localization) to detect potential network-level failures affecting a given service flow (e.g., identified by source and destination IPs).

In view of the growing popularity of DASH video traffic, the root cause localization process is very important, challenging and demanding. We present an RCA method shown in Figure 6 for the transport MD following the ZSM CCL chain. The method is based on a ML sandbox approach where a mirror (a.k.a network digital twin) of the transport network is created in an emulated environment with the same topology and link characteristics of the real network. The emulated environment is used to exercise a meaningful set of link issues (e.g., increased latency, packet loss, reduced bandwidth) all over the network using synthetic faulty link patterns or

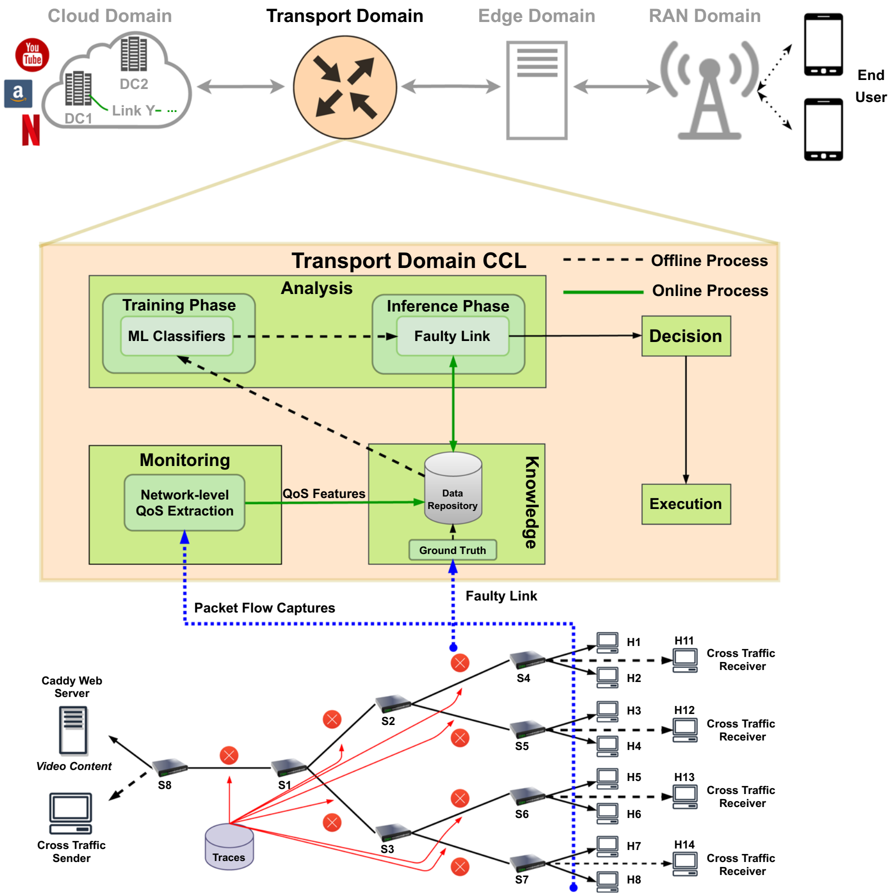


Fig. 6 MD-transport CCL leveraging ML-assisted RCA of under-performing network links

replaying observed patterns from real network traces. An ML model is then created to correlate end-user service QoE KPIs (e.g., DASH video objective QoE metrics) with network-level QoS patterns (used as features in the ML pipeline) to allow the run-time RCA to probabilistically point to the candidate network links (device interfaces) responsible for the service QoE degradation.

Similar to the MEC-based QoE Probe CCL, the RCA method can be divided into offline and online stages.

Offline The offline workflow consists of the following steps:

– Initiate the transport network digital twin, including source and destination service endpoints (DASH video server and clients) along with randomized background traffic.

- Inject faulty behaviors in the transport network digital twin on all instrumented links in a sequenced and controlled manner.
- For each faulty transport network link event, (`Monitoring`) collect all network-level QoS metrics per network element interface and record the objective QoE metrics from the emulated end-user client applications.
- Store flow QoS features together with faulty link information (`Knowledge`) and train the ML classifier (`Analysis`) to recognize transport link degradation issues.

Online The online stages of the Transport MD CCL for root cause localization runs as follows:

- Flow-level QoS features are extracted (`Monitoring`) and stored (`Knowledge`).
- The inference phase (`Analysis`) of the ML classifier use the QoS features to identify potential links as candidate root cause of E2E service quality degradation.

7 Experimental Evaluation

The experimental evaluation focuses on the two ML-assisted capabilities in the workflow of the cross-CCL use case (Fig. 4). Firstly, DASH QoE estimation and violation at the Edge domain (Experiment 1) to trigger the cross-CCL collaboration. Secondly, the root cause diagnosis at the Transport MD (Experiment 2) triggered by the E2E CCL and reporting the outcome pointing to candidate link issues.

The testbed in each experiment is different, common emulation tools and experimental practices are used, including network traces, dataset information and ML classification model performance metrics as described next:

Tools *Mininet-WiFi* [25]—high-fidelity network emulator with wireless extensions, *goDASH* [26]— lightweight DASH video player featuring objective QoE KPIs per segmet, *Caddy*—Web server hosting DASH video content, *D-ITG*—background traffic generator, *Linux TC*— traffic controller in the Linux kernel, and *Tcpdump*— passive network traffic sniffer.

Traces	*3G*[2], *4G*[3], and *5G*[4] cellular network traces' only downlink bandwidth parameter - set different link conditions through Linux TC to emulate realistic traffic patterns. The amount of total UDP cross-traffic sent by D-ITG was set to approximately 20% of the average cellular network traces bandwidth.
DASH videos	*Tears of Steel* and *Sintel* DASH video content—sourced from a publicly available [27] 4K DASH video files encoded using H.264/AVC and featuring eight resolutions across thirteen representation rates.
ML Library and Model Performance Metrics	Python-based *scikit-learn*[5] library implementation for the ML classification tasks and performance evaluation based on *Accuracy*, *Precision*, *Recall*, and *F1-score*.

7.1 Experiment 1: Edge Domain

7.1.1 Setup

The topology comprises one Access Point (AP) (i.e., Access Network) and one Open vSwitch (OvS) as shown in Figure 5. One DASH client and the D-ITG cross-traffic sender are connected to the same AP. On the opposite side, the DASH video server and the D-ITG cross-traffic receiver are connected to the OvS instance. Different network conditions are emulated dynamic Linux TC reconfiguration on the virtual interfaces between the AP and OvS following the downlink bandwidth patters from 3G, 4G, and 5G cellular network traces. With D-ITG, three concurrent flows of UDP traffic were sent from one sender point to another receiver point alongside the video traffic. On the server-side, we use 4-s video segments (*Tears of Steel*) for a 2 min long session. In the goDASH client, we explore six Adaptive Bit-rate Streaming (ABS) algorithms (e.g., Conventional, Exponential, BBA, Logistic, Arbiter, and Elastic), two transport protocols (e.g., HTTPS and QUIC). Each experiment, i.e., video streaming session with the different setup combinations was repeated five times. Network-level QoS features are extracted by processing the raw network traffic captured at the AP interface using Tcpdump.

[2] http://skulddata.cs.umass.edu/traces/mmsys/2013/pathbandwidth

[3] https://www.ucc.ie/en/misl/research/datasets/ivid_4g_lte_dataset

[4] https://github.com/uccmisl/5Gdataset

[5] https://scikit-learn.org/stable/

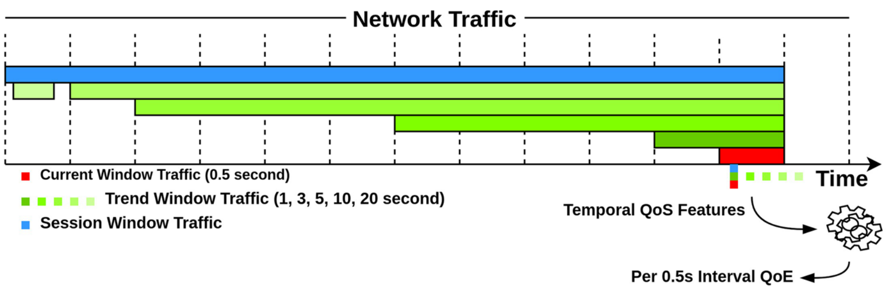


Fig. 7 Network-level QoS features extraction technique

7.1.2 Network-level Temporal QoS Features

A Python-based program was used to analyze and calculate various network-level QoS KPI statistics using packet header information. For real-time applicability, the core idea is to estimate QoE for short time windows based on temporal network-level QoS features calculated from bidirectional network traffic in a sliding window fashion, as shown in Fig. 7. Each short interval was defined as a *current* time window. Past windows were defined as two-way, most recent windows or *trend* window and past all windows or *session* window. We use half-second interval as a *current* time window length. For the *trend* window, the time windows' length was considered 1, 3, 5, 10, and 20 s. Note that the *trend* window contains traffic information also from the current window. The *session* window covers all the previous time window's traffic information from the start of the video stream to till current time window. For each type of time window, network layers QoS features are calculated for uplink and downlink directions for HTTPS and QUIC connections. More specifically, four basic network-level QoS KPI are taken into account: *throughput, number of packets, packet sizes, and packet inter-arrival time*. Overall statistics (e.g., total, min, max, median, std dev) are also computed for each QoS KPI. Thus, for real-time QoE metrics estimation, we extract a total of 252 QoS features for HTTPS connection and 168 QoS features for QUIC-based sessions.

7.1.3 Application-level QoE Metrics as Ground Truth

The ground truth (e.g., objective QoE metrics) was obtained for each experiment from the client-side as a prerequisite for training the supervised ML model. The goDASH client-side player outputs a log file of streaming content. featuring a rich set of information (e.g., arrival and delivery time, bit-rate, buffer level, stall duration, delivery rate, actual rate, resolution, five QoE models output) for each video segment. Since a video streaming ground truth was needed with a time sequence for real-time QoE prediction, we customized the goDASH provided log information by subdividing the video streaming into time sequences. We consider 0.5-s constant time length's QoE metrics (e.g., playback status, stall event, bit-rate, and resolution) information based on per segment arrival information. By default, goDASH uses two segments for the initial buffer threshold to start playback; thus, by measuring the first two segments' arrival time, we interpret whether the video play has started.

7.1.4 Supervised ML-based QoE Estimation

We work with seven supervised ML algorithms, namely, Logistic Regression (LR), K-Nearest Neighbors (KNN), Gaussian Naive Bayes (GNB), Support Vector Machine (SVM), Decision Tree (DT), Random Forest (RF), and Multi-layer Perceptron Classifier (MLP), all of them available in the scikit-learn library.

7.1.5 Target QoE Metrics

For real-time metrics, we emphasize on the displayed video quality, i.e., resolution and bit-rate metrics as the target output. Other QoE are less frequently (e.g., stall events) or hardly measurable (e.g., resolution switches).

- *Resolution (R)* To estimate real-time resolution metrics, we use the following classification method based on three classes: Low Definition (180p and 216p), Standard Definition (288p, 360p and 414p), and High Definition (720p and 1080p).
- *Bit-rate (B)* To estimate real-time bit-rate metrics, we classified bit-rate into three classes: Low (less than 700 Kbps), Medium (above 700 Kbps and below 2500 Kbps), and High (above 2500 Kbps).

7.1.6 ML Datasets

We use a rich combination of traces to generate the dataset, i.e., three cellular network technologies, for different user mobility patterns to dynamically adapt the wireless link conditions, for two transport protocols and six state-of-art ABS algorithms. The final dataset was separate for HTTPS **(H)** and QUIC **(Q)** transport concerning per time interval temporal QoS metrics as a feature and QoE metrics (e.g., resolution and bit-rate) as ground truth to build the ML models. This approach allows for real-time prediction for any video session. Specifically, it would be feasible to forecast the resolution and bit-rate outcomes for each current time interval. The dataset contains a total of 105613 entries over HTTPS and 100757 entries over QUIC for each 0.5 second time interval of video playback.

7.1.7 Model Benchmark

We benchmark seven ML algorithms using the entire dataset to train the models and select the best algorithm based on cross-validation results that suit our real-time QoE metrics prediction. The hyper-parameters were chosen for each algorithm using an exhaustive grid search[6] that selected the best parameters to maximize the scores (e.g., average accuracy) and avoid over-fitting. Each multi-class classifiers'

[6] https://scikit-learn.org/stable/modules/generated/sklearn. model_selection.GridSearchCV.html

Table 2 Benchmarking of seven supervised ML models

Model	Target	Dataset	Accuracy (%)	Training time (min)
LR	R	H	67 (± 0.91)	0.07 (± 0.002)
		Q	65 (± 1.43)	0.06 (± 0.003)
	B	H	67 (± 0.68)	0.08 (± 0.002)
		Q	64 (± 1.11)	0.06 (± 0.004)
KNN	R	H	79 (± 1.02)	0.09 (± 0.001)
		Q	82 (± 1.00)	0.05 (± 0.001)
	B	H	78 (± 0.63)	0.09 (± 0.001)
		Q	82 (± 0.78)	0.05 (± 0.001)
GNB	R	H	56 (± 2.57)	0.008 (± 0.00009)
		Q	62 (± 0.88)	0.005 (± 0.0002)
	B	H	55 (± 1.02)	0.008 (± 0.00005)
		Q	56 (± 0.97)	0.005 (± 0.0001)
SVM	R	H	64 (± 0.91)	39.01 (± 0.239)
		Q	57 (± 0.52)	27.62 (± 0.240)
	B	H	64 (± 0.73)	40.66 (± 0.137)
		Q	58 (± 0.53)	26.56 (± 0.132)
DT	R	H	88 (± 1.03)	0.20 (± 0.004)
		Q	92 (± 0.46)	0.18 (± 0.007)
	B	H	88 (± 0.80)	0.19 (± 0.003)
		Q	94 (± 0.98)	0.16 (± 0.005)
RF	R	H	91 (± 0.28)	3.41 (± 0.014)
		Q	96 (± 0.54)	3.36 (± 0.023)
	B	H	91 (± 0.44)	1.34 (± 0.022)
		Q	96 (± 0.55)	3.14 (± 0.020)
MLP	R	H	68 (± 5.19)	2.16 (± 0.020)
		Q	64 (± 7.33)	2.62 (± 0.027)
	B	H	72 (± 1.16)	2.93 (± 0.031)
		Q	69 (± 5.79)	2.51 (± 0.044)

performance was evaluated on a system with 10-core, 2.20 GHz CPUs (Intel Xeon(R) Silver 4114 CPU) processor and 64 GB of RAM.

Table 2 presents each multi-class classifier's 5-fold cross-validation average accuracy and training time. We note that the GNB classifier achieved the least amount of time for training but provided the worst accuracy. SVM classifiers took the highest amount of training time and exhibited second-worst accuracy. MLP showed the highest variability in the case of accuracy. In general, out of the seven classifier's, DT and RF classifiers achieved higher accuracy, nearly 90% for the HTTPS dataset and above 90% for the QUIC dataset. Though RF took slightly high training time than DT, we assume it is reasonable for training. However, RF achieved approximately 3% of absolute accuracy improvement compared to the DT. Based on the obtained results, we may conclude that the RF classifier was the most appropriate one. For the remainder of the evaluation, we use the RF classifier with the same parameter tuning.

7.1.8 Feature Importance

We now analyze different feature influences in terms of resolution and bit-rate predictions. For this purpose, we divide the full feature (HTTPS-252 features, QUIC-168 features) of the dataset into a total of six feature subsets in two categories. The first category contains three subsets of features from the *current* window (HTTPS-36 features, QUIC-24 features), *trend* window (HTTPS-180 features, QUIC-120 features), and *session* window (HTTPS-36 features, QUIC-24 features). In these subsets, all the features computed in each particulate window are considered. For the second category, we leverage the feature relative importance score to select the top 15 features from each window using a multi-class RF classifier with a fixed seed for controlling randomness.

We split each feature subsets into the ratio of 80% (training) and 20% (testing). We use the RF classifier to train a model with the training portion of data and evaluated trained models' weighted average precision, recall, and F1-score based on testing data from each of these six feature subsets. The model prediction performance reports for each feature subsets are given in Table 3. The analysis indicates that the top 15 feature subsets provided similar performance as the full feature subsets provided. Due to space constraints, we do not present the names of all top 15 QoS features for each case but share the main insights being the uplink and downlink packet inter-arrival time QoS KPI for the *current* window, the uplink packet size QoS KPI for the *trend* window, and the average downlink throughput QoS KPI for the *session* window hold high relative importance for both prediction metrics.

We observe that *current* and *trend* window features are not adequate to predict QoE metrics according to F1-score. For the *trend* window, we found the larger time duration interval's features (e.g., 20 and 10 seconds) had the most influence on prediction. The model with only the *session* window's features yielded the highest accuracy (98% over HTTPS and 99% over QUIC dataset) for both both QoE metrics.

7.2 Experiment 2: Transport Domain

7.2.1 Setup

As shown in Figure 6, the Mininet-WiFi emulated transport network topology consist of eight OvS instances (S1 — S8) and eight goDASH clients (H1 — H8) streaming from the server (Caddy) hosting DASH video (*Sintel*) encoded in eight different resolutions. Moreover, D-ITG was used to send cross-traffic to four different clients connected to four different sink locations such as (H11 — S4), (H12 — S5), (H13 — S6), (H14 — S7). During each experiment, we degraded a particular link's capacity (i.e., faulty link) by applying the 5G trace characteristics to the target link and keeping all other links with default parameters. We repeat the process for each link and collect network traces from the client-side switch interface and the QoE logs by goDASH.

Table 3 Different feature subsets' model performance

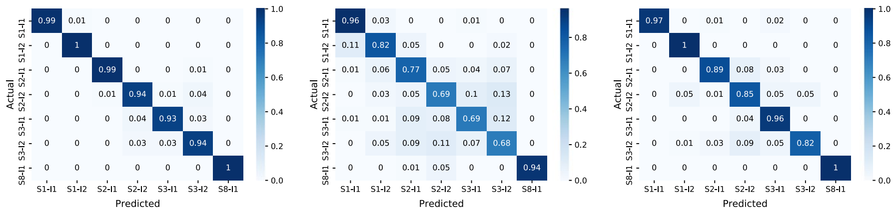
Feature	Target	Dataset	Precision (%)	Recall (%)	F1-score (%)
Current_{full}	R	H	67	62	61
		Q	71	66	66
	B	H	68	61	60
		Q	72	65	65
Current_{top-15}	R	H	67	63	61
		Q	71	67	66
	B	H	68	61	60
		Q	73	66	66
Trend_{full}	R	H	90	88	88
		Q	91	90	90
	B	H	89	87	87
		Q	92	90	90
Trend_{top-15}	R	H	88	87	87
		Q	91	89	89
	B	H	88	86	86
		Q	91	89	89
Session_{full}	R	H	98	98	98
		Q	99	99	99
	B	H	98	98	98
		Q	99	99	99
Session_{top-15}	R	H	98	98	98
		Q	99	99	99
	B	H	98	98	98
		Q	99	99	99

7.2.2 Network-level QoS Features and Faulty Link as Ground Truth

Again, a Python-based script was used to extract network-level QoS features (e.g., uplink RTT, downlink throughput, and packets) per video segment. Knowledge of the faulty link information was used as ground truth as well as the target variable (Y). During each experiment, all goDASH clients experience different QoE when we force less QoS at different links.

7.2.3 Supervised ML-based Root Cause Localization

We explore three supervised algorithms: Random Forest, Artificial Neural Network (ANN), and K-Nearest Neighbors. We analyze the performance of each model on the dataset for predicting the link causing the observed QoE issues. We input the model QoS features to predict the target class in our case (faulty link) and evaluate each model's performance. The confusion matrix of three classifiers is shown in Fig. 8, with QoS features (RTTs, throughput, packets) used as model inputs and



(a) Random Forest (b) K-Nearest Neighbors (c) Artificial Neural Network

Fig. 8 Confusion matrices for the ML-based RCA of faulty links

Table 4 Performance of ANN, RF, KNN classifier

Classifier	Precision	Recall	F1-score
ANN	0.93	0.93	0.93
RF	0.97	0.97	0.97
KNN	0.80	0.79	0.79

the faulty link as the root cause output of the model. The diagonal elements in the confusion matrices represent the true label, while the classifier mislabels are the off-diagonal elements. The labels on the axis present the switch interfaces (e.g., S8-I1) where the link performance degradation was applied through the bandwidth limitations introduced by the selected traces.

We observe RF correctly classified most of the interfaces with accuracy above 94% (see Fig. 8a). KNN performs slightly worse than RF and ANN. We observe 70% accuracy for three interfaces while the rest of the faulty interfaces are classified with an accuracy higher than 80% (see Fig. 8b). Finally, with ANN faulty links are correctly identified with an average accuracy of more than 85–86% (see Fig. 8c).

As shown in the model performance results in Table 4, RF exhibits the best performance, i.e., highest Precision, Recall, and F1-score (97% each). A high recall value indicates that RF classified most of the positive faulty links as RCA correctly. For ANN and KNN, the precision remains 93% and 80% respectively.

8 Related Work

8.1 Automated Healing Networks

Self-healing is the capacity of a network to automatically detect and mitigate system failures [28]. Consequently, different standardization bodies and projects have devoted remarkable efforts to design, develop, and implement self-healing Networks. ETSI ZSM [6], ETS ITC INT [29], and ITU-T [30] have proposed different fully automated ML-driven reference architectures supported by cross-domain integration fabric, hierarchical abstract modules, and modular pipelines, respectively.

Regarding collaborative projects, Selfnet[7] developed one of the first autonomous frameworks to provide heterogeneous networks with self-organizing capabilities. Cognet[8] also laid the foundations of the design and implementation of a network autonomous management framework with ML to adequately respond to the dynamic changes in a network. Aside from that, 5GZORRO[9], 5GROWTH [10], and INSPIRE-5Gplus [11] are working on the automation of E2E network services in vertical industries through the introduction of AI-powered solutions in self-organizing networks, including ML-based trust and security mechanisms.

8.2 CCL & Multi-domain CCL

Standards Developing Organisations (SDOs), as well as academic research work, are making progress on CCL mechanisms to pursue autonomous and automated management [3, 31–34]. ETSI ENI (Experiential Network Intelligence) ISG [31] defines a cognitive network management architecture using CCL AI mechanisms based on metadata-driven and context-aware policies to enhance the operator experience. TMF's Zero-touch Orchestration, Operations, and Management (ZOOM) project [32] is mainly guided to define a new management architecture with CCL E2E management, open and dynamic APIs, and near real-time and zero-touch. Authors in [33] propose an optimal performance CCL solution for network slicing where traffic forecasting information is ingested to maximize the number of granted network slice requests with their respective SLAs. [34] proposes a method to extend the CCL-based ZSM architecture to include service monitoring as a parallel process to service deployment. Multi-domain CCL is explored in [3], authors present a generic framework (with key elements, e.g., APIs, governance models, intent, and policies management) to enable cross-domain collaboration between CCLs for autonomous E2E service and management in 5G. A demonstration of our first CCL prototype [35] provides some additional implementation details in a single-domain scenario and with limited ML capabilities.

8.3 ML for QoE Estimation

In literature, supervised ML was used either per-session or real-time for encrypted traffic in-network DASH service QoE measurements. Authors in [36, 37] used entire video session generated network-level QoS features to make a QoS-to-QoE correlation model to estimate per video session QoE metrics. In contrast, for real-time, network-level QoS features generated on a specific time window (e.g., fine-grained granularity) in the video session were used to make a correlation model to estimate QoE metrics that time window. Authors in [38, 39] showed QoE metrics estimation with 10-second granularity. Similarly, authors in

[7] https://selfnet-5g.eu/

[8] https://5g-ppp.eu/cognet/

[9] https://5gzorro.eu/

[10] https://5growth.eu/

[11] https://inspire-5gplus.eu/

[40] showed QoE estimation with 5-second and in [41, 42] showed with 1-second granularity. Moreover, works [43, 44] depicted the QoE estimation evaluation concentrating on edge/fog location. Authors in [43] presented QoE estimation based on Deep Packet Inspection (DPI) and, in [44] introduced temporal network-level feature extraction, but the prediction granularity was vague. We adopt an approach similar to the works [41, 42] for temporal network-level QoS features extraction from encrypted traffic with 0.5-second prediction granularity at the edge location. Since a stall event or quality change is rarely observed in the shortest granularity time window, this work emphasizes displayed video quality such as resolution and bit-rate QoE metrics. However, our generated real-time datasets show the possibility of estimating per-session QoE metrics (e.g., stall ratio, no. of quality changes, average Mean Opinion Score (MOS), average resolution) by aggregating real-time QoS features for the whole session. Besides, this work is based on an emulation-based DASH video service, which allows the use of different adaptation logic (i.e., ABS algorithm) for video segment selection instead of a particular streaming service (e.g., YouTube, Netflix) adaptation logic. In previous work [45, 46], we explored ML for video service assurance through QoE estimation on a per-segment basis a smaller QoS feature set (e.g., uplink RTT, downlink throughput and packets) from unencrypted testbed traffic, whereas in [47] we showed ML classifiers of QoE prediction in a small scale from encrypted traffic packet length distribution and in [48] we focused on the QoE evaluation concerning HTTPS and QUIC transport protocol. All these previous efforts contributed to the development of DASH QoE performance evaluation methods, the emulation platform's implementation, and the reproducible framework used in the experimental evaluation of this work.

8.4 ML for RCA

In [49], authors used Neural Networks (NNs) with *syslog messages*. These messages were converted into numerical features and then forwarded as input to the ML model. The output of the model is then interpreted with decision-making rules. Similarly, in another work, authors used RNN to detect faulty nodes [50]. RNNs are a NN class where connections between neurons represent one cycle or more, allowing internal feedback to their inputs. Efforts have been made in [51, 52] using K-mean clustering to detect a network's fault. In k-mean clustering, n observations are partitioned into k clusters where each observation belongs to clusters with the highest similarities. In another work, authors analyzed YouTube traces of one month to detect QoE relevant anomalies in Content Delivery Network (CDN) distributed services [53]. In their work they use use statistical analysis methodologies to unveil the root causes behind automatically detected problems linked to the dynamics of CDNs' server selection strategies. Recent research made on this problem is by using Directed Acyclic Graph (DAG). DAG shows cause and effect relationships between observation and unobservable events. Therefore, when a set of symptoms is observed, the causes can be easily determined [54]. In DAG, nodes represent network variables such as elements, events or faults, whereas the edges denote the direct causality between the connected variables.

9 Conclusions and Future Work

Evolving networks beyond 5G seek for advancements in support of fully automated E2E network services that span multiple technology and administrative domains. This work revolves on a cross-domain collaboration framework based on the ETSI ZSM reference architecture towards the realization of zero-touch E2E network and service management. We show that ML/AI-based CCL automation is expected to play a key role in making ZSM a reality. More specifically, we present machine intelligence capabilities to estimate the intended service QoE of DASH video services at the edge domain and a CCL that builds upon ML-based diagnosis of faulty transport network links behind the inferred QoE degradation event. The ML techniques are based on network-level QoS monitoring information and an online training phase using a high-fidelity emulation environment as a ML sandbox. The presented experiments mainly serve as proof of concepts. Efforts are undergoing to tie together all pieces in a single experimental setup and explore current limitations, including scalability, adaptability to network topology changes, among others. Following standardized automated NFV performance benchmarking practices [55] is also on our roadmap.

We believe that underpinning concepts related to network digital twins for ML-assisted operations will increasingly embrace advances in the field of network automation. Further future work includes improvements around the ML pipelines such as supervised ML model reevaluation, adequate ground truth collection, and model generalization which were left out of the discussion in this work. Moreover, the performance evaluation of the entire hierarchical CCL mechanism in terms of escalation and delegation time for CCL-cross collaboration will be considered in future work.

Acknowledgements This research was supported by the Innovation Center, Ericsson S.A., Brazil, grant UNI.67. The views expressed are solely those of the authors and do not necessarily represent Ericsson's official standpoint.

References

1. Saraiva, N., Lachos, D., Rosa, R., Santos, M., Rothenberg, C.: Network service orchestration: a survey. Comput. Commun. **142–143**, 69–94 (2019)
2. Rosa, R.V., Silva Santos, M.A., Rothenberg, C.E.: Md2-nfv: the case for multi-domain distributed network functions virtualization. In: 2015 International Conference and Workshops on Networked Systems (NetSys), pp. 1–5 (2015)
3. Xie, M., Gomes, P.H., Harmatos, J., Ordonez-Lucena, J.: Collaborated closed loops for autonomous end-to-end service management in 5G. In: 2020 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), pp. 64–70 (2020)

4. Benzaid, C., Taleb, T.: AI-driven zero touch network and service management in 5G and beyond: challenges and research directions. IEEE Netw. **34**(2), 186–194 (2020)
5. ETSI: Zero-touch orchestration, operations and management (ZOOM). https://www.etsi.org/technologies/zero-touch-network-service-management
6. ETSI: ETSI GS ZSM 002: zero-touch network and service management (ZSM); reference architecture. Tech. Rep., 08 (2019)
7. Lachos Perez, D.A., Rothenberg, C.E.: MUDED: integrating networks with applications through multi-domain exposure and discovery mechanisms. In: VI IEEE NFV-SDN (2020)
8. Xie, M., Gomes, P.H., Harmatos, J., Ordonez-Lucena, J.: Collaborated closed loops for autonomous end-to-end service management in 5G. In: 2020 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), pp. 64–70. IEEE (2020)
9. Boutaba, R., Salahuddin, M.A., Limam, N., Ayoubi, S., Shahriar, N., Estrada-Solano, F., Caicedo, O.M.: A comprehensive survey on machine learning for networking: evolution, applications and research opportunities. J. Internet Serv. Appl. **9**(1), 16 (2018). https://doi.org/10.1186/s13174-018-0087-2
10. Pan, J., Cai, L., Yan, S., Shen, X.S.: Network for ai and ai for network: challenges and opportunities for learning-oriented networks. IEEE Netw **35**, 270–277 (2021)
11. ETSI: GS ZSM 001—V1.1.1—Zero-Touch Network and Service Management (ZSM); Requirements Based on Documented Scenarios. Group Specification (gs), 02 (2019)
12. ETSI. ETSI GS ZSM 009-1 V1.1.1: Zero-Touch Network and Service Management (ZSM); Closed-Loop Automation; Part 1: Enablers. Group Specification (gs), 06 (2021)
13. Strassner, J., Strassner, J.S.: Policy-Based Network Management: Solutions for the Next Generation. Morgan Kaufmann, Boston (2004)
14. Behringer, M.H., Pritikin, M., Bjarnason, S., Clemm, A., Carpenter, B.E., Jiang, S., Ciavaglia, L.: Autonomic Networking: Definitions and Design Goals. RFC 7575 (June 2015)
15. ETSI: ETSI GS NFV-SOL 005: network functions virtualisation (NFV) release 2; protocols and data models; RESTful protocols specification for the Os-Ma-nfvo reference point. Tech. Rep., 01 (2020)
16. Alimi, R., Penno, R., Yang, Y., Kiesel, S., Previdi, S., Roome, W., Shalunov, S., Woundy, R.: Application-layer traffic optimization (ALTO) protocol. RFC 7285 (2014)
17. Ceccarelli, D., Lee, Y.: Framework for abstraction and control of TE networks (ACTN). Internet Eng Task Force (IETF), RFC8453 (2018)
18. Forum, T.: TMF640 service activation and configuration API user guide v4.0.1. Tech. Rep., 07 (2020)
19. MEF: MEF 55 LSO IRP: LSO interlude. Tech. Rep., 06 (2020)
20. de la Oliva, A., Li, X., Costa-Perez, X., Bernardos, C.J., Bertin, P., Iovanna, P., Deiss, T., Mangues, J., Mourad, A., Casetti, C., Gonzalez, J.E., Azcorra, A.: 5G-TRANSFORMER: slicing and orchestrating transport networks for industry verticals. IEEE Commun. Mag. **56**(8), 78–84 (2018)
21. Bernardos, C.J., Gerö, B.P., Di Girolamo, M., Kern, A., Martini, B., Vaishnavi, I.: 5GEx: realising a Europe-wide multi-domain framework for software-defined infrastructures. Trans. Emerg. Telecommun. Technol. **27**(9), 1271–1280 (2016)
22. Dinh-Xuan, L., Seufert, M., Wamser, F., Tran-Gia, P.: Study on the accuracy of QoE monitoring for HTTP adaptive video streaming using VNF. In: 2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), pp. 999–1004 (2017)
23. Neal, A., Naughton, B., Chan, C., Sprecher, N., Abeta, S.: Mobile edge computing (mec); technical requirements. ETSI, Sophia Antipolis, France, White Paper no. DGS/MEC-002 (2016)
24. Vasilev, V., Leguay, J., Paris, S., Maggi, L., Debbah, M.: Predicting QoE factors with machine learning. In: 2018 IEEE International Conference on Communications (ICC), pp. 1–6, IEEE (2018)
25. Fontes, R.R., Afzal, S., Brito, S.H.B., Santos, M.A.S., Rothenberg, C.E.: Mininet-WiFi: emulating software-defined wireless networks. In: 2015 11th International Conference on Network and Service Management (CNSM), pp. 384–389. IEEE (2015)
26. Raca, D., Manifacier, M., Quinlan, J.J.: Godash—go accelerated has framework for rapid prototyping. In: 2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX), pp. 1–4. IEEE (2020)
27. Quinlan, J.J., Sreenan, C.J.: Multi-profile ultra high definition (UHD) AVC and HEVC 4K DASH datasets. In: Proceedings of the 9th ACM Multimedia Systems Conference, pp. 375–380 (2018)

28. Gallego-Madrid, J., Sanchez-Iborra, R., Ruiz, P.M., Skarmeta, A.F.: Machine learning-based zero-touch network and service management: a survey. Digit. Commun. Netw. (2021). https://doi.org/10.1016/j.dcan.2021.09.001
29. ETSI: Technical Committee (TC) Core Network and Interoperability Testing (INT). https://www.etsi.org/committee/int (2018)
30. ITU: Focus group on machine learning for future networks including 5g. https://www.itu.int/en/ITU-T/ (2017)
31. ETSI: Experiential Networked Intelligence (ENI). https://www.etsi.org/technologies-clusters/technologies/experiential-networked-intelligence (2018)
32. Forum, T.: Zero-touch orchestration, operations and management (ZOOM). https://www.tmforum.org/collaboration/zoom-project/ (2016)
33. Sciancalepore, V., Samdanis, K., Costa-Perez, X., Bega, D., Gramaglia, M., Banchs, A.: Mobile traffic forecasting for maximizing 5G network slicing resource utilization. In: IEEE INFOCOM 2017—IEEE Conference on Computer Communications, pp. 1–9 (2017)
34. Saraiva de Sousa, N.F., Lachos Perez, D., Esteve Rothenberg, C., Henrique Gomes, P.: End-to-end service monitoring for zero-touch networks. J. ICT Standard. **9**, 91–112 (2021)
35. Rothenberg, C.E., Lachos Perez, D.A., Saraiva de Sousa, N.F., Rosa, R.V., Mustafa, R.U., Islam, M.T., Gomes, P.H.: Intent-based control loop for dash video service assurance using ml-based edge qoe estimation. In: 2020 6th IEEE Conference on Network Softwarization (NetSoft), pp. 353–355 (2020)
36. Dimopoulos, G., Leontiadis, I., Barlet-Ros, P., Papagiannaki, K.: Measuring video QoE from encrypted traffic. In: Proceedings of the 2016 Internet Measurement Conference, pp. 513–526 (2016)
37. Khokhar, M.J., Ehlinger, T., Barakat, C.: From network traffic measurements to qoe for internet video. In: 2019 IFIP Networking Conference (IFIP Networking), pp. 1–9. IEEE (2019)
38. Mazhar, M.H., Shafiq, Z.: Real-time video quality of experience monitoring for https and quic. In: IEEE INFOCOM 2018-IEEE Conference on Computer Communications, pp. 1331–1339. IEEE (2018)
39. Bronzino, F., Schmitt, P., Ayoubi, S., Martins, G., Teixeira, R., Feamster, N.: Inferring streaming video quality from encrypted traffic: practical models and deployment experience. In: Proc. ACM Meas. Anal. Comput. Syst., vol. 3 (2019)
40. Gutterman, C., Guo, K., Arora, S., Wang, X., Wu, L., Katz-Bassett, E., Zussman, G.: Requet: real-time qoe detection for encrypted youtube traffic. In: Proceedings of the 10th ACM Multimedia Systems Conference, MMSys'19, (New York, NY, USA), pp. 48–59, Association for Computing Machinery (2019)
41. Orsolic, I., Skorin-Kapov, L.: A framework for in-network QoE monitoring of encrypted video streaming. IEEE Access **8**, 74691–74706 (2020)
42. Wassermann, S., Seufert, M., Casas, P., Gang, L., Li, K.: ViCrypt to the rescue: real-time, machine-learning-driven video-QoE monitoring for encrypted streaming traffic. IEEE Trans. Netw. Serv. Manag. (2020). https://doi.org/10.1109/TNSM.2020.3036497
43. Ge, C., Wang, N.: Real-time qoe estimation of dash-based mobile video applications through edge computing. In: IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp. 766–771. IEEE (2018)
44. Zheng, H., Zhao, Y., Lu, X., Cao, R.: A mobile fog computing-assisted dash qoe prediction scheme. Wirel. Commun. Mob. Comput. (2018). https://doi.org/10.1155/2018/6283957
45. Ul Mustafa, R., Islam, M.T., Rothenberg, C., Ferlin, S., Raca, D., Quinlan, J.J.: DASH QoE performance evaluation framework with 5G datasets. In: In IEEE CNSM AnServApp Workshop (2020)
46. Ul Mustafa, R., Ferlin, S., Esteve Rothenberg, C., Raca, D., Quinlan, J.J.: A supervised machine learning approach for dash video qoe prediction in 5g networks. In: Proceedings of the 16th ACM Symposium on QoS and Security for Wireless and Mobile Networks. Q2SWinet '20, (New York, NY, USA), pp. 57–64. Association for Computing Machinery (2020)
47. Ul Mustafa, R., Moura, D., Rothenberg, C.E.: Machine learning approach to estimate video QoE of encrypted DASH traffic in 5G networks. In: In IEEE Singal Processing (SSP2021) Workshop (2021)
48. Islam, M., Rothenberg, C.: HAS based empirical qoe study over TCP and QUIC on diverse networks. In: Anais do VII Workshop Pré-IETF, (Porto Alegre, RS, Brasil), pp. 29–42. SBC (2020)
49. Zhaojun, G., Chao, W.: Statistic and analysis for host-based syslog. In: 2010 Second International Workshop on Education Technology and Computer Science, vol. 2, pp. 277–280. IEEE (2010)
50. Moustapha, A.I., Selmic, R.R.: Wireless sensor network modeling using modified recurrent neural networks: application to fault detection. IEEE Trans. Instrum. Meas. **57**(5), 981–988 (2008)
51. Qader, K., Adda, M., Al-kasassbeh, M.: Comparative analysis of clustering techniques in network traffic faults classification. Int. J. Innov. Res. Comput. Commun. Eng. **5**, 6552–6563 (2017)

52. Akbari, A., Beikmahdavi, N.: Cluster-based & cellular approach to fault detection & recovery in wireless sensor networks. In: 3rd Int. Conf. on Advanced Computer Theory & Eng. (ICACTE) (2010)
53. Casas, P., D'Alconzo, A., Fiadino, P., Bär, A., Finamore, A., Zseby, T.: When youtube does not work—analysis of qoe-relevant degradation in google cdn traffic. IEEE Trans. Netw. Serv. Manage. **11**(4), 441–457 (2014)
54. Tembo, S.R., Courant, J.-L., Vaton, S.: A 3-layered self-reconfigurable generic model for self-diagnosis of telecommunication networks. In: 2015 SAI Intelligent Systems Conference (IntelliSys), pp. 25–34. IEEE (2015)
55. Rosa, R.V., Bertoldo, C., Rothenberg, C.E.: Take your vnf to the gym: a testing framework for automated nfv performance benchmarking. IEEE Commun. Mag. **55**(9), 110–117 (2017)

Nathan Franklin Saraiva de Sousa has a BSc and an MSc in Computer Science from Federal University of Piaui (UFPI), in 2005 and 2015, respectively. He is currently a Ph.D. candidate in the Department of Computer Engineering and Industrial Automation, School of Electrical and Computer Engineering at the University of Campinas (UNICAMP). He worked on projects funded by Ericsson Research Brazil. Since 2010 he is a Computer Analyst at the Federal Institute of Piaui (IFPI). His research interests span SDN, NFV, policy-driven closed control loops, and Beyond 5G (B5G) networks management.

Md Tariqul Islam is currently enrolled in the Ph.D. program in the Department of Computer Engineering and Industrial Automation (DCA), School of Electrical and Computer Engineering (FEEC) at the University of Campinas (UNICAMP), Brazil. He is one of the active members of the INTRIG Lab. He obtained the MSc degree from UNICAMP, Brazil, in 2021 and the BSc degree from Mawlana Bhashani Science and Technology University (MBSTU), Bangladesh, in 2018. His research interests are SDN and NFV, Edge Computing, QoS and QoE of DASH Video, Machine Learning and Deep Learning, In-band Network Telemetry, Wireless and Cellular Network Protocol Stack.

Raza Ul Mustafa is a full-time Ph.D. scholar at the Department of Computer Engineering and Industrial Automation (DCA) School of Electrical and Computer Engineering (FEEC), UNICAMP, Brazil. He is researching to find QoS metrics to map QoE of video streaming in 5G networks. Previously he was a full-time Lecturer at the Department of Computer Science, COMSATS University Islamabad, Sahiwal Campus Pakistan. He taught third year students, focused primarily on "Automata Theory, Web Engineering, Machine learning, Python, C and C++. He has experience in various positions such as Data scientist, Machine learning engineer, Web-engineer (developer), Computer Science teacher.

Danny Alex Lachos Perez received his Ph.D. and M.Sc. degree in Computer Engineering from the University of Campinas (UNICAMP), Brazil, in 2021 and 2016, respectively. During this time, He was involved as a software and Internet researcher in different academy-industry collaboration projects related to networked systems. Danny has a particular interest in new and flexible network and application integration mechanisms in the context of multi-domain environments.

Christian Esteve Rothenberg is an Assistant Professor in the Faculty of Electrical & Computer Engineering (FEEC) at University of Campinas (UNICAMP), Brazil, where he received his Ph.D. and currently leads the Information & Networking Technologies Research & Innovation Group (INTRIG). His research activities span all layers of distributed systems and network architectures and are often carried in collaboration with industry, resulting in multiple open source projects in SDN and NFV among other scientific results.

Pedro Henrique Gomes has more than 15 years Researching, Enhancing, and Building wireless communications systems, from Wi-Fi-based local networks, RFID-based applications, wireless sensor networks, and IoT, to 4G and 5G mobile cellular networks. Currently working as a senior researcher at Ericsson Research, Brazil, engaged in the management of 5G networks and services. He is a delegate in the ETSI Zero-Touch Network & Service Management working group, contributing to the architecture definition especially with closed-loop automation and intent-driven enablers. Received Ph.D. (2019) and M.Sc. (2015) in electrical engineering from the University of Southern California, Los Angeles, USA, also M.Sc. (2011) in computer science and B.Sc. (2007) in computer engineering from the University of Campinas, Brazil.

Authors and Affiliations

Nathan Franklin Saraiva de Sousa[1] · **Md Tariqul Islam**[1] · **Raza Ul Mustafa**[1] · **Danny Alex Lachos Perez**[1] · **Christian Esteve Rothenberg**[1] · **Pedro Henrique Gomes**[2]

Md Tariqul Islam
tariqsaj@dca.fee.unicamp.br

Raza Ul Mustafa
razaul@dca.fee.unicamp.br

Danny Alex Lachos Perez
dlachosp@dca.fee.unicamp.br

Christian Esteve Rothenberg
chesteve@dca.fee.unicamp.br

Pedro Henrique Gomes
pedrohenriquegomes@gmail.com

[1] UNICAMP, Campinas, Brazil

[2] Ericsson Research, São Paulo, Brazil