An Exposed Closed-Loop Model for Customer-Driven Service Assurance Automation

Min Xie*, Foivos Michelinakis†, Thomas Dreibholz†, Joan S. Pujol-Roig‡, Sara Malacarne*,
Sayantini Majumdar§, Wint Yi Poe§, Ahmed M. Elmokashfi†
*Telenor Research, Telenor, Norway; †Simula Metropolitan Centre for Digital Engineering, Norway;
‡Samsung Electronics R&D Institute, UK; §Munich Research Center, Huawei Technologies

***Abstract*—Artificial Intelligence (AI) is widely applied in telecommunications to enable zero-touch automation in network operation and service management. Due to the high complexity, deploying advanced AI mechanisms is not always feasible inside the operator's network domains. Instead, via service exposures, it becomes possible for vertical customers to integrate their external AI solutions with the network and service management system to form a closed loop (CL) and contribute to the automation process. In this paper, we propose an exposed CL model based on service exposure and apply it to automate service assurance tasks like auto-scaling in a network function virtualization (NFV) system orchestrated by ETSI Open Source MANO (OSM). A testbed is built to validate the model. It collects monitoring data from the OSM monitoring module and external monitoring tools. Vertical customers drive and customize their AI solutions to aggregate these data sets and run analytics to detect and predict anomalies prepared for scaling. Preliminary analysis demonstrates the added values of customer-driven monitoring and analysis via the exposed CL.**

I. Introduction

The increasing complexity, flexibility and scale of mobile networks have driven the growing interest in automating the provisioning, operation and maintenance of network services. Closed-loop (CL) control is a typical way to automate network and service management, in which Artificial Intelligence (AI) and Machine Learning (ML) are a key enabler. As highlighted in [1], AI and ML are important to facilitate such automation through the lifecycle of network services, especially in a highly diverse end-to-end (E2E) network with numerous network slice instances. Since AI/ML mechanisms are heavily dependent on the richness of data to achieve high accuracy and reliability, the quality and quantity of data acquired in the network and exposed to AI/ML modules is critical.

Conventionally, the network data, collected by network operators (NOPs), is analyzed internally with built-in analytics functionality. With limited resources, the capability and performance of AI/ML inside NOP are too limited to meet the high requirements of diverse vertical customers. One option is to expose the network data to external customers and allow them to build customized AI/ML solutions to solve their specific problems. However, exposing network data, particularly user-related data, might pose significant risk in privacy and security. Proper models are developed to regulate the management of exposure capabilities, as in GSMA [2] and NGMN [3]. 3GPP defines Network Data Analytics Function (NWDAF) [4] allowing external customers to retrieve network data from 5G System via application functions and vice versa.

In [5], we proposed a service assurance (SA) framework with service exposure, which showed a high feasibility of exposing Monitoring and Analyzing services to external customers. In this paper, we apply the framework to a MAPE-K (monitor-analyze-plan-execute with knowledge) [6] CL model and propose a modified MAPE-K model with service exposure, called *exposed MAPE-K*. It exposes monitoring data to external customers, who, in turn, provide insight via external monitoring and analysis. A hybrid CL is therefore formed to boost the internal MAPE-K CL. The realization of this model involves multiple parties (internal and external) and thus requires for demarcating spheres of influence. Currently, each party can run monitoring independently, with their own metrics, granularity and coverage, which brings up two challenges.

First, the multiple data sets generated by multi-party monitoring contain information of different aspects. On one hand, they complement each other and produce *diversity gain*, *e.g.*, in enhancing visibility or improving root cause analysis (RCA). In [7], the necessity of multi-level monitoring was verified for classifying the root causes of performance bottlenecks in cloudified mobile networks. On other hand, *new data aggregation and fusion mechanisms* need to be developed to achieve such a diversity gain. Second, multi-party monitoring increases the cost of monitoring, especially when multiple parties monitor the same entity and cause a redundancy in the monitoring data sets. Then it is significant to select the *right* monitoring party whose data is valuable of high information but with a low cost. Since there is no one-size-fits-all solution to optimize the balance between high diversity gain and low monitoring cost, in this paper, we build a testbed and define a use case to study the tradeoff optimization.

We make three contributions. First, we present the exposed MAPE-K model to involve external customers in the automation CL. Then, a testbed of network function virtualization (NFV) type is built to verify the exposed MAPE-K model.The testbed deploys a LTE network with OpenAirInterface (OAI) virtualized mobile core network (vEPC), orchestrated by Open Source MANO (OSM), which has an internal Monitoring module OSM MON. External Monitoring tools, *sysstat* for VM performance and packet captures (*pcap*), are added to complement OSM MON. Last, we run External Analytics to investigate how each data set is insightful in detecting and predicting specific service issues emulated in the experiments. Preliminary results are presented to verify the values of external monitoring and analytics in enhancing the performance of the automation system.

The paper is organized as follows. Section II introduces the exposed MAPE-K CL model. Section III applies the model to build hybrid CL and automate SA in an OSM-based orchestration system. Section IV describes the testbed and experiment scenarios, followed by the analysis results in Section V. The paper concludes with challenges and future work in Section VI.

II. Exposed MAPE-K CL Model

We first apply the MAPE-K model to the SA framework [5] and then elaborate the benefits for stakeholders.

A. Architecture

CL control is a mechanism using feedback to monitor and regulate a managed entity to achieve a specific target. The MAPE-K CL model, introduced in the autonomic computing paradigm (green components in Fig. 1), is aligned with the CL model proposed in ETSI GS ZSM009-1 specification [8]. It consists of four stages (Monitor, Analyze, Plan and Execute) plus one Knowledge base. The CL starts with the *monitor* stage that tracks the status and behaviour of the managed entity, which could be a network system (element, domain or service) or another CL. The collected data is processed and aggregated by the *analyze* stage to assess the performance of the managed entities and determine if the pre-defined targets are met. If not, stage *analyze* feeds insights into the *plan* stage that makes appropriate decisions against the managed entity. The generated plans are usually high-level and technology-agnostic (*e.g.*, in the form of workflows) and thus need to be executed by the *execute* stage in the form of executable actions, applicable to the context and technology in use. The outcome of executions is validated by observing the new state of the managed entity and evaluating the gap with the desired targets, via the *monitor* stage. All four stages can access and update the *knowledge* base, the logical center of the CL and an important source of context information for each stage.

In [8], each stage of the MAPE-K model can take external *stimuli* signals from external entities or CLs to enrich the Knowledge base, *e.g.*, context and environment information that might influence the behavior of the managed entity. The analysis of [5] exhibited that the *monitor* and *analyze* stage are most likely to be exposed. Correspondingly, we establish an exposed MAPE-K CL with these two stages exposed. Note that service exposure could be bidirectional (depending on the exposure level [3]). The the internal MAPE-K CL (green stages) can take stimuli from external stages, E(xternal)-*monitor* and E(xternal)-*analyze* (the orange stages in Fig. 1), operated by external entities (3rd party solution providers or vertical customers). Hybrid CLs are formed by combining the internal CL with the external stages in two ways: i) with only *E-monitor*; or ii) with both *E-monitor* and *E-analyze*. The former relies on I(nternal)-*analyze* (*I-analyze*) to aggregate the external and internal monitoring data whereas the latter uses *E-analyze*. In practice, many open source orchestration systems like OSM provide plug-ins to accept monitoring data from various monitoring tools and support *E-monitor* to broaden their telemetry scope. However, their *I-analyze* is usually limited with resource or agility (*e.g.*, lack of adaptivity in AI/ML models for different types of data collected by *E-monitor*). Then, it is more flexible and preferable to use *E-analyze* and offload the data aggregation task from *I-analyze* to *E-analyze*.

E-analyze is driven by customers. It does not consume network resources and thus can optimize the performance against the cost based on the customer's budget. Moreover, directly feeding *E-monitor* data into *E-analyze* saves the cost of developing extra plug-ins in internal CL to integrate with *I-analyze*. The outcomes of *E-analyze* (*e.g.*, data interpretation, detected anomalies, diagnosed root causes, recommended scaling policies) are fed into *I-analyze* to i) generate harmonized insight for *I-plan*, and ii) update the Knowledge base. In this way, external entities only receive monitoring data eligible to be exposed and provide analytical insight as recommendations, without meddling with operator's operations in the *I-plan* and *I-execute* stages. The internal CL has a full control of the *plan* decisions. On the other hand, its data richness and analytical power are enhanced with the addition of *E-monitor* and *E-analyze*. Then the exposed MAPE-K CL reaches a balance between data protection and performance improvement.

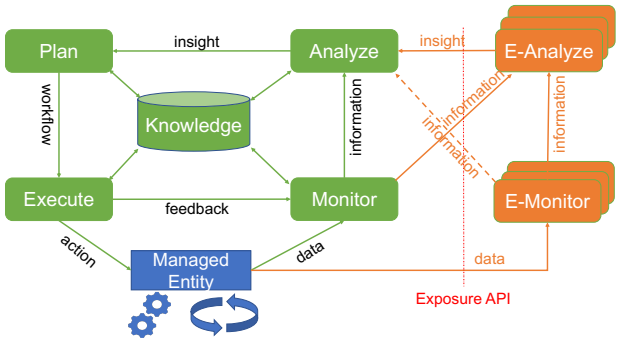

Figure 1: Exposed MAPE-K closed loop

B. Benefits

The exposed MAPE-K CL is beneficial for a wide scope of stakeholders.For **NOP/CSP** (communication service provider), offloading the workload of *I-monitor* and *I-analyze* to *E-monitor* and *E-analyze* not only saves the consumed resources and cost, but also enhances the flexibility and diversity of the hybrid CL. Most *I-analyze* solutions are relatively simple and cannot meet the requirements of modern automation systems. The inclusion of *E-monitor* and *E-analyze* creates new opportunities for NOP/CSP to build customized communication services with value-added Monitoring and/or Analyzing services to meet the differentiated demand of vertical customers, based on their service requirements and budget. Besides, the insight from *E-analyze* is customer-related and a valuable addition to the Knowledge base, *e.g.*, useful for NOP/CSP to correlate its network behaviour or service performance with customer experience.

For **vertical customers**, participating in the exposed MAPE-K CL allows them to directly interact with network operations and management and thus improve their experience more effectively. They can initiate more proactive actions to reduce or eliminate the root causes responsible for their experience degradation, *e.g.*, regulate traffic flows or update SLA if they generate more traffic than agreed. Additionally, with an access to the *I-monitor* data, they will be notified of service degradations earlier than before.

For **solution providers**, AI/ML solutions can be customized to target at a specific automation task, which increases the analytics performance. With a broader and deeper collaboration between *E-monitor* and *E-analyze*, solution providers can focus on optimizing the tradeoff between monitoring cost and analytics performance. Fig. 1 shows a limited number of interfaces between the internal CL and external stages such that solution providers can dedicate to developing efficient solutions instead of concerning with how to integrate with the internal CL.

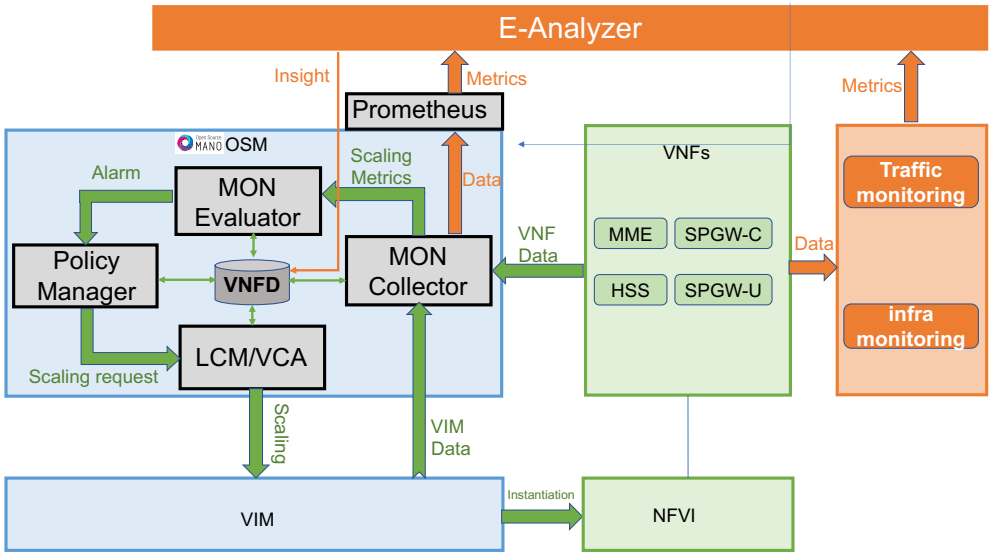

Figure 2: Architecture of joint internal and external CL based on OSM

III. Use Case: SA auto-scaling

The exposed MAPE-K CL can be applied to solve various automation tasks, such as SA. Resource scaling is a typical SA action and has been widely studied to optimize resource utilization. In 5G networks with NFV, software-defined network (SDN), and multi-access edge computing (MEC), auto-scaling is expected at individual network domains (*e.g.*, transport network with SDN [9], core network, and E2E network) or different planes (*e.g.*, scaling Data Plane (DP) virtual network function (VNF) [10]).

Most auto-scaling mechanisms are based on active monitoring of the resources and usually reactive and threshold-based. Their performance is not ideal due to the incapability of dealing with and adapting to frequent changes in networks or traffic timely. One direction to improve auto-scaling efficiency is *traffic prediction*. In [11], proactive auto-scaling decisions are made preemptively by assessing the behavior of historical traffic over time. In [12], centralized and federated deep learning is leveraged to forecast traffic models in a Kubernetes prototype ans scale VNFs. However, traffic prediction is based on traffic monitoring, which is costly to deploy in operating networks. Then a challenge arises, *i.e.*, running traffic prediction in a network without sufficient traffic data. One solution is to introduce substitute data via the exposed MAPE-K model.

Consider a LTE network with an orchestrator of OSM. Compliant with the ETSI NFV MANO architecture, OSM contains functionality modules to support VNF auto-scaling with a Virtual Deployment Units (VDU) granularity for any available metric. OSM MON collector gathers metrics on the infrastructure, *e.g.*, NFV infrastructure (NFVI) metrics like CPU and memory utilization, packets sent and received. These metrics are evaluated by MON evaluator, based on the VDU metrics and the corresponding thresholds predefined in the VNF descriptor (VNFD). If a threshold is crossed, OSM MON evaluator sends a notification on the Kafka bus, which triggers the OSM POL (policy manager) module to call for scaling actions. Then the lifecycle management (LCM) module executes scaling by instantiating a new VDU instance. In this way, an internal MAPE CL is created with VNFD acting as a Knowledge base (Fig. 2).

In this OSM internal CL, the *I-monitor* OSM MON collects basic infrastructure metrics and the *I-analyze* runs simple one-metric threshold-crossing (*e.g.*, CPU over-/under-utilization) to raise alarms for VNF scaling. Such a scaling policy is simple to implement but susceptible to the fluctuation of the monitored metrics, which would lead to frequent but unnecessary scaling actions. Particularly, OSM MON does not monitor traffic. To minimize the impact of fluctuation and facilitate prediction, an *E-Monitor* is added to acquire more data and produce more stable scaling decisions: VM-level *infrastructure data* and packet-level *traffic data* (Fig. 2). Then *E-analyer* aggregates and analyzes data from OSM MON collector (via Prometheus) and *E-monitor*, detects and predicts the need for scaling and finally recommends new scaling policies. Since OSM does not open the interfaces with MON evaluator directly, the insight from *E-analyze* is used to update the VNFD scaling policy, which impacts MON Evaluator and POL indirectly. Eventually an exposed MAPE-K CL is established on top of the OSM internal CL.

With the inclusion of traffic data from *E-monitor*, the exposed CL provides an opportunity to use the well-developed and mature traffic-prediction AI/ML mechanism and improve the auto-scaling performance. Furthermore, *E-analyze*, by aggregating and analyzing data of the same entity but at different angles, realizes the *diversity gain*, which can reduce the monitoring cost. Specifically, we develop a three-step analysis approach: 1) identify the mapping between data sets and events causing performance degradation; 2) find the correlations between the three data sets; 3) select low-cost data set to detect and predict events, based on the results of 1) and 2).

IV. Testbed and Experiment Setup

A. The OSM Testbed

Testing the concepts above requires full control of a testbed, thus we build our own OpenAirInterface / OSM based testbed which we have presented in [13], [14]. Fig. 3 illustrates our testbed. OSM interacts with virtual infrastructure manager (VIM) (OpenStack in this case) to instantiate the VDU as virtual machines (VM)

at the NFVI. The vEPC is realized by the EPC implementation of OAI at SIMULAMET [13], [14]. It consists of 4 VDUs *Home Subscriber Server* (HSS), *Mobility Management Entity* (MME), *Control and User Plane of the Packet Data Network Gateway* (SPGW-C and SPGW-U). The testbed creates a 4G network topology, but since we focus on the feasibility of a service assurance solution, the exact technology of the testbed's components is not important. In this context, our results can be generalized to any mobile network technology such as 5G.

The User Equipment (UE) is a regular PC, running UBUNTU 20.04 LTS "Focal Fossa", equipped with a HUAWEI E392 4G USB modem to emulate mobile devices. NETPERFMETER[1] is used for generating various traffic flows (TCP, SCTP, UDP) between the UE and a peer server in the Internet, running UBUNTU 18.04 LTS "Bionic Beaver". Besides, eNodeB (eNB) is emulated by a regular Linux PC, running the eNB software from OAI. It has a Software-Defined Radio (SDR) board with an ETTUS B210 connected via USB 3.1.

B. Data description

The testbed monitors the health of the vEPC system as well as the traffic at three vantage points (Fig. 3):

a) OSM *Metrics:* exposed via Prometheus: In OSM, the metrics collected by MON collector are stored in a PROMETHEUS time series database (TSDB) and exposed via a REST API (with port 9091) to external customers. The MON collector collects metrics of each VDU VM from two sources: NFVI metrics via OpenStack telemetry and customized performance metrics via juju charms in OSM, set up in KUBERNETES and running in a container of the corresponding VM hosting the VDU.

b) SYSSTAT *metrics:* : As a complement to the OSM metrics, each VDU VM runs SYSSTAT[2] to collect fine-granular statistics about CPU utilisation, disk utilisation, and network I/O. Compared to OSM, *sysstat* monitors with a *higher monitoring frequency*, produces much *more VM metrics*, and separates *individual network interfaces*.

c) Packet capture PCAP *metrics:* : Both vEPC VDUs and the UE PC run `tcpdump` to record packet capture traces (PCAP) at all network interfaces. As a network analysis tool, `tcpdump` captures and filters packets (*e.g.*, by protocols, IPs, ports, etc.) flowing through each network interface. Flow/packet-level metrics can be calculated from the PCAP metrics, such as packet loss rate or packet delay and jitter, which are more reflective of the network performance than that of the infrastructure metrics in OSM and SYSSTAT.

In summary, by monitoring individual network interfaces, SYSSTAT data has a higher granularity than OSM data whereas PCAP data further distinguishes the packet flows of each network interface by protocol, IPs and ports. Naturally, the higher monitoring granularity provides a higher chance for *E-analyze* to succeed, but at the price of higher monitoring cost. In practical large-scale operations, packet capture is usually turned off to save on data storage. The selection of proper *E-monitor* tools depends on the *scenarios* where SA is triggered due to the service performance degradation.

[1]NETPERFMETER: https://www.uni-due.de/~be0001/netperfmeter/.

[2]SYSSTAT: https://github.com/sysstat/sysstat.

Exp	Scenario	UE Traffic	Description
1	normal	TCP	Link rate 1 Mbps
2	normal	TCP or SCTP	Upload, Download, bidirectional
3	normal	TCP + UDP	Light UDP: exponential distribution
4	network interface misbehaviour	TCP + UDP	10% loss rate @egress of SPGW-U
5	link limitation	TCP + UDP	link rate 2 Mbps with 250ms delay
6	link congestion	TCP + UDP	Heavy UDP: rate 9 Mbps UDP: SPGW-U $\longrightarrow$ eNB

Table I: Overview of experiments

C. Experiment scenarios

A series of experiments are designed to emulate the *scenarios* where service performance degradation occurs and scaling would be needed. Referring to [7] for wireless networks with cloudified architectures, we select the following scenarios:

1. Link Congestion: competing flows are generated by NETPERFMETER to congest a link at which the UE traffic flow traverses. The competing flows are UDP, generated at the SPGW-U and of various data rate to reflect the different interference level towards the UE traffic.

2. Link limitation: since the connection links are shared among multiple UEs, the link capacity available for a UE is limited. On the SPGW-U, we run Linux Traffic Control [3]on the egress of the SGi interface (towards the server) and the S1-U interfaces (towards the UE) to configure NETEM and TBF (token bucket filter) queuing disciplines and control the link rate (Fig. 3). Additional delays can be added on each network interface to emulate the consequence of lowered link rate, *e.g.*, on satellite communications.

3. Network interface misbehaviour: when a network interface is faulty, it often causes packet losses. We emulate it by configuring NETEM on the target interface.

4. VM resource overload: A Linux stress tool STRESS is applied to the VMs hosting the vEPC VDUs and stresses resources like CPU, memory and storage.

Table I provides a summary of the experiments and related scenarios. Experiments 1 to 3 are normal and set up baselines for normal behaviour. Experiments 4 to 6 emulate scenarios $1-3$ and investigate whether and how the three data sets capture the performance degradation caused by these scenarios. The related data sets are publicly available [15] to satisfy the reproducibility principle.

V. EXPERIMENT RESULTS

In this section, we analyze the data of the six experiments in Table I and verify the value of the proposed exposed MAPE-K CL. First, a comprehensive filtering process is conducted toward the monitored data to reduce the number of metrics fed into the ML models. Considering that all three data sets are generated from the same VMs (PCAP has an additional source, which is the UE), their metrics are correlated at a certain level, which results in redundancy. Table II lists the key metrics of each data set considered as of high interest after filtering.

In all experiments, the UE traffic is of the TCP type, arriving at the interface *ens*6 and leaving the interface *ens*5 of the SPGW-U VM, in the downlink direction. The three scenarios of Exp. 4 - 6 emulate events that

[3]http://linux-ip.net/articles/Traffic-Control-HOWTO/

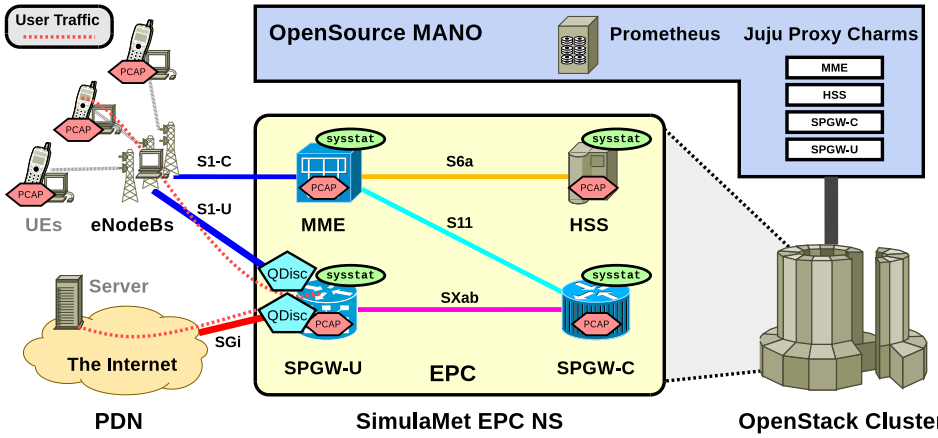

Figure 3: The SimulaMet Testbed Setup: OPEN SOURCE MANO with NFVI and vEPC

Data type	granularity	Key Metrics
OSM	VM	CPU load, memory/disk usage, bytes in/out, packets in/out
SysStat	Network interface	disk/memory usage, data written/discarded, cache metrics, I/O tasks/faults,udp/tcp connection counters, packets tx/rx
PCAP	Protocol, IP, port	IP source and destination, tcp traffic metrics, tcp/udp source/destination port, IP/frame/frame capture length

Table II: Example metrics of the three data sets

are not originated at the infrastructure where vEPC is operating (server or VMs). Plus, the OSM data does not have interface-level granularity. As a result, the OSM metrics behave flatly without visible variations in Fig. 4 and do not capture any misbehaviour. On the other hand, SYSSTAT data has a finer granularity of individual network interface. As shown in Fig. 5 for Exp. 4, the fluctuation in the received packets at *ens*6 indicates the impact of the UDP background traffic whereas the obvious changes, observed after time $11:26$ in the transmitted packets at the egress interface *ens*5, are aligned with the event that 10% packets are purposefully dropped at *ens*5. In other words, the SYSSTAT data is more insightful than the OSM data to capture the network interface misbehaviour.

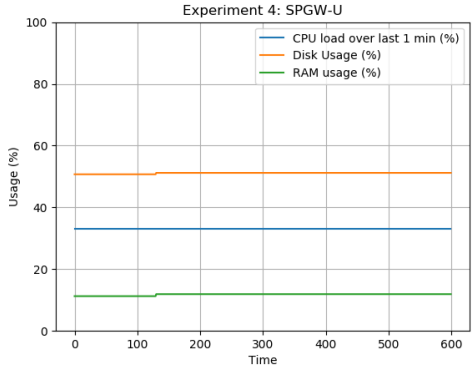

Figure 4: Physical components usage from OSM metrics.

Given the above observations, we first select the SYSSTAT data to run ML algorithms for forecasting the anomalies.With the limited volume and features of available SYSSTAT data, traditional time series forecasting techniques are employed, such as autoregressive moving average (ARMA) and vectorised auto regression (VAR). For a stationary time series y_t, the ARMA(p, q) model can be broken down into two parts:

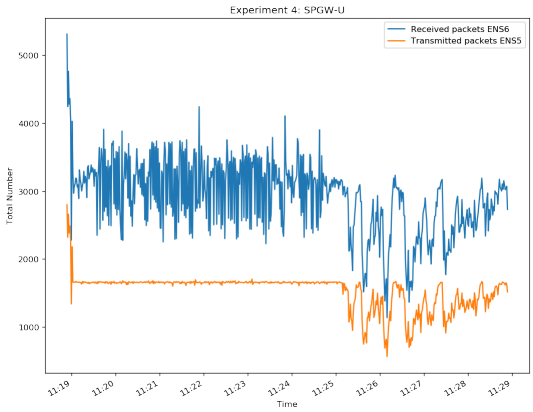

Figure 5: Received and transmitted packets in the *ens*6 and *ens*5 interfaces of the SPGW-U.

- The autoregressive (AR) part: taking into account the influence of the previous values on the predicted one:

$$y_t = a_1 y_{t-1} + a_2 y_{t-2} + \cdots + a_p y_{t-p}$$

- The moving average (MA) part: modeling the influence of noise on the future values:

$$y_t = \alpha + \varepsilon_t + \phi_1 \varepsilon_{t-1} \ldots \phi_q \varepsilon_{t-q}.$$

Finally, given values $y_{t-1}, \ldots y_{t-p}$, ARMA(p, q) estimates the value y_t as follows:

$$y_t = \sum_{i=1}^{p} a_i y_{t-i} + \alpha + \varepsilon_t + \sum_{j=1}^{q} \phi_j \varepsilon_{t-j}. \qquad (1)$$

Fig. 6 shows the results of ARMA fit by using the SYSSTAT metrics. The ARMA forecast is able to predict the downtrend behavior. In particular, ARMA$(2, 3)$ has been found to be the best fit for forecasting both transmitted and received packets at a network interfaces.

VAR is a multi-dimensional generalisation of the AR model. Given a k-dimensional time series y_t =

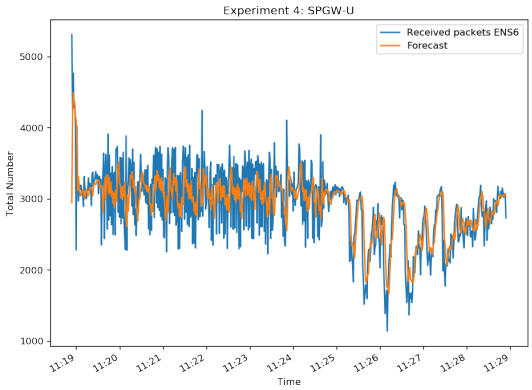

Figure 6: *Ens*5 received packets ARMA fit.

$(y_{1,t}, \ldots, y_{k,t})$, the VAR$(p)$ model relates y_t to its previous p components of the time series as follows:

$$y_t = \alpha + \sum_{i=1}^{p} a_i y_{t-i}, \tag{2}$$

where α is a k-dimensional vector and a_i is a square matrix of order k, for any i in $1, \ldots, p$. Note that, the VAR(p) model is a very powerful forecasting tool as it encodes the correlation between the components of the time series. Fig. 7 displays the VAR(9) fit of the transmitted packet rate at ens5 interface of SPGW-U and received packet rate at ens6 interface of SPGW-U, exhibiting a visible negative trend in correspondence to the packet loss event. As one can deduce from the plot, the prediction leverages on the correlation between the two components, that is, transmitted and received packets.

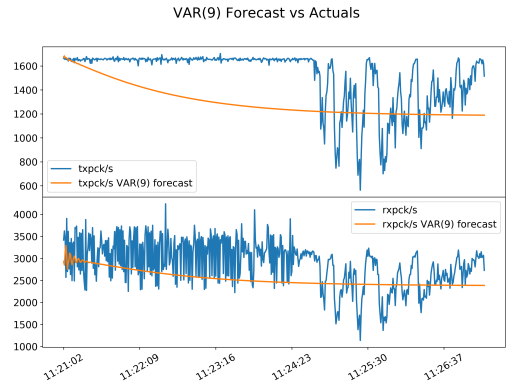

Figure 7: Transmitted packet rate from the *ens*5 interface of the SPGW-U and received packet rate at the *ens*6 interface forecast by the VAR(9) model.

Overall, the SYSSTAT metrics are good indicators of events causing a reduction in the service rate (*e.g.*, transmitted packet rate) of a UE traffic flow, which may demand resource scaling. They can be used to predict and recommend proactive measures in advance to avoid the corresponding performance degradation. Nevertheless, due to the lack of separations between traffic flows, SYSSTAT does not fully answer the question on what exactly causes the service rate decrease, *e.g.*, the network interface misbehaviour as in Exp. 5 or the heavy interference UDP flow as in Exp. 6. Further analysis on PCAP is needed to identify the root cause.

VI. Conclusions

The observations from the experiments justify that the exposed MAPE-K model, with *E-monitor* and *E-analyze*, indeed creates more opportunities to capture anomalies and paves the way for more a advanced and accurate analysis. A complex network may fail for many different reasons but exhibit similar symptoms. Data diversity is necessary to deal with such a delicate complexity and precisely localize the root causes, but it is also expensive. A key objective is to monitor the *right data* with the *right granularity* as the foundation of selecting and running the *right AI* model. The exposed MAPE-K model is proved to be a useful tool for learning the knowledge and gaining the experience to achieve the objective. As next steps, we plan to study ind-depth the tradeoff between monitoring granularity and analytics performance.

Acknowledgment

This work has been supported by the European Community through the 5G-VINNI project (grant no. 815279) within the H2020-ICT-17-2017 research and innovation program.

References

[1] B. Schmidt, "Artificial Intelligence Applications in Telecommunications and other network industries," *Telecommunications Policy*, p. 101977, 2020.
[2] GSMA, "White Paper: An Introduction to Network Slicing," 2017.
[3] NGMN Alliance, "Security Aspects of Network Capabilities Exposure in 5G, v1.0," 2018.
[4] 3GPP, "5G System: Network Data Analytics Services," 2020.
[5] M. Xie, J. S. Pujol-Roig, F. Michelinakis, T. Dreibholz, C. Guerrero, A. G. Sánchez, W. Y. Poe, Y. Wang, and A. M. Elmokashfi, "AI-Driven Closed-Loop Service Assurance with Service Exposures," in *29th IEEE European Conference on Networks and Communications (EuCNC)*, Jun. 2020.
[6] IBM, "An architectural blueprint for autonomic computing," *IBM Autonomic Computing White Paper*, 2006.
[7] Patounas, Georgios and Xenofon Foukas and Ahmed Elmokashfi and Mahesh K. Marina, "Characterization and Identification of Cloudified Mobile Network Performance Bottlenecks," *IEEE Transactions on Network and Service Management*, 2020, ISSN 1932-4537.
[8] ETSI, "ETSI GS ZSM 009-1 v0.10.5: Zero-touch network and Service Management (ZSM); Closed-loop automation; Enablers," Feb 2020.
[9] Y. Lee, R. Vilalta, R. Casellas, R. Martinez, and R. Munoz, "Auto-Scaling Mechanism in the ICT Converged Cross Stratum Orchestration Architecture for Zero-Touch Service and Network Management," in *IEEE International Conference on Transparent Optical Networks (ICTON)*, 2018.
[10] B. Tamma, "Auto Scaling of Data Plane VNFs in 5G Networks," in *13th International Conference on Network and Service Management (CNSM)*. IEEE, 2017, pp. 1–4.
[11] R. Mukherjee, "Auto-Scaling VNFs using Machine Learning to Improve QoS and Reduce Cost," in *IEEE International Conference on Communications (ICC)*. IEEE, 2018, pp. 1–6.
[12] S. Riggio, "Centralized and Federated Learning for Predictive VNF Autoscaling in Multi-Domain 5G Networks and Beyond," *IEEE Transactions on Network and Service Management*, 2021.
[13] T. Dreibholz, "Flexible 4G/5G Testbed Setup for Mobile Edge Computing using OpenAirInterface and Open Source MANO," in *Proceedings of the 2nd International Workshop on Recent Advances for Multi-Clouds and Mobile Edge Computing (M2EC) in conjunction with the 34th International Conference on Advanced Information Networking and Applications (AINA)*, Caserta, Campania/Italy, Apr. 2020, pp. 1143–1153, ISBN 978-3-030-44037-4. [Online]. Available: https://www.simula.no/file/m2ec2020pdf/download
[14] ——, "A 4G/5G Packet Core as VNF with Open Source MANO and OpenAirInterface," in *Proceedings of the 28th IEEE International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, Hvar, Dalmacija/Croatia, Sep. 2020.
[15] M. Xie, F. Michelinakis, T. Dreibholz, J. S. Pujol-Roig, S. Malacarne, S. Majumdar, W. Yi Poe, and A. M. Elmokashfi, *An Exposed Closed-Loop Model for Customer-Driven Service Assurance Automation*. Zenodo, Apr 2021, doi: 10.5281/zenodo.4707588.