An Autonomic Management System for IoT Platforms based on Data Analysis Tasks

Clovis Anicet Ouedraogo, Jose Aguilar, Christophe Chassot, Samir Medjiah, Khalil Drira

An Autonomic Management System for IoT Platforms based on Data Analysis Tasks

Clovis Anicet Ouedraogo[1], Jose Aguilar[2, 5, *], Christophe Chassot[1,4],
Samir Medjiah[1, 2], Khalil Drira[3]

[1] CNRS-LAAS, Toulouse, France
[2] CEMISID, Universidad de Los Andes, Mérida, Venezuela
[3] Université de Toulouse, UPS, Toulouse, France
[4] INSA, F-31400 Toulouse, France
[5] Universidad EAFIT, Department of Informatics and Systems, Medellin, Colombia
[*]Corresponding author
{ouedraogo, chassot, medjiah, khalil}@laas.fr, aguilar@ula.ve

Abstract: In this work, we propose an Autonomic Management System (AMS) for the Internet of Things (IoT) platforms, which uses the concept of autonomic cycle of data analysis tasks to improve and maintain the performance in the IoT platforms. The concept of "Autonomic Cycle of Data Analysis Tasks" is a type of autonomous intelligent supervision that allows reaching strategic objectives around a given problem. In this paper, we propose the conceptualization of the architecture of an AMS composed by an autonomic cycle to optimize the Quality of Services (QoS), and to improve the Quality of Experiences (QoE), in IoT platforms. The autonomous cycle detects and discoveries the current operational state in the IoT platform, and determines the set of tasks to guarantee a given performance (QoS/QoE). This paper presents the details of the architecture of the AMS (components, knowledge models, etc.), and its utilization in two case studies: in a typical application in an IoT context, and in a Tactile Internet System.

Keywords: Autonomic Computing, Internet of Things; Data analysis tasks; Autonomic Management System; Quality of Experiences

1. Introduction

The IoT integrates different domains, such as context awareness, autonomous computing, mobile computing, communication protocols and devices with embedded sensors and actuators, among other things. This integration allows generating a global dynamic system with a myriad of objects that can be used for different applications, in order to develop smart systems. These smart IoT systems (applications) are being developed in several areas, such as health, agriculture, industry, and transportation, among others.

The classical components in an IoT Ecosystem, according to the oneM2M international standard as detailed in [1], are: Devices, Network, IoT Platform, and Applications. Devices collect the data, and in some cases, preprocess these data and execute specific tasks. The Applications produce something exploiting the set of interconnected Devices; the Network enables the communication between the IoT Ecosystem components; and finally, the IoT

Platform solves the non-functional requirements of Applications and Devices, using smart capabilities like autonomy, among other things.

Some of the factors that have a significant impact on the behavior of IoT applications are the QoE and the QoS, in order to better meet the requirements of the context. QoE (also called QoX or QX) is a holistic concept about the user's experience (taking into account emotion like pleasure or anger) with a service (e.g., web browsing). Quality of service (QoS) is a measure of the overall performance of a context, such as a service or communication platform, seen by the users. In general, these concepts indicate the degree of conformity of the users or applications with the prevailing situation in the environment and their requirements. An IoT platform must guarantee the provisioning of the resources to reach a given QoS or QoE. In this paper, we propose an Autonomic Management System (AMS) for IoT platforms, in order to improve its performance, such that the AMS guarantees the provisioning of the resources to reach a given QoS or QoE. The proposed AMS for IoT uses the concept of the autonomic cycle of data analysis tasks.

Data Analysis (DA) is a science to analyze data, in order to convert them in knowledge to improve the context [2, 3]. The concept of "Autonomic Cycle of DA Tasks" has been presented in [2, 3], and it is an autonomous intelligent supervision approach to reach strategic objectives. The autonomic cycles integrate a set of DA tasks to achieve the strategic objectives [2, 3, 30]. The tasks interact with each other and have different roles in the cycle: Observing the process, interpreting what happens in it, and making decisions to reach the objectives.

Particularly, in this paper, we propose an Autonomic Cycle of DA Tasks, as the main component of the AMS architecture, to guarantee the provisioning of the resources to reach a given QoS or QoE. Our autonomous cycle detects the current operational state in the IoT platform, and according to the detected operational state, determines the set of tasks to guarantee a given QoS/QoE.

This work focuses on the conceptualization of the architecture of the AMS, and describes two case studies to determine its usefulness in different contexts. The first one is a typical application of IoT, which automatically watches and monitors wildfires. It enables early fire spotting by using temperature sensors and surveillance cameras. When it has detected a fire, it alerts wildfire management authorities, with live streaming over the fire zone. In this context is mainly important the QoS. The second case study is a Tactile Internet Architecture that must guarantee haptic communication, in order to establish a link between humans and unknown environments, in a similar way as the auditory and visual senses (in this case, for the sense of touch). In this context, haptic communications add an extra dimension to traditional audiovisual communication, for truly immersive steering and control in remote environments. In this case is mainly important the QoE.

The main contributions of this paper, with respect to previous papers like [12, 18, 21, 22, 29], are: an exhaustive description of the AMS architecture, an extensive comparison with previous similar works, and finally, a detailed illustration of the utilization of the architecture in two case studies. This paper is organized as follows: the next section introduces the related works, section III the preliminary ideas, and section IV presents the architecture of the AMS. Then, section V describes the case studies, next, section VI outlines some discussions, and finally, the last section presents the conclusions.

2. Related Works

In this section, we present several papers about QoS/QoE management in IoT platforms. We evaluate the literature according to the following requirements (see Table 1):

- R1: Dynamic adaptation
- R2: Traffic differentiation
- R3: Autonomic approach

According to their scope, we identify in the literature three categories of strategies to provision QoS/QoE in IoT platforms:

- S1: The first group is papers that *ensure QoS/QoE in IoT platforms*.
- S2: The second group is papers that *use IoT platforms to provide application QoS*, through the reconfiguration of the underlying network.
- S3: The last group is papers that propose *hybrid approaches*, where *IoT platforms* are considered as problematic, but also, like a tool to overcome the problem of the network.

Table 1 presents a summary of the related works, where:

- P is a prototype, S is a simulation, and x means no evaluation is carried out (see Evaluation column).
- ✓ means the criterion is met and X the criterion is not met (see the criteria column).

Table 1. A summary of related works

Work	Strategy	Major Contribution	Evaluation	Criteria		
				R1	R2	R3
[5]	S1	It defines a holistic approach through specific resource management policies, takes into account the functional dependency between application components.	P	✓	X	X
[6]		It introduces a QoS-oriented AMS for IoT platforms and an architecture to detect QoS violations.		X	✓	✓
[7]		It proposes a static management approach of an IoT platform, which includes behavioral and structural reconfiguration actions.	P	X	✓	✓
[8]		It defines a model using the queuing theory to determine the performance of the IoT platform entities.	P	X	✓	✓
[9]		It presents dynamic and autonomic management of the QoS required by IoT applications in a heterogeneous environment. Particularly, it proposes a redirection mechanism in a QoS-oriented NF.	P	✓	X	✓
[10]		It proposes a modular framework to ensure at the middleware-level the QoS of applications, through QoS-oriented mechanisms deployed dynamically on the middleware entities.	P	✓	✓	X
[11]		It proposes a QoS control scheme for IoT systems, based on the Markov game model, to allocate IoT resources while maximizing QoS performance.	S	✓	X	✓
[12]		It proposes IoT-MP (Internet of Things Management Platform), which considers the low-cost and low-power requirements of things, and the heterogeneity, scalability, and autonomy of the communications supported in the IoT.	S	✓	X	X

[13]		It presents a state-of-the-art survey specifically aimed at analyzing the middleware with quality of context support for IoT applications.	S	✓	X	X
[14]		It proposes an architectural approach that they argue that can be used as a reference model for IoT platforms.	X	X	X	X
[15]		It proposes a classification of functional and non-functional aspects of IoT applications, and the requirements for IoT middleware, to simplify the application management for different stakeholders.	X	X	✓	X
[31]		They propose a method to improve the resource allocation problem in IoT applications	S	✓	✓	X
[32]		A scheduling algorithm improves the QoS parameters	S	X	✓	X
[33]		They propose a dew computing architecture to solve the lack of service-oriented manifestation of timely signaling, which may hinder back the IoT's envisaged speculation	P	X	X	X
[16]		It develops a middleware, called MiLAN, which allows the specification of policies for the management of the networks and sensors, in IoT platforms.	X	✓	X	X
[17]		It proposes a service-oriented platform for developers to help them define the logical functionalities.	S	X	X	X
[18]		It proposes PRISMA, a resource-oriented publish/subscribe IoT platform, which main goals are to provide: programming abstractions, services, runtime support, and QoS mechanisms, to meet application constraints.	P	X	X	X
[19]		It proposes an autonomic IoT platform for the QoS management composed of a set of QoS-oriented mechanisms that can be dynamically executed at the platform level to correct QoS degradation.	P	X	✓	✓
[20]		It presents EMMA, an edge-enabled publish–subscribe platform that addresses the QoS requirements imposed by the applications.	P	✓	X	X
[34]	S2	They design an IoT platform for smart homes based on ZigBee wireless sensor network	S	✓	X	X
[35]		They present an architecture for IoT to manage end-to-end workflow management processes	S	✓	✓	X
[36]		They propose s pattern for self-adaptive systems, named Self-organizing Coordination Regions (SCR), which goal is to organize a process of interconnecting devices into teams	P	✓	X	X
[37]		They present a platform that enables analytics on IoT captured data from smart homes.	P	X	X	X
[38]		They proposed a Quality of Experience (QoE) Management model for Future Internet Architectures	S	X	✓	X
[39]		They propose a simulation-driven platform to support both Edge and Cloud Computing paradigm to develop Ambient Assisted Living services	S	X	X	X
[21]		It presents an active QoS infrastructure of wireless sensor networks (WSN), named QISM, which is based on middleware and service-oriented. QISM includes the active regulation mechanism based on feedback and negotiation between applications and network	S	✓	X	X
[40]	S3	They propose an approach using the concept of virtualized network functions to dynamically deploy, on under-loaded nodes, additional network functions that exploit the available computing resources to differentiate the traffic processing level and to apply a QoS-oriented policy.	S	✓	✓	✓
[41]		They define a WSN-cloud-assisted internet of things (CIoT) based on two	S	✓	X	X

		optimization problems: the optimal cluster head selection, and the optimal shortest path selection				
[42]		A framework for the integration of autonomous processes based on the needs for coordination, cooperation, and collaboration in the Industry 4.0	S	✓	X	✓
Our proposal		-	X	✓	✓	✓

For the first group, these approaches consider *IoT platform* as a bottleneck, and use mechanisms to differentiate the services offered by the *IoT platform*. Agirre, et al. [5] propose a middleware that supports complete lifecycle management of a system consisting of several concurrent applications running over a distributed platform. The *platform* is driven by a quality of service-aware self-configuration algorithm, which has reconfiguration capabilities for internal and external reconfiguration events. This platform is an extension of DAMP (Distributed Applications Management Platform), which provides basic lifecycle management support for distributed applications. This work proposes a holistic approach to the entire system QoS enforcement through specific resource management policies, takes into account the functional dependency between application components. The eventual availability of redundant component replicas in alternative hardware nodes is also considered. Barnouar et al. [6] introduce a QoS-oriented AMS for IoT *platforms*. Additionally, the authors in [6] propose a concept-oriented architecture for the monitoring component, allowing detecting QoS degradation symptoms, and demonstrate the benefits that could be gained from simple network-inspired QoS-oriented adaptation actions. The paper [7] defines a static management approach of an *IoT platform,* which considers reconfiguration actions from the point of view behavioral and structural for platform-level resources and application traffic. They describe the QoS management mechanisms for the self-adaptive management of QoS at the platform level. Additionally, in [8] the same authors propose a model based on the queuing theory to analyze the *IoT platform* performance of the entities involved under two approaches, reactive and proactive. In the first case, with techniques of CEP (Complex Event Processing), and in the second case with an ARMA prediction model (Auto-Regressive Moving Average). This model is then tested in an IoT entity (a Gateway). On the other hand, in [9, 10] the authors identify two bottlenecks with respect to the QoS in the IoT platforms: the network and entities that enable an application can interact with the IoT devices, which can be solved by acting at the level of the network or the IoT platform. Also, they describe a solution that consists of dynamic and autonomous deployment of QoS management mechanisms based on the "network function" concept. Finally, they present a redirection mechanism evaluated in a case study related to vehicular transportation. Also, in [11] proposes a QoS control scheme for IoT systems, based on the Markov game model. The QoS scheme can effectively allocate IoT resources while maximizing QoS performance. Thus, they use a game theory approach to provide an effective decision-making framework for resource allocation problems that guarantee to reach the QoS requirements. In [12], the authors propose an IoT Management Platform, called IoT-MP, which considers the low-cost and low-power requirements of things, and the heterogeneity, scalability, and autonomy, of the communications supported in the IoT, in order to give control to the applications of the granularity of the disclosed information based on the context of their use (e.g., based on the time or the current location of the user). The authors of [13] present a state-of-the-art survey specifically aimed at analyzing the *IoT platform* with quality of context (QoC) support and; an *IoT platform* with QoC support for IoT Applications. This *IoT platform* was evaluated in a case study involving the development of a mobile remote patient monitoring application, in order to meet the quality of context requirements of the application. On the other hand, in [14] is carried out a systematic review of the literature, presenting a discussion of the challenges and future perspectives on IoT platform. They highlight the difficulties in achieving and enforcing a universal standard and propose an architectural approach that they argue that can be used as a reference model for IoT platforms. Finally, in [15] is presented a classification of functional and non-functional aspects of IoT applications; and the requirements for an IoT platform to simplify application management. Particularly, the platform allows application creators to define functional and non-functional aspects of applications, in a way that it can correctly deploy and manage applications. Accordingly, they define expressive abstractions to specify applications as service compositions, while hiding the complexity of IoT infrastructures. In the work [31], Mahini et al. propose a method to improve the resource allocation problem in IoT applications using fog computing. They consider two objectives: 1) reducing the internet core traffic; 2) serving subscribers with the appropriate quality of service level. In order to optimize the QoS in IoT applications, a task scheduling algorithm is proposed in [32]. Their scheduling algorithm improves the QoS parameters and comprises metrics such as response time, computation time, availability and cost. Ray et al. [33] investigate three possible solutions to the lack of service-oriented manifestation of timely signaling, which may hinder back the IoT's envisaged speculation. These solutions are considered in a dew computing architecture (i) multi-tasking of popular micro-processing modules, (ii) sensor-generated IoT stream processing in cloud-centric medium, and (iii) dew computing-based context-aware local computing.

In the second group of papers, they do not consider the platform as problematic, but rather, as a tool to overcome the problem of the network. In [16], they describe a platform called MiLAN that allows applications to specify a policy for managing the network and sensors. This is effected within MiLAN, which is very useful for sensor network applications to meet their needs. They show the effectiveness of MiLAN for the design of a sensor-based personal health monitor. In [17], the authors propose a platform based on the service-oriented paradigm to logical functionalities. The platform maps an application onto physical smart devices and actuators. The paper studies the QoS characteristics for applications using QoS attributes. After developers specify how each attribute contributes to the overall QoS, the platform determines the best mapping solution. The mapping problem is formalized as a maximum weighted bipartite problem. Finally, Silva et al. propose PRISMA[18], a resource-oriented publish/subscribe IoT platform, which main goals are to provide: programming abstractions, services, runtime support, and QoS mechanisms to meet application constraints. The programming abstractions are provided through REpresentational State Transfer (REST) interfaces; the main services provided are asynchronous communication, resource discovery and topology control; and the runtime support is for the creation, configuration, and execution of new applications in wireless sensor networks (WNS). In [19] is proposed an autonomic IoT platform-level QoS management for IoT Ecosystems. It is defined by a set of QoS-oriented mechanisms dynamically executed to solve QoS degradation problems. They analyze a case study in an Enhanced Living Environment. In [20], the authors present EMMA, an edge-enabled publish–subscribe platform that addresses the QoS requirements imposed by the applications. EMMA continuously monitors network QoS and orchestrates a network of Message Queue Telemetry Transport (MQTT) protocol brokers. It transparently migrates MQTT clients to brokers in close proximity to optimize QoS. Experiments in a real-world testbed show that EMMA can significantly reduce end-to-end latencies that incur from network link usage, even in the face of client mobility and unpredictable resource availability. Yuan and He designed an IoT platform for smart homes based on ZigBee [34]. Particularly, the smart home system is designed by combining ZigBee wireless sensor network, gateway technology and mobile terminal remote control app. Also, they improve the routing algorithm of ZigBee based on a neighbor table, and the location algorithm of ZigBee based on RSSI location and DV-Hop algorithms. Serhani et al. [35] propose an architecture for IoT to manage end-to-end workflow management processes including declarative specification and composition, orchestration, adaptation, and quality enforcement. These characteristics provide runtime intelligence for IoT workflow orchestration. In addition, it supports other smart features that include: (1) data compression for fast data transmission, and data storage adaptation, (2) integration of edge computing for local data processing that is very important for life-critical IoT workflows, among others. In the work [36], the authors propose the design pattern for self-adaptive systems, named Self-organizing Coordination Regions (SCR), in order to organize a process of interconnecting devices into teams, to solve local tasks in cooperation. This approach is decentralized, allows the integration and coordination of devices, based on continuous adaptively to context change to provide a distributed decision-making. Yassine et al. [37] present a platform that enables analytics on IoT captured data from smart homes. They use fog nodes and a cloud system to allow data-driven services and address the challenges of complexities and resource demands for online and offline data processing, storage, and classification analysis. Pereira et al. [38] proposed a Quality of Experience (QoE) Management model for Future Internet Architectures. They defined a knowledge representation model of QoE incorporated into a service delivery platform oriented to user's needs, in order to measure UX (User Experience). They configured an experimental environment to provide eHealth services to a healthcare facility. The paper [39] proposes a simulation-driven, platform named E-ALPHA (Edge-based Assisted Living Platform for Home cAre), to support both Edge and Cloud Computing paradigm to develop AAL (Ambient Assisted Living) services. E-ALPHA combines Edge, Cloud or Edge/Cloud deployments, supports different communication protocols, and fosters interoperability with other IoT platforms.

Finally, the last group of papers is about Hybrid Approaches. Hua et al., present a QoS-oriented platform for WSN, called QISM, which is based on service-oriented [21]. QISM has a regulation mechanism with feedback and negotiation between applications and network, a service publishing and subscribing architecture, among other components. QISM makes the applications adapt to the network, has good QoS control ability, and is independent of network architectures. The systematic scaling approach is used in the literature as the only option to meet Quality of Service (QoS) requirements in response to the traffic load increase in IoT platforms. In the paper [40], the authors propose an alternative approach using the concept of virtualized network functions (VNFs). In this approach, they dynamically deploy, on under-loaded nodes, additional NFs that exploit the available computing resources to differentiate the traffic processing level and to apply a QoS-oriented policy. The considered NFs extend the notion of VNF defined within NF virtualization (NFV), to take advantage of component-based software design. Finally, they formulate a multiobjective optimization problem for the efficient planning of adequate NFs, according to the considered multi-constrained context. Alameen and Gupta [41] implement a WSN-cloud-assisted internet of things (CIoT) based on two processes: one optimal cluster head selection, and one optimal shortest path selection. The first process selects a cluster head using a hybrid optimization algorithm to minimize the distance between each IoT sensor node and cluster head and consumed energy. Moreover, the same hybrid algorithm is used in the second process. Two meta-heuristic algorithms, deer hunting optimization algorithm (DHOA) and particle swarm optimization (PSO), are merged in a hybrid algorithm to solve the optimization problem. In [42], Sanchez et al. present a framework for the integration of autonomous processes based on the needs for coordination, cooperation, and collaboration in Industry 4.0. They define three autonomic cycles that allow the actors of manufacturing processes (Data, People, Things, and Services) to interoperate, considering the IoT and Big data paradigms. These autonomic cycles can create a coordinated plan for self-configuration, self-optimization, and self-healing during the manufacturing process. Like a continuation of this paper, in [43] is implemented one of these autonomic cycles, allowing self-supervising of the coordination process. This autonomic cycle is designed using the

MIDANO's methodology, and implemented and tested using an experimental tool to replay the production process event logs, in order to detect failures and invoke the autonomic cycle for self-healing when needed.

Research on autonomous management for IoT is still a maturing field. Braten et al. [44] present a review of the mechanisms for autonomous device management of IoT devices considering the management tasks, operational environment, network topology, resource constraints, scalability and management categories. In the end, they present a generalized model for autonomous device management that describes and explains the processes required for autonomous operation, unifying the insights from previous works in a framework. In [45] is presented a survey to steer IoT developers by 1) providing baseline definitions about classes of development products-methodologies, frameworks, platforms, and tools 2) reviewing IoT products through a comparative and practical approach, based on the main engineering features of IoT systems (i.e., smartness, interoperability, autonomy, and scalability). Also, in [46] another literature review has been carried out for the healthcare IoT systems. The aim of this study was to present a systematic literature review of the technologies for fog computing in healthcare IoT systems. On the other hand, Patibandla et al. [47] present a review on autonomic self-management attributes and capabilities for cloud computing, define autonomic computing architectures, autonomic requirements and modes, and examine their properties. Finally, [48] presents a systematic review of the literature on researches to improve energy management systems for smart buildings based on artificial intelligence, which grouped them according to the concept of "Autonomous Cycles of Data Analysis Tasks". This organization allows them to establish that many types of researches are in the domain of decision-making (a large majority on optimization and control tasks), and to identify potential challenges in the areas of feature engineering, or multi-agent systems, among others.

According to this literature review and previous literature review work on IoT, there are several studies about QoS management in IoT platforms. The majority of works relied on the first approach (64 % of the papers) and some apply the second approach (30% of the papers). To the best of our knowledge, only one work [21] proposes a hybrid approach, where *IoT platforms* are considered as problematic, but also, like a tool to overcome the problem of the network. Also, in the previous approaches, there is no work that proposes a data-driven approach to the management of QoS/QoE in an IoT Ecosystem. Neither, there are IoT platforms that allow the integration of different machine learning methods to improve the performance of these platforms, like is the case of the "Autonomic Cycle of Data Analysis Tasks" concept used in the definition of AMS. Finally, these works do not analyze the flexibility and scalability of the platform, to be used in different contexts (e.g., tactile Internet), to include new capabilities (e.g., fault tolerance), to simultaneously manage QoE and QoS requirements, among other things. The goal of the AMS presented in this paper is to enable the full support of these IoT requirements.

3. Preliminary ideas

In previous work, we have proposed an approach to dynamically and seamlessly deploy QoS network functions (NFs) to sustain QoS in IoT platforms [22, 23, 10]. An NF can be deployed into different forms, from classical virtualization containers (e.g. virtual machines, application containers, like NFV) to software modules in an application (e.g. the Applicative Network Functions (ANF). The dynamic deployment of NFs to meet the QoS needs of IoT applications leads to the design of the following architecture.

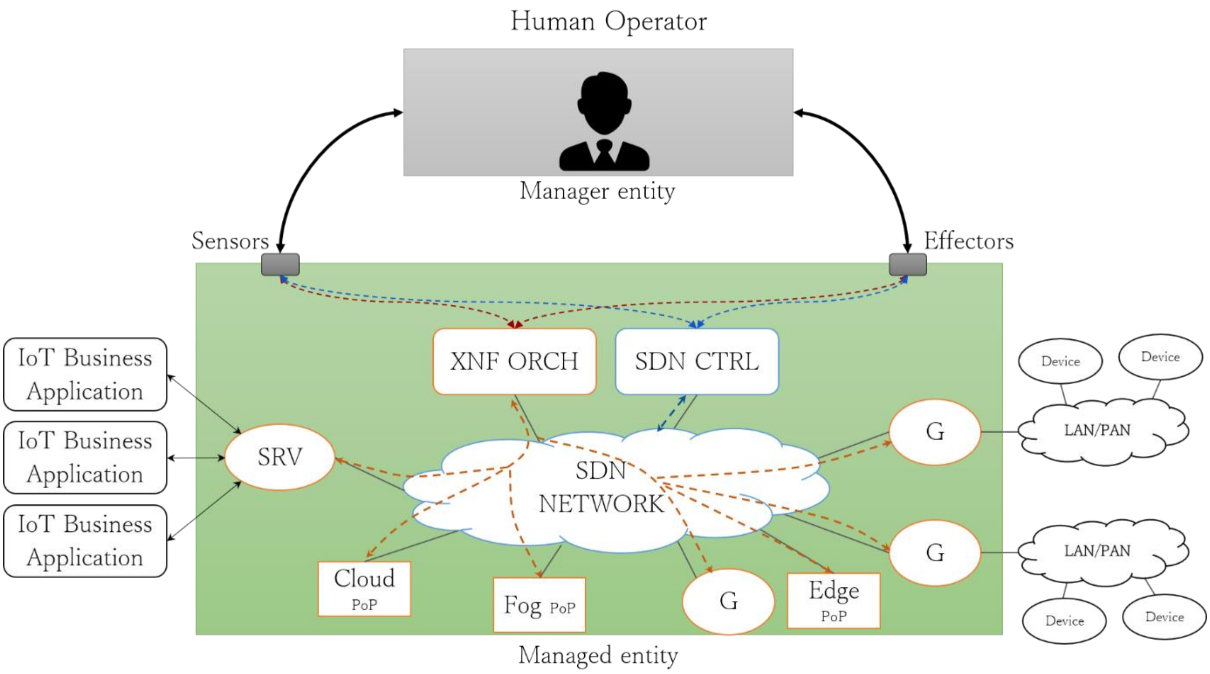

Fig. 1: Human-assisted provisioning system

In this architecture, a human operator is required. This architecture has three parts: the business applications/devices, the IoT platform (the managed entity) and the Human operator. The IoT applications can exchange data with the platform, the actuators can be controlled by them, and the sensor supply data to applications. The applications use the services offered by the IoT platform to interact with the objects (sensors, actuators), and have strict QoS/QoE requirements, for example, a bounded delay in retrieving data from sensors.

As is presented in Fig. 1, we assume three types of elements in the managed entity: the IoT platform elements (gateways and servers) on which adaptation actions are applied, the SDN switches involved in the network, and the orchestrations elements (XNFs orchestration and SDN controller) in charge of enabling these adaptations. This distributed IoT platform respects the ETSI specifications and has two types of entities: servers and gateways. The internal software architecture of these elements is supposed to be modular. They support scaling actions (via, for example, a load balancer) and/or resizing (a dynamic increase of the CPU) in case of overload. The platform elements are assumed to be located in virtual environments like Cloud, Fog or Edge. XNF orchestrator is responsible for the activation/deactivation, parameterization, deployment and migration of xNFs (VNF/ANF [10]). The SDN Controller reconfigures the SDN Switches and makes the routing between the NFs via a protocol such as OpenFlow [24]. More detail about each part is available in [10, 23].

4. Proposed AMS Architecture

The complexity of the management increases with the size and degree of heterogeneity of the considered system. This complexity can quickly become hard to manage for a human administrator [25], hence the need for the system to manage itself. One way to address this need for autonomy in the management of NF is based on the implementation of the paradigm of autonomic computing [26]. In this work, we enhance the previous architecture by adding an AMS to autonomously manage the system (Fig 2).

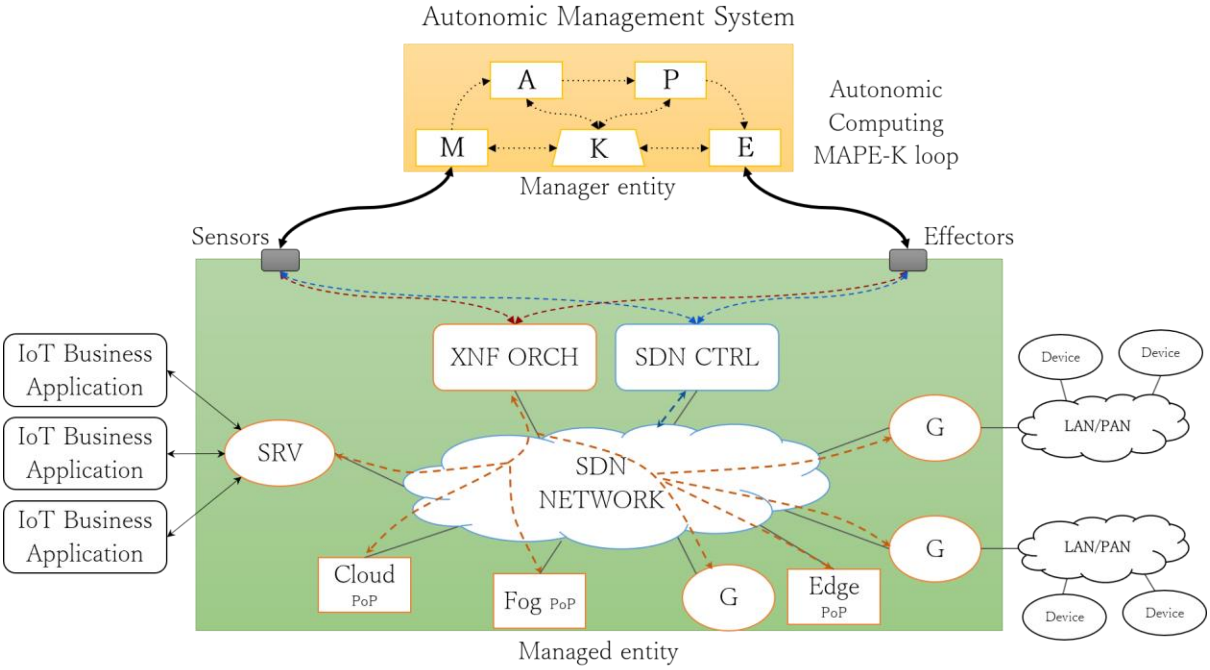

Fig. 2: Autonomous High-level architecture

The AMS is responsible for choosing the adaptation actions to be performed by the different orchestrators and controllers, with the IoT platform entities and the Cloud/Fog/Edge nodes. It ensures its general function, according to three objectives:

- To sustain the performance of the IoT platform, in order to meet the "end-to-end" QoS/QoE required by the applications. Therefore, it must ensure the coherence of the adaptation actions envisaged throughout the platform level data path, followed by the applications;
- To ensure that the implementation of its actions be done without modification of the entities located upstream and downstream of the ones concerned by the adaptation;
- To ensure the implementation of its actions without (or with a minimum) human intervention. This property is based on the autonomic computing paradigm (see Fig. 3) [4]

For this last goal, we propose a model based on the “Autonomic Cycle of Data Analysis Tasks” concept [2, 3, 4], which instances the MAPE+K model of the autonomic computing paradigm in the following way:

- Monitor (M): it catches the information from the managed entities. In addition, based on this acquired information, in case of difficulties, it generates alarms to the analysis component. In our autonomic cycle, it is composed of 3 tasks: (1-3).
- The Analyzer (A): starting from the alarms and information received, it prepares an interpretation and diagnosis, which can lead to issue or not a request to change the configuration of the managed system, to the planning component. In our autonomic cycle, the analyzer has 3 tasks (4-6)
- The Planner (P): starting from the request of the analyzer, it defines the adaptation actions to be performed to maintain the targeted "end-to-end" QoS/QoE, and sends this selection to the next component.

- The Executor (E): using the information from the previous component, it runs the adaptation actions through the ANF/VNF's orchestration and SDN control entities.

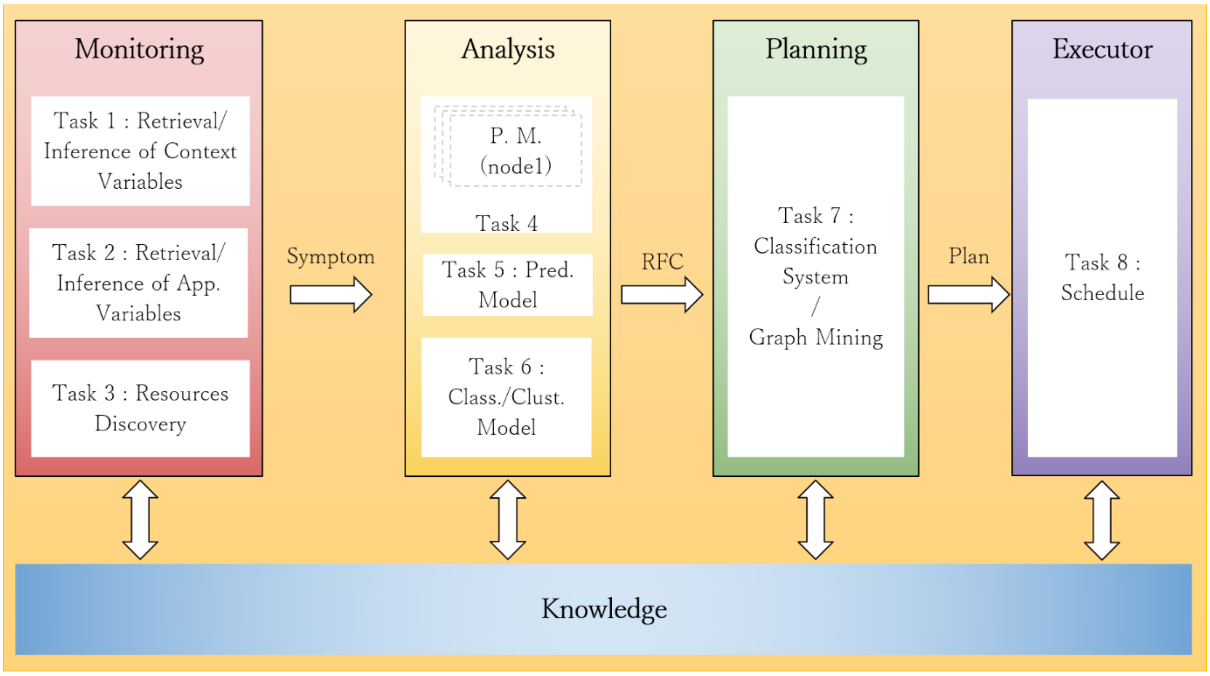

Fig. 3: AMS: Autonomic Cycle for QoS/QoE provisioning.

Let us note that in the Autonomic Computing model, the four functional elements in charge of implementing the autonomy, rely on a knowledge base (K) that including parameters, rules and models necessary for their realization (e.g. targeted QoS/QoE parameters, resource thresholds not to be exceeded, etc.)) [26]. In addition, our AMS has a set of data analysis tasks (see Table 2), which must be defined.

Table 2. Data analysis tasks in the proposed AMS

N	Task	Description
1	Retrieval and Inference of Context Variables	It carries out the recovery in the log files of the IoT platform entities, of metrics like: i) Processor (CPU) utilization, ii) Memory (ram) utilization iii) Rate of the arrival of the requests (Ta), iv) Size of the queue (Tf) of requests from each entity. This task can also infer new values since those retrieved.
2	Retrieval and Inference of App. Variables	It carries out the recovery since the log files of the Applications, of the history of the times of answer (RTT) of the requests. This RTT metric is expressed in units of time (usually ms) and must not exceed the tolerable threshold of the application.
3	Resources Discovery	It carries out the discovery since the log files of the architecture, of new resources in the platform
4	Prediction model 1	It consists in predicting the load of the IoT platform entities over a time window. The source of the data is Task 1
5	Prediction model 2	It consists of predicting the QoS/QoE offered by the IoT platform to the IoT applications, like response times (RTT) to the requests of the applications. Its data sources are Task 2 and Task 3, of all IoT platform components.
6a	Classification model	It builds the classification model for the identification of the states in different system levels
6b	Clustering Model	It builds a clustering model for the definition of clusters of the states of the platform. The inputs of this model are the previous outputs of the Ti tasks (the state in our system).

7a	Classification System	This system uses the result of tasks 3 and 7, to decide which action plan is best suited. That is, the inputs of this system are the result of the classification model and a description of the resources available in the environment (ex: fog node, ANF-compatible edge node).
7b	Graph Mining	This system builds an action plan according to the results of task 6b. The inputs of this system are: 1) the result of the clustering model, 2) a description of the topology of the environment (represented by a graph) and 3) the set of constraints to respect, in order to modify this topology (Graph Grammar).
8	Scheduler	This task sends the actions to be executed by the different members of the IoT platform, according to the action plans

5. Case Study

In this section, we present two case studies to illustrate the generality of the proposed AMS. The first case study deals in an IoT context, and the second in the Tactile Internet context; each one has different challenges, requirements, and specificities.

5.1 Case Study in the IoT context

Description of the context

Wildland fires are very frequent worldwide. For instance, according to the National Interagency Fire Center (NIFC), the number of U.S. wildfires, in 2017, was 10,026,086 acres [27]. This is an increase over the previous annual average by 36%. Adding to that, the fact of the significant related financial loss. For instance, and according to the same reference, more than 3.000 structures were lost in Southern California, and insured losses exceeded $2.5 billion during summer 2017. To solve this problem is proliferating the utilization of wildfire surveillance systems over risk area, which can simultaneously monitor wider areas, and send alarms through the network. Early detection of smoke can be carried out using pattern-recognition software. That's why in this case study, we propose to provide a forest fire intelligent system able to self-manage. This case study is based on the prototype presented in [22].

The IoT application in a wildfire surveillance system allows early fire spotting because (i) it can monitor the temperature over the forest with temperature sensors and (ii) it can detect suspecting smoke with surveillance cameras. In case of fire, it can alert with live streaming over the fire zone, which can be sent to wildfire management authorities. This prototype enables maintaining operative QoS (including latency), using SDN, NFV and edge resources.

Description of the Platform

The end-to-end architecture for IoT can be split into four domains (see Figure 4): the users' domain, the application domain, the platform domain, and the things domain. In this case study, the user domain includes the fire station, which primarily provides firefighting services for the specific geographic area. The application domain is the Wildfire detection system described above. The platform domain includes an IoT server hosted in a cloud, a distribution gateway (intermediate) hosted in a fog node, and two access gateways at the edge of the network. This IoT platform collects and routes data to the wildfire detection application. The things domain includes basic sensors, such as temperature and image sensors.

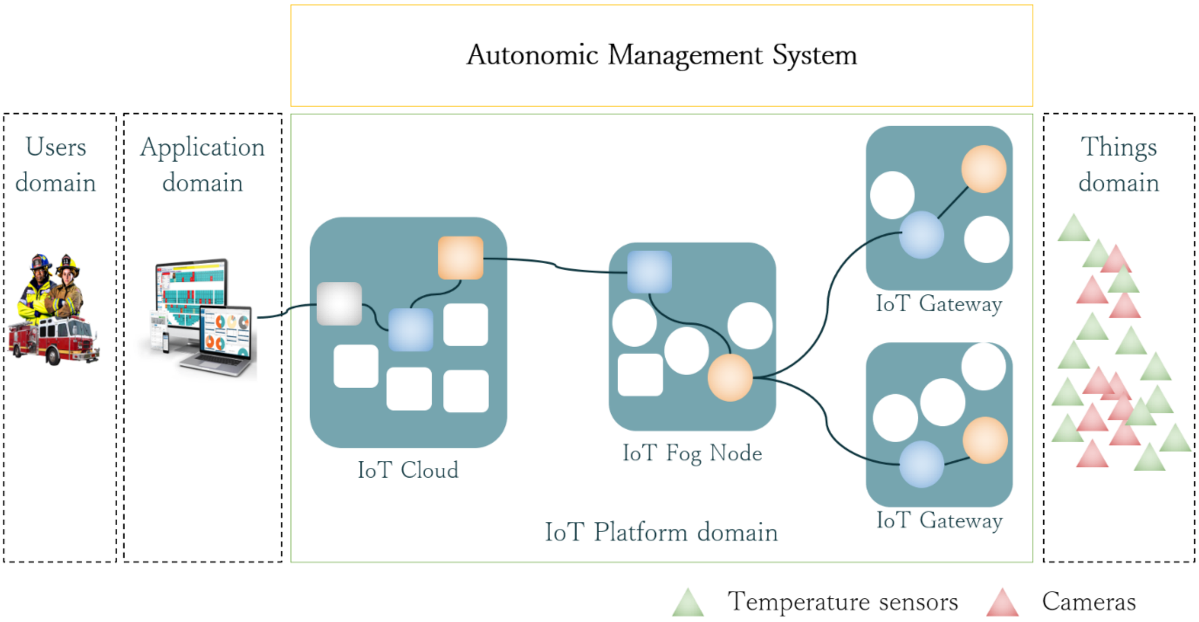

Fig. 4: IoT Platform for wildfire surveillance.

Instantiation of the proposed AMS

- The Monitor (M) is used for the collection and calculation of the data necessaries for the autonomous maintenance of the application QoS/QoE requirements. The monitor collects data from the environment and the sensors. For example, it collects the resource consumption of each IoT platform node (MEM/CPU) and the latency (time of a message to reach the source) from the environment. Also, it collects temperatures and images from the sensors deployed in the forest. The monitor also calculates data of Node Workload (ω), for characterizing the environment, and the Wildfire QoS (QoS), for characterizing the forest fire detection application
- The Analyzer (A), must detect the abnormal situation from the data collected by the sensors in the forest (collected by the monitor). We consider as abnormal situation all values of temperature (T) higher than a factor α, to the previous β temperatures (TM) in this zone. Also, it is considered an abnormal situation any image (I) recognized, which contains flammable products, smoke, flames, etc.

$$S_{abnormal} = \begin{cases} true, & if\ (T_c \geq \alpha\, TM_{c-1} \ldots AND\ T_c \geq \alpha TM_{c-\beta})\ or \\ & (I_c \subseteq [flammable\ products, smoke, flames]) \\ false, & otherwise \end{cases}$$

Also, the analyzer can diagnose future situations in the environment (IoT platform). In our case study, we distinguish 5 states of the IoT platform, based on the next thresholds (see Figure 4): H1 for unloaded, H2 for unloading, H3 for loading, H4 for loaded, and H5 for abnormal.

$$System_{status} = \begin{cases} \text{unloaded}, & if \ \text{values}[\omega_{\text{predicted}}, \text{QoS}_{\text{predicted}}, \ \text{S}_{\text{abnormal}}] \leq \ \text{H}_2 \\ \text{unloading}, & if \ \text{H}_2 \ \leq \ \text{values}[\omega_{\text{predicted}}, \text{QoS}_{\text{predicted}}, \ \text{S}_{\text{abnormal}}] \ \leq \ \text{H}_3 \\ \text{loading}, & if \ \text{H}_3 \ \leq \ \text{values}[\omega_{\text{predicted}}, \text{QoS}_{\text{predicted}}, \ \text{S}_{\text{abnormal}}] \ \leq \ \text{H}_4 \\ \text{loaded}, & if \ \text{H}_4 \ \leq \ \text{values}[\omega_{\text{predicted}}, \text{QoS}_{\text{predicted}}, \ \text{S}_{\text{abnormal}}] \leq \text{H}_5 \\ \text{abnormal}, & if \ \ \text{H}_5 \leq \ \text{values}[\omega_{\text{predicted}}, \text{QoS}_{\text{predicted}}, \ \text{S}_{\text{abnormal}}] \end{cases}$$

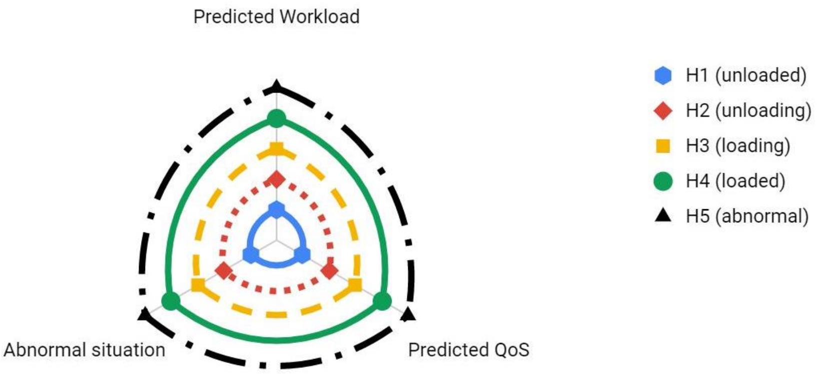

Fig. 4: States of the IoT Platform

- The Planner defines the plans, according to the diagnosed states. The planner determines:
  - ◆ The QoS mechanisms. In our case study, we consider the next QoS mechanisms: Scale in/out mechanisms and Traffic management mechanisms (xNF of traffic prioritization, xNF of Image Compression/Decompression).
  - ◆ The fault tolerance mechanisms to be implemented, in particular, for the resilience of gateways, likely to catch fire, and to be lost in the fire.
- The Executor (E) implements the generated plans. The implementation consists of the deployment of redundant gateways and QoS mechanisms.

At the level of the autonomic cycle, the specific DA tasks required in this context are described in Table 3.

Table 3. Data analysis tasks in the first case study

Data Analysis Task			Input	Output	Description
Monitor	1	Context Variables	CPU/MEM	Workload	This task calculates the load of each node, from the data recovered from the server and gateway log files. The metrics are: processor [CPU], and memory [ram] utilization.
	2	Wildfire detection App. QoS Variables (a)	Latency Bandwidth	QoS	This task calculates the overall QoS of the Wildfire detection application, from the data recovered in the application's log files. The metric is the latency [*Lat*].
		Wildfire detection App. Data (b)	Sensors Values	Abnormal situation	This task detects anomalies from data collected by the temperature and image sensors.
Analyze	3	Node-level Prediction model	Workload	Predicted Workload	From the Workload compute in task 1, this task predicts the future load at $\text{t} = t + \varepsilon$, of each node.
	4	App-level Prediction model	QoS	Predicted QoS	From QoS compute in task 2a, this task predicts the future QoS at $\text{t} = t + \varepsilon$, of the application.
	5	Classification model	Predicted Workload, Predicted QoS,	System future state	This task consists of identifying the future state of the system, from the predictions of tasks 2b, 3 and 4. System state: loading; loaded; unloading; unloaded; abnormal.

			Abnormal situation		
Plan	6	Classification System	System future state	Plan	This task consists in proposing action plans corresponding to the identified state of the system. $unloaded \rightarrow P_A$ $unloading \rightarrow P_B$ $loading \rightarrow P_C$ $loaded \rightarrow P_D$ $abnormal \rightarrow P_E$
Execute	7	Scheduler	Plan	Action workflow	This task executes the plan in the appropriate execution sequencing. For example, in P_C, the Workflow associated is**: Step 1: Add Traffic Prioritization Step 2: Add Compression and Decompression Step 3: Traffic redirection

**Some examples of plans are:

$P_A = [Remove\ Traffic\ Prioritization]$

$P_B = [Scale\ out\ Gateways, Scale - out\ Server, Remove\ Traffic\ Load\ balancing]$

$P_C = [Add\ Traffic\ Prioritization, Add\ Compression, Add\ Decompression, Traffic\ redirection]$

$P_D = [Scale\ in\ Gateways, Scale\ in\ Server, Add\ a\ Traffic\ Load\ balancing]$

$P_E = [Add\ Resilient\ Gateways, Scale\ in\ Server, Add\ a\ Traffic\ Load\ balancing]$

5.2 Case Study in the Tactile Internet

Description of the context

Tactile Internet Architecture is based on the haptic sense (sense of touch) idea. Differing from the auditory and visual senses, the haptic sense is bilateral, i.e. a touch is sensed by imposing a motion on an environment and feeling the environment by a distortion or reaction force. Haptic communications must provide this additional dimension over traditional audiovisual communication, for truly immersive steering and control in remote environments. According to [28], one of the key functionalities of the 5G core network relevant to the Tactile Internet, is the Dynamic application-aware QoS provisioning. For a real implementation of tactile internet, it is necessary an Autonomic Cycle for QoS/QoE provisioning.

Description of the platform

The end-to-end architecture for the Tactile Internet can be split into three contexts (see Fig 5): the master domain, the network domain, and the slave domain. The master domain normally has a human (operator) and a human-system interface (HSI), which is a haptic device (master robot) that transforms the human input to haptic input. The haptic device enables a user to touch, feel, and manipulate objects in real and virtual environments, and controls the operation of the slave domain. The network domain is the medium for bilateral communication between the master domain and the slave domain, and allows the immersion of the operator in the remote environment. The slave domain has a teleoperator (slave robot) controlled by the master domain using command signals. The teleoperator allows interaction with the objects in the remote environment.

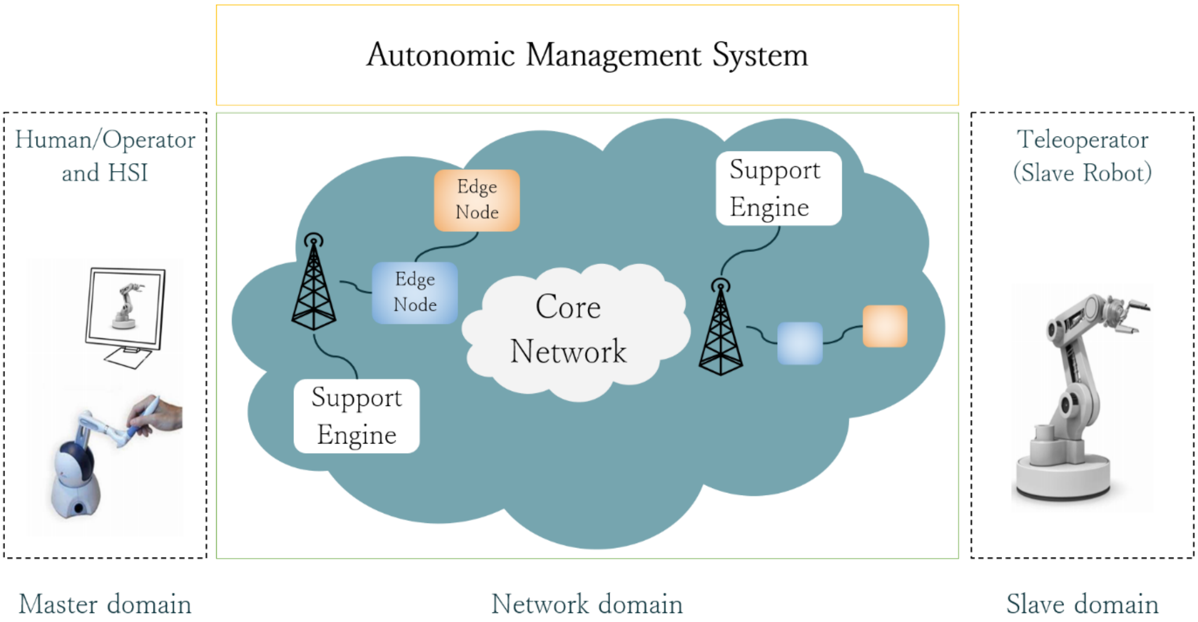

Fig. 5: Tactile Internet Platform.

Instantiation of the proposed AMS

- The Monitor is used for the collection and calculation of data. The collected data are the end-to-end delay (the delay between a user performing an action and when he perceives it), and the resource consumption in every node (CPU/MEM/queue sizes), from the environment; and the commands from the operators and the haptic feedback from teleoperator. Then, these data are used to calculate the Nodes Workload (ω) to characterize the environment, and the Tactile QoS/QoE (QoS) to characterize the tele-application
- The Analyzer must detect an abnormal situation, with respect to the data exchanged between the operator and the teleoperator. We consider an abnormal situation when the delay (D) has high values. It also diagnoses the future state of the environment (Haptic System). In our case study, we distinguish 4 states of the system, based on the next thresholds (see Fig 6): H1 for overprovisioned, H2 for normal, H3 for unprovisioned, H4 for abnormal.

$$System_{status} = \begin{cases} \text{overprovisioned}, & if \ \text{values}[\omega_{predicted}, QoS_{predicted}, \tau] \leq H_2 \\ normal, & if \ H_2 \leq \text{values}[\omega_{predicted}, QoS_{predicted}, \tau] \leq H_3 \\ \text{unprovisioned}, & if \ H_3 \leq \text{values}[\omega_{predicted}, QoS_{predicted}, \tau] \leq H_4 \\ \text{abnormal}, & if \ H_4 \leq \text{values}[\omega_{predicted}, QoS_{predicted}, \tau] \end{cases}$$

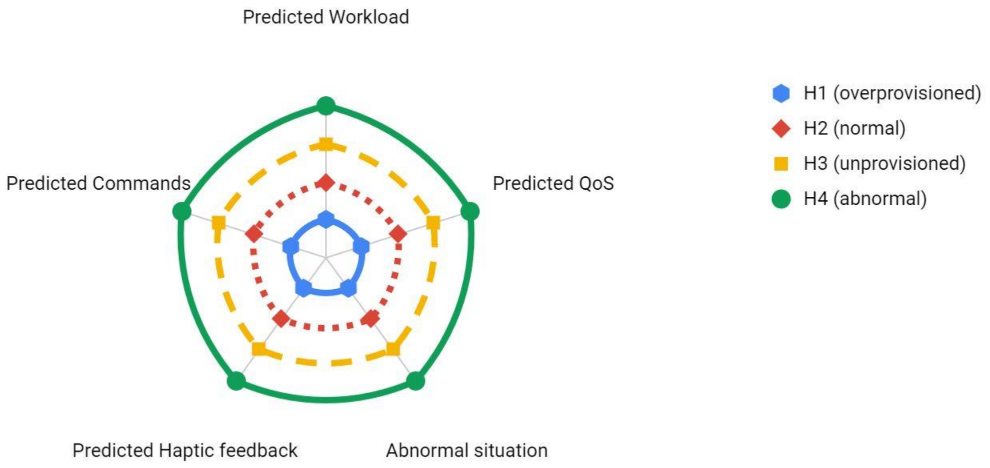

Fig. 6: States of the Tactile Internet Platform

- The Planner defines the plans, according to the diagnosed states. The plans are composed of the QoS/QoE mechanisms. Let's consider in this case study scale in/out mechanisms and Traffic management mechanisms (Multiplexing xNF and xNF of data Compression/Decompression). Also, it determines the fault tolerance mechanisms to implement, especially for the resilience of the support engine at the edge of the system.
- The Executor (E): in this case study, it is the implementation of the generated plans. The implementation is the deployment of redundant "Support Engines" and QoS/QoE mechanisms.

At the level of the autonomic cycle, the specific DA tasks required in this context are described in Table 4.

Table 4. Data analysis tasks in the second case study

Data Analysis Task			Input	Output	Description
Monitor	1	Context Variables	CPU/MEM	Workload	This task calculates the load of each edge node from the real-time measurement of the CPU [CPU], and Memory [ram] consumption.
	2	Tactile QoS Variables (a)	End-to-End Delay	QoS	This task calculates the overall QoS/QoE of the teleoperation application, from the real-time measure of the latency [*Lat*].
		Tactile Data (b)	Commands Haptic feedback	Abnormal situation	This task detects anomalies from the data exchanged between the operator and the teleoperator
Analyse	3	Node-level Prediction model	Workload	Predicted Workload	From the Workload compute in task 1, this task predicts the future load at $t = t + \varepsilon$, of each node.
	4	Tactile QoS Prediction model	QoS	Predicted QoS	From QoS compute in task 2a, this task predicts the future QoS at $t = t + x$, of the application.
	5	Haptic Prediction model (a)	Haptic feedback	Predicted Haptic feedback	From Haptic feedback monitored in task 2b, this task predicts the haptic feedback at $t = t + \varepsilon$
		Command Prediction model (b)	Commands	Predicted Commands	From the Commands monitored in task 2b, this task predicts the future command at $t = t + \varepsilon$.

	6	Classification model	Predicted Variables and Abnormal situation	System future state	This task consists of identifying the future state of the system, from the predictions of tasks 2, 3, 4 and 5. System status: normal; unprovisioned; passive; abnormal.
Plan	7	Classification System	System future state	Plan	This task consists in proposing action plans corresponding to the identified state of the system. overprovisioned → P_A Normal → P_B unprovisioned → P_C abnormal → P_D
Execute	8	Scheduler	Plan	Action workflow	This task executes the plan in the appropriate execution sequencing. For example, in P_C, the Workflow associated is**: Step 1: Scale Support Engine Step 2: Scale Edge Node Step 3: Add Compression Step 4: Add Decompression

**Some examples of plans are:

$P_A = [Scale\ out\ Edge\ Nodes, Scale\ out\ Support\ Engine\ , Remove\ Compression, Add\ Multiplexage]$

$P_B\ = [\emptyset]$

$P_C\ = [Scale\ in\ Edge\ Nodes, Scale\ in\ Support\ Engine\ , Add\ Compression, Add\ Decompression]$

$P_D\ = [Freeze\ Teleoperator, Reset\ Teleoperator]$

6. Discussions

The case studies show the versatility of our AMS to provide QoS/QoE services in different contexts. We are going to compare our AMS with several works, based on the next criteria: C1) They are data-driven approaches? C2) They can be used in different contexts (IoT, Tactile Internet)? C3) They can optimize QoS and QoE simultaneously? C4) They allow an autonomous self-configuration?

Table 5. Data analysis tasks in the second case study

Works	Criteria			
	C1	C2	C3	C4
[5, 6, 9]				x
[7, 11]	x			x
Our AMS	x	x	x	x

In general, there are some IoT AMS that use the data of the context in order to make decisions about the QoS mechanisms to deploy in a system [3, 11]. Also, some works consider the idea of an autonomous self-configuration [5, 6, 7, 11]. However, our paper is the only one where is used the AMS in different contexts, additionally, to reach QoE / QoS requirements. Also, our model can be extended with more autonomic cycles with other goals: monitor the IoT/Tactile Internet Platforms for fault tolerance, or provide "time-slicing" for applications with hard real-time requirements, among other things.

7. Conclusions

In this paper, we have proposed an architecture of an AMS for the management of IoT platforms, which can be used on other platforms, like Tactile Internet. This AMS is based on a set of data analysis tasks, which must be adequate to the context of applications. This set of tasks composes a cycle autonomous, which works in order to reach a given goal. Particularly, in this paper is defined an autonomic cycle in order to provide the QoS/QoE required by the applications.

We first introduced a general architecture of the AMS, for the management of IoT Platforms. Then, we have defined the set of data analysis tasks required by the AMS, based on the MAPE+K principle. Then, we have shown its utilization in different contexts, defining the instantiation of the tasks, the information required of the context, and the plans deployed by our AMS for different contexts, SDN and non-SDN network environments. It is the only work that is based on data-driven models, that can be used in different contexts (IoT, Tactile Internet), in order to reach different goals (e.g., optimize QoS and QoE requirements simultaneously), via an autonomous self-configuration.

Next works must define new autonomic cycles with other important goals in this context. For example, the management of "time-slicing" is a very important problem in the context of the Tactile Internet, and it is possible to propose an autonomic cycle for this goal, in the context of our architecture. An important challenger is the implementation of this architecture in a real scenario, particularly, to evaluate the cost of deployment of the different machine learning methods defined in each data analysis task. For that, it will be necessary to define efficient machine learning algorithms, like has been proposed recently in [29], adequate IoT entities, network functions in the correct place (virtual or not), among other things. It will be studied in the next works.

References

[1] OneM2M, "Technical Specification TS-0002-V2.7.1: Requirements,"vol. 1, pp. 1–24, 2016.

[2] J. Aguilar, M. Cerrada, G. Mousalli, F. Rivas, F. Hidrobo A Multiagent Model for Intelligent Distributed Control Systems. Lecture Notes in Computer Science, vol 3681, pp 191-197, 2005.

[3] J. Aguilar, O. Buendia J. Cordero "Specification of the Autonomic Cycles of Learning Analytic Tasks for a Smart Classroom", Journal of Educational Computing Research, vol. 56, no. 6, pp. 866-891, 2018.

[4] J. Vizcarrondo, J. Aguilar, E. Exposito, A. Subias "MAPE-K as a Service-oriented Architecture", *IEEE Latinoamerica Transactions,* vol. 15, no. 6, pp. 1163-1175, 2017.

[5] A. Agirre, et al. "QoS aware middleware support for dynamically reconfigurable component based IoT applications," International Journal of Distributed Sensor Networks, vol. 12, no. 4, 2016.

[6] Y. Banouar, S. Reddad, C. Diop, C. Chassot "Towards Autonomic Middleware-Level Management of QoS for IoT Applications", Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 169. 2015.

[7] Y. Banouar. “Gestion autonomique de la QoS au niveau middleware dans l’IoT. Réseaux et télécommunications”. PhD Thesis, Université Paul Sabatier - Toulouse III, 2017.

[8] Y. Banouar, T. Monteil and C. Chassot, "Analytical model for adaptive QoS management at the Middleware level in IoT," Proc. IEEE Symposium on Computers and Communications (ISCC), Heraklion, pp. 1201-1208, 2017.

[9] C. Ouedraogo, S. Medjiah, C. Chassot, K. Drira, “Enhancing Middleware-based IoT Applications through Run-Time Pluggable QoS Management Mechanisms. Application to a oneM2M compliant IoT Middleware”, Procedia Computer Science, vol 130, pp. 619-627, 2018.

[10] C. Ouedraogo, S. Medjiah, C. Chassot. “A Modular Framework for Dynamic QoS Management at the Middleware Level of the IoT: Application to a OneM2M Compliant IoT Platform”. Proc. IEEE International Conference on Communications, 2018.

[11] S. Kim, “Learning-Based QoS Control Algorithms for Next Generation Internet of Things”, Mobile Information Systems, vol. 2015, 2015

[12] M. Elkhodr, S. Shahrestani, H. Cheung, “A middleware for the Internet of Things”, International Journal of Computer Networks & Communications, vol.8, no.2, pp. 159-178, 2016.

[13] B. Gomes, L. Muniz, F. da Silva, D. Dos Santos, R. Lopes, L Coutinho, M. Endler, “A Middleware with Comprehensive Quality of Context Support for the Internet of Things Applications”. Sensors, vol. 17, no. 12, 2017

[14] M. da Cruz, J. Rodrigues, J. Al-Muhtadi, V. Korotaev, V. de Albuquerque, "A Reference Model for Internet of Things Middleware," IEEE Internet of Things Journal, vol. 5, no. 2, pp. 871-883, 2018.

[15] S. Akkermans, S. Peros, N. Small, W. Joosen, D. Hughe, “Supporting IoT Application Middleware on Edge and Cloud Infrastructures”, Proc ZEUS Workshop, 2018.

[16] W. Heinzelman, et al., “Middleware to Support Sensor Network Applications,” Network, IEEE, vol. 18, no. 1, pp. 6-14, 2014.

[17] S. Yu, Z. Huang, C. Shih, K. Lin, J. Hsu, “QoS Oriented Sensor Selection in IoT System,” Proc. IEEE International Conference on Internet of Things, pp. 201-206, 2014.

[18] J. Silva, et al., “PRISMA: A publish-subscribe and resource-oriented middleware for wireless sensor networks,” Proc. Tenth Advanced International Conference on Telecommunications, pp. 8797, 2014.

[19] Y. Banouar, C. Ouedraogo, C. Chassot, A. Zyane. “QoS management mechanisms for Enhanced Living Environments in IoT” Proc. IFIP/IEEE Symposium on Integrated Network and Service Management, 2017

[20] T. Rausch, S. Nastic, S. Dustdar, “EMMA: Distributed QoS-Aware MQTT Middleware for Edge Computing Applications”. Proc. IEEE International Conference on Cloud Engineering, pp. 191-197, 2018.

[21] N. Hua, N. Yu and Y. Guo, "Research on Service Oriented and Middleware Based Active QoS Infrastructure of Wireless Sensor Networks," Proc. 10th International Symposium on Pervasive Systems, Algorithms, and Networks, Kaohsiung, pp. 208-213, 2009.

[22] C. Ouedraogo, E. Bonfoh, S. Medjiah, C. Chassot, S. Yangui, "A Prototype for Dynamic Provisioning of QoS-oriented Virtualized Network Functions in the Internet of Things."

Proc. 4th IEEE Conference on Network Softwarization and Workshops (NetSoft). pp. 323-325, 2018.

[23] S. Medjiah, C. Chassot, "On the enhancement of non-functional requirements for cloud-assisted middleware-based IoT and other applications," Proc. 2nd Workshop on Adaptive Service-oriented and Cloud Applications, 2017.

[24] Open Networking Foundation, "OpenFlow Switch Specification Version 1.5.1", 2015.

[25] IBM, "Autonomic Computing: IBM's Perspective on the State of Information Technology," 2001

[26] O. Kephart and D. M. Chess, "The vision of autonomic computing," January 2003, Computer, v.36 n.1, p.41-50.

[27] National Interagency Coordination Center. "Wildland Fire Summary and Statistics", Annual Report 2017, Available: https://www.predictiveservices.nifc.gov/intelligence/2017_statssumm/annual_report_2017.pdf

[28] A.Aijaz, M. Simsek, M. Dohler, G. Fettweis. " Shaping 5G for the Tactile Internet". 5G Mobile Communications (W. Xiang et al. (eds.)), Springer, pp. 677–691. 2016.

[29] L. Morales, C. Ouedraogo, , J. Aguilar, C. Chassot, S. Medjiah, K. Drira, "Experimental Comparison of the Diagnostic Capabilities of Classification and Clustering Algorithms for the QoS Management in an Autonomic IoT Platform", Service Oriented Computing and Applications, vol. 13, pp 199–219, 2019.

[30] J. Vizcarrondo, J. Aguilar, E. Exposito, A. Subias, "ARMISCOM: Autonomic Reflective MIddleware for management Service COMposition". Proc 4th Global Information Infrastructure and Networking Symposium, IEEE Communication Society, 2012.

[31] H. Mahini, R. Hajisheykhi, M. Shahini, "QoS and traffic aware greedy resource allocation in foggy internet of things", International Journal of Communication Networks and Distributed Systems, vol. 25, no. 1, 2020.

[32] V. Pandi, P. Perumal, B. Balusamy, M. Karuppiah, "A Novel Performance Enhancing Task Scheduling Algorithm for Cloud-Based E-Health Environment". International Journal of E-Health and Medical Communications, vol. 10, no. 2, pp. 102-117, 2019.

[33] P. Ray, D. Dash, D. De. "Internet of things-based real-time model study on e-healthcare: Device, message service and dew computing", Computer Networks, vol 149, pp 226-239, 2019.

[34] J. Yuan, Y. He "Research on intelligent home design of internet of things based on ZigBee", International Journal of Communication Networks and Distributed Systems, vol. 26, no. 3, 2021.

[35] M. Serhani, H. El-Kassabi, K. Shuaib, A. Navaz, B. Benatallah, A. Beheshti, "Self-adapting cloud services orchestration for fulfilling intensive sensory data-driven IoT workflows", Future Generation Computer Systems, vol. 108, pp. 583-597, 2020.

[36] D. Pianini, R. Casadei, M. Viroli, A. Natali, Partitioned integration and coordination via the self-organising coordination regions pattern, Future Generation Computer Systems, vol 114, pp. 44-68, 2021.

[37] A. Yassine, S. Singh, M. Hossain, M. Ghulam, "IoT big data analytics for smart homes

with fog and cloud computing”, Future Generation Computer Systems, vol. 91, pp. 563-573, 2019.

[38] M. Pereira, A. Gonçalves, M. Ribeiro, “A conceptual model for quality of experience management to provide context-aware eHealth services”, Future Generation Computer Systems, vol 101, pp. 1041-1061, 2019.

[39] G. Aloi, G. Fortino, R. Gravina, P. Pace and C. Savaglio, "Simulation-Driven Platform for Edge-Based AAL Systems", IEEE Journal on Selected Areas in Communications, vol. 39, no. 2, pp. 446-462, 2021.

[40] C. Ouedraogo, S. Medjiah, C. Chassot, K. Drira, J. Aguilar, "A Cost-Effective Approach for End-to-End QoS Management in NFV-Enabled IoT Platforms", IEEE Internet of Things Journal, vol. 8, no. 5, pp. 3885-3903, 2021.

[41] A. Alameen, A. Gupta, “Enhancement of network lifetime of cloud-assisted internet of things: new contribution of deer hunting and particle swarm optimization”, International Journal of Communication Networks and Distributed Systems, vol. 26, no. 3, 2021.

[42] M. Sanchez, E. Exposito, J. Aguilar, “Autonomic computing in manufacturing process coordination in industry 4.0 context”, Journal of Industrial Information Integration, vol. 19, 2020.

[43] M. Sánchez, E. Exposito, J. Aguilar, Implementing self-* autonomic properties in self-coordinated manufacturing processes for the Industry 4.0 context, Computers in Industry, vol. 121, 2020.

[44] A. Braten, F. Kraemer, D. Palma, "Autonomous IoT Device Management Systems: Structured Review and Generalized Cognitive Model", IEEE Internet of Things Journal, vol. 8, no. 6, pp. 4275-4290, 2021.

[45] G. Fortino, C. Savaglio, G. Spezzano and M. Zhou, "Internet of Things as System of Systems: A Review of Methodologies, Frameworks, Platforms, and Tools", IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 51, no. 1, pp. 223-236, 2021.

[46] A. Mutlag, M. Ghani, N. Arunkumar, M. Mohammed, O. Mohd, Enabling technologies for fog computing in healthcare IoT systems, Future Generation Computer Systems, vol 90, pp 62-78, 2019.

[47] R. Patibandla, V. Narayana, A. Gopi, “Autonomic Computing on Cloud Computing Using Architecture Adoption Models: An Empirical Review”. In Autonomic Computing in Cloud Resource Management in Industry 4.0, Innovations in Communication and Computing, Spring, pp. 195-212, 2021.

[48] J. Aguilar, A. Garces-Jimenez, M. R-Moreno, R. García, “A systematic literature review on the use of artificial intelligence in energy self-management in smart buildings”, Renewable and Sustainable Energy Reviews, vol. 151, 2021.