Edge Computing in IoT ecosystems for UAV-enabled Early Fire Detection

Nikos Kalatzis, Marios Avgeris, Dimitris Dechouniotis, Konstantinos Papadakis-Vlachopapadopoulos, Ioanna Roussaki, Symeon Papavassiliou
National Technical University of Athens
Athens, Greece
{ nikosk@cn, mavgeris@netmode, ddechou@netmode, cpapad@netmode, ioanna.roussaki@cn, papavass@mail }.ntua.gr

***Abstract*—Unmanned Aerial Vehicles (UAV) facilitate the development of Internet of Things (IoT) ecosystems for smart city and smart environment applications. This paper proposes the adoption of Edge and Fog computing principles to the UAV based forest fire detection application domain through a hierarchical architecture. This three-layer ecosystem combines the powerful resources of cloud computing, the rich resources of fog computing and the sensing capabilities of the UAVs. These layers efficiently cooperate to address the key challenges imposed by the early forest fire detection use case. Initial experimental evaluations measuring crucial performance metrics indicate that critical resources, such as CPU/RAM, battery life and network resources, can be efficiently managed and dynamically allocated by the proposed approach.**

Keywords*— IoT, edge computing, fog computing, cloud computing, UAV, fire detection, dynamic resource allocation

I. Introduction

As stated in the 17th issue of the annual EU report on forest fires concerning Europe, Middle East and North Africa [1], year 2017 will be remembered as one of the most devastating in Europe since records began. Nearly 700000 ha of land were burned in EU and there is a very heavy toll of human lives in Southern Europe. In the same report, it is indicated that due to the climate changes, it is expected that extreme weather conditions such as extended heat waves, drought, and strong winds will affect many of Europe's forests more frequently and more severely. To this end, forest-fires fighting plays a critical role in the protection and preservation of natural resources given the deteriorating natural danger imposed by wild mega fires. A crucial factor for minimizing the damage is the early detection and suppression of the forest fires. Thus, massive efforts have been put into monitoring, detecting and rapidly extinguishing them before they become uncontrollable [2].

The synergistic combination of recent technological developments from areas such as cloud computing, sensor networks, internet of things, and big data analytics improves our ways of living in various societal domains. Over the last years, forest fire fighting technology enabling smart computing features on Unmanned Aerial Vehicles (UAVs) has shown significant progress making the deployment of small-size UAVs for forest fire detection a natural and increasingly realistic option [3]. UAVs are relatively inexpensive, easily manoeuvrable, can cover various terrain types under different weather settings, both at day and night and most importantly, their missions can be achieved autonomously with minimal or even with zero human involvement. UAVs equipped with remote sensing and data communication facilities demonstrate excellent potential for monitoring, detecting, and fighting forest fires. On the other hand, the potential advantages of UAVs, depend on many factors, such as aircraft type, sensor types, mission objectives, and the current regulatory requirements in the application domain the UAV operates in [4]. Specifically for the fire-fighting domain, UAV technology still faces various obstacles that need to be confronted in order to be applied in fully operational environments. One of these barriers is the scarcity of available energy and processing resources on UAV platforms. Aerial monitoring of large fields and forests during dry spells to reduce the risk of wildfires requires increased energy resources for UAVs to prolong their mission's endurance, which is very difficult to secure.

The challenge of optimised resource utilisation on resource-constraint devices has also been tackled in the setting of IoT computing. In order to support the requirements of real-time and latency-sensitive applications engaging largely geo-distributed IoT devices/sensors, the emerging "Edge computing" paradigm has been introduced [5]. Edge computing establishes a hierarchy of intelligent processing elements between devices and gateways, as well as between the cloud and endpoint devices, to meet the processing challenges in a scalable, context-aware and interoperable manner. This paper presents an innovative approach of employing the IoT Edge computing paradigm in support of early detection of forest fires.

The rest of the paper is structured as follows. In Section II, the current state of the art is presented, while in Section III the key requirements of this domain are discussed along with the proposed high-level architecture. Section IV presents the formal problem model of minimising the time to fire detection decision given the underlying resource constraints. Section V elaborates on the design of the emergency detection experiments that aim to evaluate the proposed architecture. Initial evaluation findings are presented in Section VI, while conclusions and future plans are exposed in Section VII.

II. UAV-based Emergency Detection in the Literature

One of the first approaches, where UAVs are utilised in support of forest-fire detection and fighting is presented in [6].

The paper elaborates on the potential of UAV applications distinguishing between before/during/after the fire settings. The authors present how UAVs can be utilised, before a fire occurs (for monitoring the vegetation and estimating hydric stress and respective risk index), but also for forest-fire early detection, confirmation, localisation and monitoring. Finally, after the fire, the UAVs are also useful for measuring the size of burnt areas and evaluating the overall fire consequences. Authors propose a cooperative perception system featuring infrared, visual camera, and fire detectors mounted on various UAV types. A fire detection and dynamic tracking algorithm is presented in [7] that is based on fire images segmentation analysis in different colour spaces in order to extract fire-pixels by making use of chromatic features of fire. The proposed algorithm includes techniques such as median filtering, colour space conversion, Otsu threshold segmentation, and other morphological operations. A similar approach is presented in [8] where a colour index, called the Forest Fire Detection Index is introduced, in order to evaluate images captured by UAVs. The index is based on methods for vegetation classification and has been adapted to detect the tonalities of flames and smoke. An interesting aspect of this approach is that only low cost cameras and sensors are utilised in an attempt to make the UAV utilisation cost-benefit relationship better than this of the traditional existing forest-fire detection systems. In [9], a decentralized control algorithm is presented designed for UAV fleets that can autonomously and actively track the fire spreading boundaries in a distributed manner. The approach also supports the exchange of information among the UAVs, ensures that safe distance is maintained in order to avoid in-flight collision and supports dynamic adjustment of captured image resolution on the border of the wildfire d. In [10], a detailed review of current progress in the field of UAV-based forest fire fighting technology is presented. Technologies suitable for UAV forest fire monitoring, detection, and fighting have been reviewed, including ICT for fire detection, diagnosis and prognosis, image vibration elimination, and cooperative control of UAVs. One of the significant outcomes of this review is that it is crucial to use multiple sources of information at several locations with a high rate of updates for efficient forest fire monitoring, detection, and even fighting.

Towards this scope, recent approaches attempt to combine innovative concepts lying in the Internet of Things domain in order to improve the efficiency of UAVs' utilisation. For example, the Internet of Flying Things [11] focuses on services and applications provided to mobile users using airborne computing infrastructure. It presents a novel application where UAVs act as "Data Mules" that provide services after natural disasters to rural regions and address challenges related with big data analytics, situation-awareness and incremental extensibility of the computing infrastructure. In [12], a UAV based IoT platform is proposed for face recognition from large altitudes. The respective experiments indicate that the energy consumption of UAVs significantly decreases if they offload short videos to a local edge server without any performance degradation of the recognition a. In [13], a comprehensive survey is presented with regards to the potential of UAV-based value-added IoT services. The authors claim that due to its ubiquitous usability, UAV's will play an important role in the Internet of Things (IoT) vision, and may even become key IoT enablers in numerous application domains (e.g., emergency management, precision agriculture, forest fire monitoring, architecture surveillance, goods transportations). However, deploying UAVs for such purposes envisioned cannot be realised before various relevant challenges are sufficiently addressed, such as the need for real-time data processing, meeting regulatory/standardization matters, as well as public safety & privacy protection concerns for people on the ground.

III. Requirements & Architecture

The state of the art review above indicates that there are still many open challenges that need to be addressed in order to allow for usage of UAVs in real operational forest-fire detection missions. On the other hand, existing data processing mechanisms proved to be efficient in IoT ecosystems can well be adapted to serve the UAV-based fire detection domain. Studying the outcomes of systematic state of the art reviews, such as [13] and [10], as well as experts' reports in the emergency handling field [14], it is concluded that the IoT paradigm can provide efficient solutions regarding the following key requirements in UAV-based fire detection:

Early incident detection and identification: It is of high value to detect and suppress wild fires prior to reaching an uncontrolled state. According to experts, a typical wildfire tends to double in size every few minutes while in high-wind conditions or extremely dry conditions, the rate of growth can be much greater [15]. Detecting fire incidents using sensors (e.g., visual sensor networks) and GPS systems to determine the location of fire is a complicated task and high detection probability, low false alarm rates, and enhanced adaptive capabilities in various environmental conditions must be ensured.

Optimised utilisation of resources: Current multirotor UAVs demonstrate increased agility and fault tolerance compared to fixed wing vehicles. However, multirotors are based on electric power sources with an average endurance of 10 to 50 minutes [14]. Hence, it is significant to minimise the utilisation of sensing, communication & data processing resources to reduce energy consumption in favour of increased mission endurance, without endangering the early and accurate detection of fire.

Centralised and intelligent decision making: Large area monitoring and fire detection are complicated tasks that require a decision support system able to process information from vast numbers of deployed sensors and at the same time to avoid false alarms or failing to detect critical incidents. Coordination and control of the overall effort requires all necessary information to be available in a centralised manner without inducing overloads in computation and communication resource usage. Massive deployments of sensing devices may result in congestion problems when systems generate simultaneous data at periodic time intervals or due to the existence of events, such as the detection of fire in a forest.

Conforming with regulations: The efficient and safe deployment of UAV technology must be driven by clear regulatory rules that will eliminate the risk ofphysical collision, arbitrary path planning, or endangering the privacy of people.

To address the aforementioned requirements, a distributed agent-based layered architecture is proposed that aims to enable UAV-based fire detection suitably exploiting sophisticated features originating from the IoT ecosystem. This approach mainly focuses on evaluating the usage/consumption of UAV processing, energy and communication resources, aiming for early and accurate fire incident detection within the areas under surveillance. An illustration of the proposed architecture is provided in Fig. 1.

One of the key features of the proposed solution is the establishment of low-latency access to servers at the edge of the network. Based on this edge computing approach, the resource-constrained UAVs and their sensors, are able to selectively offload energy consuming tasks, such as image recognition, to the dedicated services deployed on the respective fog server in proximity. More specifically, each fog server is equipped with the Agent version of the proposed software. The latter includes among others a hybrid Controller-Hypervisor (CH) component able to create, run, scale and stop application-specific virtual services. For this purpose, the Docker Platform is adopted, where flexible, lightweight Docker Containers facilitate the aforementioned services. Essential to this architecture is the Logging Service component (also implemented within the Agent), which is responsible for monitoring the volume of image processing requests, the network traffic and the server CPU/RAM resource availability and report back to the CH. Based on these data and exploiting novel control theory techniques, CH is able to calculate and ensure the proactive optimal allocation of the server's resources to the dockerized services, at each given moment. In this way, resource over (or under) provisioning is avoided and high availability at all times, on the fog server is achieved, while minimum delay in response times is guaranteed.

The second offloading decision takes place at the fog layer, where the data mentioned above are now available. For this purpose, an Image Classifier module is deployed in the dockerized service, aiming to detect potential emergency situations. Images captured by the UAVs are classified based on whether a forest-fire incident is detected or not. Then, only selected data (e.g., classification outcome) regarding the detected event are published to the Cloud Backend for further analysis. This controlled flow of data, in addition to optimizing bandwidth utilization for the links between fog and cloud nodes, allows for addressing security and privacy concerns; as potentially sensitive data are not disclosed to 3rd parties, as they are in principle stored within the premises where the Agent software is deployed. Even in the emergency event detection case, data published to the Cloud are limited in volume and stripped of any privacy-invasive information, as the entire procedure is supervised by the owning entity.

Finally, at the cloud layer, a central repository is deployed that receives notifications from the distributed gateways. The NGSI data model [16] and API has been utilised to ensure efficient and interoperable communication among the Agents and the centralised data repository. The Orion Context Broker, a FIWARE Generic Enabler that implements the NGSI API, has been utilised for synchronous and asynchronous management of information from the distributed nodes. In addition, a decision engine based on a semantic reasoner is deployed at the cloud that is able to combine data originating from various heterogeneous sources – not limited to the UAV-Gateways in order to extract additional knowledge in support of early detection of critical situations. The description of the individual components deployed at the fog node follows:

Controller-Hypervisor: Manipulates the Docker containers that implement the virtual services which are deployed to the Fog Servers. The CH component gathers workload & utilization data. Image Classification Service: Tensorflow [17] deep learning framework by Google is utilized for the classification process of images captured by the UAVs. This open source, deep learning framework has built-in libraries that have been leveraged to train a Deep Neural Network (DNN) able to recognize images containing events of interest (i.e., fire, smoke). The training process included the off-line, supervised feeding of the DNN with a large volume of such images. Incoming offloaded data from the edge are then accordingly classified into discrete categories, based on their estimated criticality. Instances of this service are deployed within the Docker containers. Containers were selected instead of Virtual Machines, as a means of virtualization, because of their overall lower overhead, smaller footprint and lightweight vertical scalability. Logging service: Monitors various systems parameters (e.g., CPU/storage/bandwidth utilization, time intervals, etc.) during the execution of the experiments. NGSI Client: Translates data from custom data models (e.g. UAV sensors) to NGSI data model. The translator processes both (near) real time streams of data or stored datasets in batch mode. In addition, provides synchronous and asynchronous communication mechanisms and at the same time ensures authentication and authorization with the Cloud components.

The description of the individual components deployed at the Cloud node follows:

Orion Context Broker: Maintains data received from Agents, supports pub/sub operations with the Agents and/or other eligible consumers. It ensures authentication and authorization. Decision-Making Service: This service processes data offloaded from the Fog Layer that have been classified above a threshold of criticality and thus potentially indicate an active emergency situation. A reasoning mechanism is responsible for inferring knowledge regarding such situations, based on the aggregated offloaded data, automatically. More specifically, an on-line Knowledge Base (KB) comprising the offloaded data constantly expands with inferred knowledge coming from this service. This 'extra' knowledge is a result of specific logical rules applied on the KB. These rules are merely a sequence of well-defined conditions that, when satisfied, denote the emergency situation level based on a predefined scale: low, medium, high.

IV. FORMAL PROBLEM STATEMENT

As already discussed, the fog computing layer is the intermediate layer between the edge and cloud computing layers and provides essential mechanisms for the transparent communication and resource allocation for the overall communication. As opposed to cloud computing, edge and fog computing resources are not abundant and thus, a dynamic resource allocation and scaling mechanism is essential to meet the requirement of time- or mission-critical IoT applications. In this section, a high level model of the optimal resource allocation problem is presented that allows for the ad-hoc allocation of available resources across the three layers.

In our studied problem setting, the fog layer includes $i = 1,2,\ldots,N$ nodes, each of which may serve $j = 1,2,\ldots,M_i$ edge nodes (in our use case, all established in resource light devices attached to UAVs). Each of the edge nodes is responsible for monitoring specific rural areas, to capture aerial images of the assigned areas and use these as input to decide about whether or not a fire emergency has occurred. The rate of the images captured by edge node e_{ij} corresponds to rate r_{ij} (in Bytes/sec) that will be used to determine the occurrence of potential fire emergency. Processing these captured images via suitable trained software requires p_{ij} processing resources. In case the image processing is performed locally on edge level, the time to decision is t_{ij}^E (that includes the time for the decision to be transmitted to the cloud), in case it is performed on fog level, the time to decision is t_{ij}^F (that includes the time for the image to be transmitted from the edge to a fog node and the time for the decision to be transmitted to the cloud) and in case the image processing is performed on cloud level, the time to decision is t_{ij}^C (that includes the time for the image to be transmitted from the edge node to the cloud). Of course, these parameters depend heavily not only on the rate r_{ij} of the data to be processed induced by edge node e_{ij}, but also on several additional parameters, such as the CPU/RAM properties of the nodes involved in each layer, on the telecom infrastructure and available network resources enabling the communication of nodes laying in different layers, etc.

In this framework, the problem of optimal resource allocation, aims to minimise the average time to decision z for all images captured by the UAVs. The respective objective function can thus be formulated as follows:

$$z = \frac{1}{\sum_{i=1}^{N} M_i} \cdot \sum_{i=1}^{N} \left[\sum_{j=1}^{M_i} \left(d_{ij}^E \cdot t_{ij}^E + d_{ij}^F \cdot t_{ij}^F + d_{ij}^C \cdot t_{ij}^C \right) \right]$$

where $d_{ij}^E = 1$, in case the image processing is performed locally on edge level by node e_{ij}; $d_{ij}^F = 1$, when the images captured by edge node e_{ij} are processed on fog level by node f_i; and $d_{ij}^C = 1$, in case the processing of images captured by edge node e_{ij} is performed on cloud level. There are several constraints that apply that are related to the resource availability at each layer. Thus, it is assumed that P_{ij}^M indicates the available processing resources of edge node e_{ij}, P_i^M indicates the available processing resources of fog node f_i, N_{ij}^M indicates the maximum data transmission rate that can be supported by the network infrastructure available to edge node e_{ij}, N_i^M indicates the maximum data transmission rate that can be supported by the network infrastructure available to fog node f_i, and E_{ij}^M indicates the battery life of edge node e_{ij}. It is also assumed that the decision regarding the allocation of resources will be performed regularly at time intervals of duration T. If finally, E_{ij}^E indicates the energy consumed at edge node e_{ij}, in case the image processing load is executed locally for duration T, E_{ij}^F indicates the energy consumed at edge node e_{ij}, in case the image processing load is executed on fog level for duration T (and the actual energy consumption on e_{ij} is due to the transmission of images to the respective fog node f_i), and E_{ij}^C indicates the energy consumed at edge node e_{ij}, in case the image processing load is executed on cloud level for duration T (and the actual energy consumption on e_{ij} is due to the transmission of images to the cloud), then the optimal resource allocation problem that aims to minimise the average time to decision can be formally stated as follows:

$$\min z = \frac{1}{\sum_{i=1}^{N} M_i} \cdot \sum_{i=1}^{N} \left[\sum_{j=1}^{M_i} \left(d_{ij}^E \cdot t_{ij}^E + d_{ij}^F \cdot t_{ij}^F + d_{ij}^C \cdot t_{ij}^C \right) \right]$$

subject to

$$0 \le d_{ij}^E \cdot p_{ij} \le P_{ij}^M, \forall i,j$$

$$0 \le \sum_{j=1}^{M_i} \left(d_{ij}^F \cdot p_{ij} \right) \le P_i^M, \forall i$$

$$0 \le d_{ij}^E \cdot r_{ij} \le N_{ij}^M, \forall i,j$$

$$0 \le \sum_{j=1}^{M_i} \left(d_{ij}^F \cdot r_{ij} \right) \le N_i^M, \forall i$$

$$0 \le d_{ij}^E \cdot E_{ij}^E + d_{ij}^F \cdot E_{ij}^F + d_{ij}^C \cdot E_{ij}^C \le E_{ij}^M, \forall i,j$$

$$d_{ij}^E + d_{ij}^F + d_{ij}^C = 1, \forall i,j$$

$$d_{ij}^E, d_{ij}^F, d_{ij}^C \in \{0,1\}$$

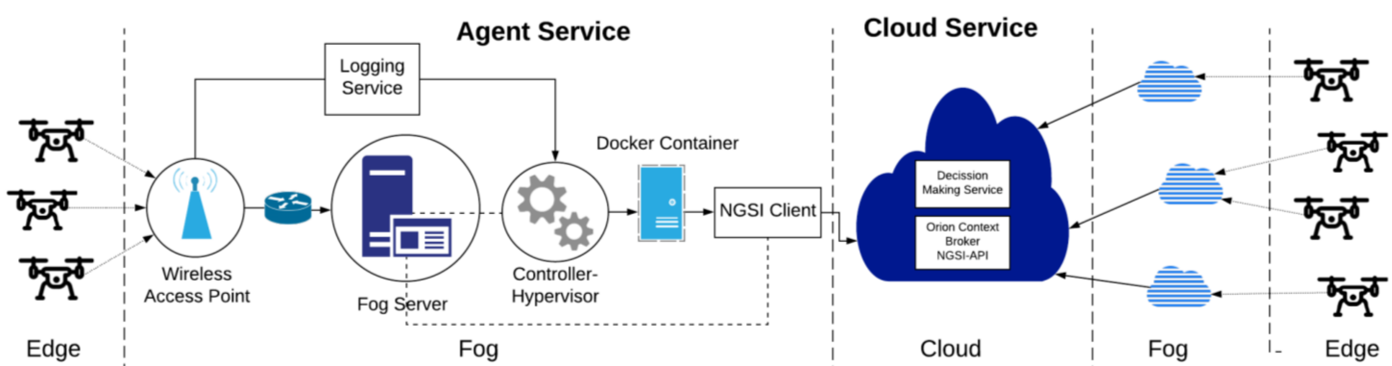

Fig. 1. High level architecture of the IoT ecosystem built for UAV-enabled early fire detection.

Of course, in order for the model above to be usable, the functions that calculate the values of t_{ij}^{E} , t_{ij}^{F} and t_{ij}^{C} given the available CPU, RAM, network and energy resources on all nodes involved, need to be specified as well.

V. Emergency Detection Demo Design

The experiment aims to employ varying numbers of UAVs, across several sites (testbeds) with the mission to monitor large rural areas, in order to detect potential fire incident as early as possible, and to support the initiation of a series of actions towards the handling of the emergency situation. This approach simulates the operation of a centralized command and control centre that is responsible for coordinating the utilization of sensing resources, evaluate detected emergency incidents and coordinate consequent actions. The UAV sensed data (e.g., visual and IR images, location, temperature, air quality) may either be transferred to the base stations or to a centralized Cloud service in order to be jointly processed. If a fire incident is detected, new mission directions are send to the UAV, e.g., reduce speed, increase sampling rate, enable additional sensors, increase rate of messages send to base station, update mission's waypoints - fly around the point of interest, etc. The new directions aim to collect more data from the area and to verify with higher accuracy that a fire incident is really taking place and to avoid the case of a false alarm. Higher sampling of images requires more processing resources, which should be automatically allocated on the Fog servers. The classification scores of the reasoning process on the images along with the location coordinates of the incident is forwarded to the Cloud service. The decision-making engine at the Cloud is expected to automatically detect how critical the emergency situation is based on a predefined scale (e.g., low, medium, high). The expected confidence of the reasoning outcome is higher and the result is more accurate, when richer information available is made available. In this approach, the following performance indicators are considered: faster identification of emergency situation, more accurate evaluation of situation, limited engagement of human resources. The demo realisation consists of the following three main phases:

1. The initial experiments aim to validate the proposed architecture implementation deployed on a single testbed using limited populations of sensors that are not yet deployed on UAVs. At this stage, no advanced intelligent offloading process is employed. This allows for specifying reference thresholds for the respective performance metrics that are subsequently used as baseline values to evaluate the results obtained when the proposed intelligent mechanisms are employed.
2. The second round of experiments gradually utilizes increased populations of data sources. These experiments focus on the evaluation and refinement of the semantic reasoning support both at fog as well as at cloud level. At this stage, the full-scale version of the experiments aim at engaging larger populations of UAVs, sensors, and data sources in general. These experiments primarily focus on the evaluation of the proposed intelligence distribution and resource allocation optimization mechanisms, as well as the collection and preparation of experimental data that enable the specification of the exact control function.
3. At the last round of experiments, numerous UAVs in several remote sites are engaged, allowing for the full-scale evaluation of the proposed offloading and optimisation mechanisms. It should be mentioned that the adoption of the NGSI open standards for modelling the datasets to be collected during the experiments enables their dissemination to the respective data portal platforms (e.g., CKAN), so that they are made available to the wider research community.

The second and third phase of the experiments will be realised within a federation of testbeds provided by the H2020 RAWFIE project. RAWFIE testbeds are part of the Future Internet Research Experimentation (FIRE+) initiative and aims at providing research facilities for Internet of Things (IoT) devices by integrating numerous testbeds of unmanned vehicles for research experimentation in vehicular, aerial and maritime environments. The RAWFIE platform supports experimenters with smart tools [18] to conduct and monitor experiments that involve UAVs along with mechanisms for safely controlling various vehicle populations deployed at various sites. In order to simulate the patrol over forest area, CATUAV (Spain) and CESA-DRONES (France) testbeds will be engaged.

The first phase of the implementation of the proposed architecture along with the planned evaluation has been completed. We evaluated the prototype implementation of our framework with respect to the average time elapsed between the event portrayal and its detection by the system, i.e. response time, the total energy consumption at the network edge, as well as the network traffic generated. For this purpose, three scenarios are described and examined, capturing the main strategies that can be employed regarding the node level of image processing (depicted in Fig. 2):

1. All images are processed on the UAV nodes and the Cloud nodes are notified only if an emergency situation is detected with high probability ("Edge Only").
2. All images once captured by the UAVs are forwarded to and subsequently processed exclusively on the Cloud for detecting emergencies. ("Cloud Only").
3. All images captured by the UAV nodes forwarded to the Fog node where they are subsequently processed. The Cloud is notified only if an emergency situation is detected with high probability ("Fog Only").

VI. Experimental Evaluation

In this section the setup of the experiments and the findings of the experiment execution are presented, along with the respective evaluation and comparison of obtained results. As aforementioned, this evaluation corresponds to the first phase of the proposed architecture implementation.

A. *Experimental Setup*

The basic hardware setup, used in all three aforementioned experiment scenarios, consists of three mobile clients, one wireless access point, one Fog server and one Cloud server based on the architecture described in Section III. To capture accurately the UAVs' plausible computation and networking capabilities, as well as their mission duration, three Raspberry Pis 3 Model B have been utilized equipped with quad core ARM Cortex – A53, 1.2GHz processors, 1GB RAM, Camera Modules v2 and 802.11ac Dual Band WiFi dongles, running on 10000mAh Power Banks; an ALIX.3 system board was used as the access point (AP). As Fog server a local desktop within the NETMODE testbed premises was engaged, equipped with an Intel i7 Quad core 2.7GHz processor and 8GB of available RAM. For the Cloud server a physical node from Imec's Virtual Wall testbed was reserved, equipped with 2x Quad core Intel E5520 (2.2GHz) CPU and 12GB RAM. This server's location was intentionally selected far from our premises to achieve an unbiased simulation of the Cloud. To precisely monitor and evaluate the real-time energy consumption of the Edge nodes, BOSS 220 Smart Plugs were installed on site.

With regards to the simulation configuration itself, each experiment scenario spanned for 15 minutes. The Raspberry Pis where powered by the BOSS 220 Smart Plugs and where connected via 802.11ac communication to the ALIX.3 board, which served as a local wireless and wired access point and an internet gateway. For the Edge Only scenario, all images were processed locally and sequentially on the Pis and the classification results were sent to the NGSI component on the cloud through the ALIX.3 board AP. This approach resulted to an average rate of three processed images per minute. In the Cloud Only scenario, all images were sent for processing to the remote server through the ALIX.3 board AP, image processing was executed and the results of the classification were directly provided to the NGSI through the loopback interface. For the Fog Only Scenario, all images were sent to the edge server through the ALIX.3, were the Pi was connected via WiFi and the local server was connected via Ethernet. After each image classification, the result was sent to the NGSI on the remote server through the ALIX.3 gateway. In these two scenarios, if the requests were to be submitted sequentially and continuously this would have resulted to an average processing of fifteen images per minute. So, in order for the results to be comparable, we restrained the request rate for both scenarios to three requests per minute, by interjecting idle time between each offloaded request.

The results depicted in Fig. 3, 4, 5 and 6 are used to evaluate the efficiency of the three scenarios. Fig. 3 depicts the mean power consumption, as well as the energy consumption per request of the processing and offloading actions (i.e., minus the idle consumption of the nodes). Fig. 4 illustrates the request response times (sorted in ascending order) along with their mean values and Fig. 4 shows the average data volume transmitted to the Cloud, in a logarithmic scale.

B. Edge Only Scenario

In this first scenario, we demonstrate the edge case where the resource-constrained UAV nodes undertake all the processing load, assessing the impact that this decision has on the system's response time, the nodes' battery life and the network traffic generated towards the Cloud (Fig. 2 (a)). The UAV nodes are responsible for periodically capturing images of the field and natively executing the image classification service. If an emergency situation is detected with probability higher than a predefined threshold, i.e. the image is classified as fire-containing when the classification score is above 50% in our case, this information is transmitted to the Cloud for further analysis by the Decision-Making module.

As expected, the Edge nodes struggle at the execution of such compute-intensive tasks, like the image classification process, and the results come as a confirmation; as shown in Fig. 3, power consumption at the UAVs is twice as much as in the other two scenarios. This effect becomes more acute when investigating the energy consumption per request, which, as we observe, is about twelve times higher than when the processing is offloaded at the Fog or the Cloud. Additionally, the struggle is evident in the time it takes for the system to come up with a decision; we can see in Fig. 4 that this performance metric averaged a 19.5 sec/req in the Edge Only scenario, while at the same time, the Cloud and Fog Only scenarios averaged only 4 and 4.5 sec/rec respectively. However, discussing about the average network traffic generated towards the cloud tips the scale in favour of the current scenario (Fig. 5); this behaviour is expected as the Edge nodes transmit only the minimum information to the Cloud, this of the classification score.

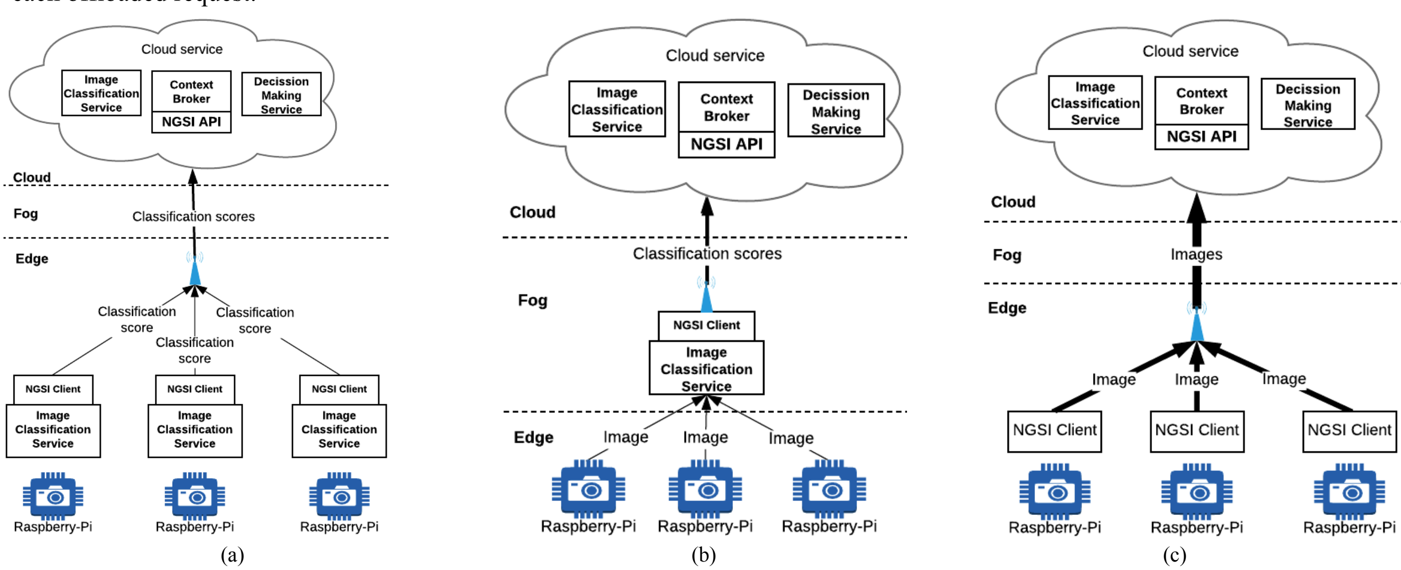

Fig. 2. Processing schemes for the emergency detection demonstrator: (a) "Edge Only", (b) "Fog Only", (c) "Cloud Only".

C. Cloud Only Scenario

In this scenario, the opposite edge case is investigated, where all the data processing takes place at the Cloud; every image captured by the UAVs is transmitted to the Cloud where it is evaluated by the image classification service (Fig. 2 (c)). The classification information is further analyzed in the Decision-Making module, running also in the Cloud. This situation, as foreseen, results in lower power and energy consumption in the Edge nodes, as the UAVs only periodically offload the images to the remote server. The "infinite" (compared to the classification process' requirements) resources allocated there easily undertake the image processing, helping this scenario achieve the lowest response time among the three, as well. These results are depicted again in Fig. 3, 4, 5 and 6. Nonetheless, these savings do not come without a cost; as illustrated in Fig. 6 the average volume of the data transmitted to the Cloud is substantially larger, compared to the other scenarios, and this is a consequence of the decision to offload whole images to the remote server.

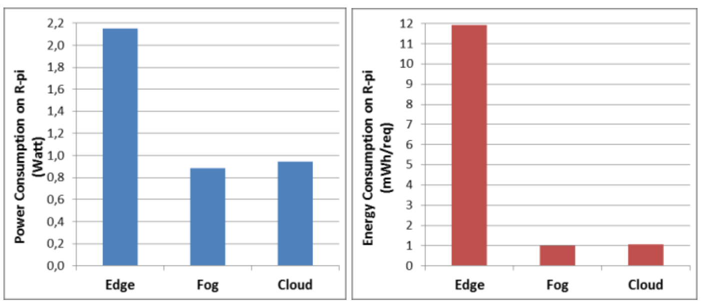

Fig. 3. Power consumption (in Watt) and Energy consumption (in mWh/request) for the Edge-only, Fog-only and Cloud-only approaches measured via the emergency detection demonstrator.

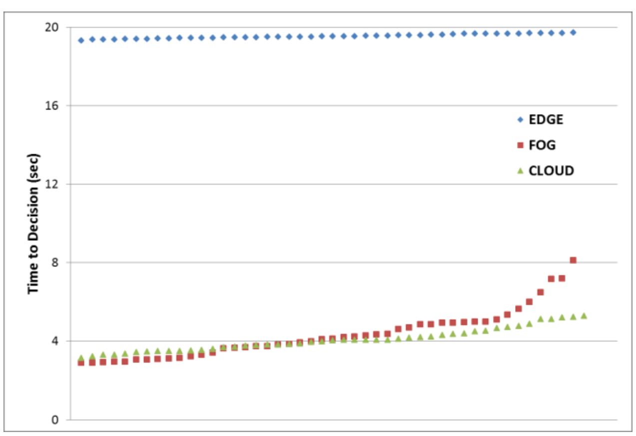

Fig. 4. Time to decision (in sec) for the Edge-only, Fog-only and Cloud-only approaches.

D. Fog Only Scenario

Finally, in the last experiment scenario, the utilization of the proposed intermediate Fog layer is evaluated and compared with the abovementioned edge cases; as already discussed, this time, the images captured by the UAVs are offloaded to the workstation located in the local area network, where they are classified by the containerized image classification service (Fig. 2 (b)). Being the most balanced among the presented scenarios, offloading the processing tasks to the Fog layer combines the best of both previous worlds; energy and power consumption at the Edge nodes is approximately as low as in the Cloud Only scenario (Fig. 3) because once again the UAVs only periodically offload the captured images for classification. In the same manner, response times fluctuate around the Cloud Only scenario measurements, averaging only 10% above them, as shown in Fig. 5. This slight delay is a consequence of the increased classification execution time at the local server, where the available to the algorithm resources are more constrained. Furthermore, the observed substantial variation in the response time is attributed to the connection instability induced by other services using the local AP at the time of the experimentation. Finally, the average network traffic generated towards the Cloud (illustrated in Fig. 5) is the same as the one in the Edge Only scenario, as in both cases only the classification score string is transmitted to the remote server.

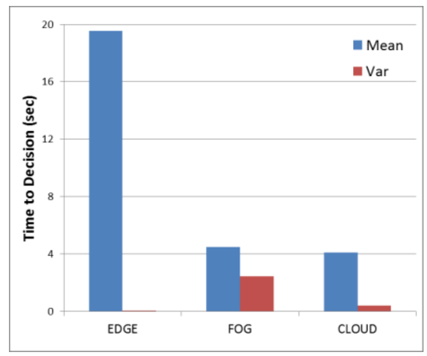

Fig. 5. Mean values and variation for the time to decision parameter for the Edge-only, Fog-only and Cloud-only approaches.

As presented thoroughly in Subsections B, C and D, each scenario has its benefits and drawbacks. However, some of them could not be easily plotted in figures and assessed. The issue of data privacy is a critical and difficult to visualise one and this is an area where the Fog Only and Edge Only scenarios excel. Contrary to the Cloud Only scenario, sensitive data like images of private premises travel solely in "in-house" networks and nodes, while only neutral information, like classification scores, are exposed to the public Internet. Another issue is the coverage of the event of interest; as mentioned earlier, request rate in Fog and Cloud Only was manually restricted to match that on the Edge Only scenario. Yet, UAVs in these scenarios are able to potentially provide better coverage with additional angles by capturing and offloading more images within the same time period.

Finally, it should be highlighted that due to the fact that our measurements were conducted in a testbed environment, using the available 802.11ac communication channels instead of -the more realistic- 4G LTE ones, the differences between the three approaches are potentially underestimated in this initial experimentation. Eliminating this "distortion" is expected to result in a reducing the difference between the energy and power consumptions of the three scenarios, as mentioned in [19], but the response times and the generated network traffic are not expected to significantly change. This will be investigated thoroughly in the second experimentation phase.

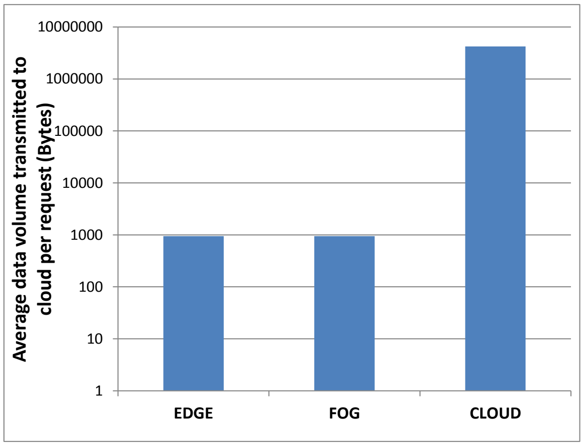

Fig. 6. Average volume of data transmitted to cloud (in Bytes) per emergency detection request for the Edge-only, Fog-only and Cloud-only approaches.

VII. CONCLUSIONS

This paper presented a three-layer architecture for early fire detection in large forest areas. UAVs equipped with Raspberry Pi devices capture the necessary image data that can be processed locally or in the fog or cloud layer. Initial experiments indicate that the fog offloading approach demonstrates the most balanced results among the executed evaluation scenarios, in terms of response time, network utilization and energy consumption. These results will act as the necessary input for proceeding with the next experimentation phase, towards the integration of well-established IoT based solutions in the forest-fire detection application domain. Future plans of the authors include the development of an intelligent resource allocation algorithm that will be able to dynamically predict the workload of the various nodes and proactively reserve adequate resources on fog nodes to optimally accommodate the respective processing load.

ACKNOWLEDGMENT

This work has been partially supported by the European Commission, Horizon 2020 Framework Programme for research and innovation under grant agreement no 645220.

REFERENCES

[1] J. San-Miguel-Ayanz et al., Forest fires in Europe, Middle East and North Africa 2016, vol. 28707. Publications Office of the European Union, 2017.

[2] V. V, “Image Processing Based Forest Fire Detection, “Emerging Technology and Advanced Engineering, pp. 87-95, 2012 .

[3] C. Yuan, Y. Zhang, and Z. Liu, “A survey on technologies for automatic forest fire monitoring, detection, and fighting using unmanned aerial vehicles and remote sensing techniques,” Canadian Journal of Forest Research, vol. 45, no. 7, pp. 783–792, Jul. 2015.

[4] S. G. Gupta, M. M. Ghonge, and D. P. M. Jawandhiya, “Review of Unmanned Aircraft System (UAS),” vol. 2, no. 4, p. 13, 2013.

[5] M. Satyanarayanan, “The Emergence of Edge Computing,” Computer, vol. 50, no. 1, pp. 30–39, Jan. 2017.

[6] L. Merino, F. Caballero, J. R. Martínez-de Dios, J. Ferruz, and A. Ollero, “A cooperative perception system for multiple UAVs: Application to automatic detection of forest fires,” Journal of Field Robotics, vol. 23, no. 3–4, pp. 165–184, 2006.

[7] C. Yuan, Z. Liu and Y. Zhang, "UAV-based forest fire detection and tracking using image processing techniques," 2015 International Conference on Unmanned Aircraft Systems (ICUAS), Denver, CO, 2015, pp. 639-643.

[8] A. C. Watts, V. G. Ambrosia, and E. A. Hinkley, “Unmanned Aircraft Systems in Remote Sensing and Scientific Research: Classification and Considerations of Use,” Remote Sensing, vol. 4, no. 6, pp. 1671–1692, Jun. 2012.

[9] H. X. Pham, H. M. La, D. Feil-Seifer, and M. Deans, “A distributed control framework for a team of unmanned aerial vehicles for dynamic wildfire tracking,” in 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2017.

[10] C. Yuan, Y. Zhang, and Z. Liu, “A survey on technologies for automatic forest fire monitoring, detection, and fighting using unmanned aerial vehicles and remote sensing techniques,” Canadian Journal of Forest Research, vol. 45, no. 7, pp. 783–792, Jul. 2015.

[11] Loke SW. The internet of flying-things: Opportunities and challenges with airborne fog computing and mobile cloud in the clouds. arXiv preprint arXiv:1507.04492. 2015.

[12] N. H. Motlagh, M. Bagaa and T. Taleb, "UAV-Based IoT Platform: A Crowd Surveillance Use Case," in IEEE Communications Magazine, vol. 55, no. 2, pp. 128-134, February 2017.

[13] N. H. Motlagh, T. Taleb, and O. Arouk, “Low-Altitude Unmanned Aerial Vehicles-Based Internet of Things Services: Comprehensive Survey and Future Perspectives,” IEEE Internet of Things Journal, vol. 3, no. 6, pp. 899–922, Dec. 2016.

[14] “EENA - 112 - European Emergency Number Association,” EENA - 112 - European Emergency Number Association. [Online]. Available: http://www.eena.org/. [Accessed: 19-Mar-2018].

[15] E. Koo, P. Pagni, S. Stephens, J. Huff, J. Woycheese, and D.R. Weise, “A Simple Physical Model For Forest Fire Spread Rate” in Fire Safety Science 8: 851-862. 2005.

[16] AIOTI WG03 – loT Standardisation, “High Level Architecture (HLA)”, Release 3.0, June 2017. [Available online at: https://aioti.eu/wp-content/uploads/2017/06/AIOTI-HLA-R3-June-2017.pdf]

[17] “TensorFlow,” TensorFlow. [Online]. Available: https://www.tensorflow.org/. [Accessed: 20-Mar-2018].

[18] M. Avgeris, N. Kalatzis, D. Dechouniotis, I. Roussaki, and S. Papavassiliou, “Semantic Resource Management of Federated IoT Testbeds,” in *Ad-hoc, Mobile, and Wireless Networks*, 2017, pp. 25–38.

[19] J. Huang, F. Qian, A. Gerber, Z. M. Mao, S. Sen, and O. Spatscheck, “A close examination of performance and power characteristics of 4G LTE networks,” in Proceedings of the 10th international conference on Mobile systems, applications, and services - MobiSys ’12, 2012.