Deployment of IoT applications on 5G Edge

Péter Kiss[1], Anna Reale[2], Charles Jose Ferrari[3], Zoltán Istenes[4]

Abstract— **The key for the future Internet of Things (IoT) is the shift from a static architecture to a dynamically evolving, self-organizing one. Devices of extremely varying capabilities need to collaborate and access all information necessary to ensure their optimal work, keeping a certain flexibility in the network configuration. The supporting infrastructure should be able to optimize resource consumption and to allow the connected device to interact with the most convenient node in the network, without compromising the perceived Quality of the Service.**

The work as a whole is an effort to summarize the process to build an offloading framework for arbitrary task of IoT devices. Another main contribution is the collection of requirements for this new IoT networks and computing systems, in particular the need for Edge Computing, Offloading paradigms and an underlying infrastructure which could be supported by means of 5th Generation mobile networks standards, Clustering and the use of Artificial Intelligence. In this paper we do not present an actual, finalized implementation but we propose and analyze a global view of the problem and identify some specific possible technical solutions.

I. INTRODUCTION

Internet of Things (IoT) applications need to handle a number of information from a great amount of heterogeneous devices. In the past IoT devices were typically considered as external, small hardwares of limited resources, like sensors, residing at the edge of the involved network infrastructure. Due to this assumption, the main role of an IoT device is to blindly transmit sensed data or to react to environment changes up to some extent.

Due to limitation on the computing resources of IoT devices, a common practice is to offload tasks of various applications to computing systems with sufficient computing resources: data centers in the cloud. However, the offloading method's drawbacks are high latency and network congestion in the IoT infrastructures [9]. In relation to this issue, the paradigm of Edge Computing (EC), with the idea to support the devices with a cloud closer to the edge of the network appears as an appealing solution. Adding Edge resources though complicates the management of the network because multiple devices will contend them.

Furthermore, the recent evolution of IoT brings more and more devices which are not simple sensors or transmitters anymore, but also provide limited execution environments. This opens up a huge opportunity to utilize this previously untapped processing power in order to offload custom application logic directly to these edge devices [3]. In this extremely complex landscape, it is an essential question how to balance the resources available and their tasks in a way that profits from the added capabilities of these new IoT devices without compromising the final performance of the network.

The first key question in future IoT networks then is how to enable devices of extremely varying capabilities to collaborate and to access all information necessary for ensuring their optimal work while keeping a certain flexibility in the network configuration. In relation with collaboration of IoT devices an important issue is the connectivity since the continuously growing number of devices generate congestions in the communication channels.

Third and last main change in the IoT is the advent of mobility: we have to consider many different kinds of new objects. The definition of "Things" is quite broad, involving now from smart phones to even smart cars. Things are actually any physical objects, anything that has a real life presence. Such object can nowadays be installed in moving vehicles or are mobile themselves.

For all the presented scenarios, there are dynamic changes in the configuration at the edge of the network which can happen frequently; for example, connected devices may be physically moving, the network might need to balance resources or to reallocate them to achieve system faults tolerance. These continuous changes require from the IoT network to be able to reorganize itself in a way that optimizes resources consumption (e.g. bandwidth, storage, power) and allow the connected device to interact with the most convenient node in the network, without compromising the perceived Quality of the Service (QoS). Simply copying the whole applications in every node that requires them cannot be a scalable nor a maintainable solution, thus an offloading framework is still needed.

To answer the connectivity question of the ever-increasing number of devices a solution - already applied for Wireless Sensor Networks (WSNs) - is to form groups of devices and manage their connections in a collaborative way.

A network of such devices could profit from the 5th Generation (5G) mobile networks standards and infrastructures to achieve goals otherwise unattainable with only Edge computing and offloading such as real time processing of massive amount of data and low latency of 1 millisecond.

The network should be able to adapt to fluctuations of resources load. Since the forthcoming 5G networks are more complex architectures than their predecessors, the number of configuration variable makes it very difficult to apply some deterministic adjustment approaches. For this reason, we believe that the IoT framework will benefit also from the

[1],[2],[3],[4], ELTE University - Hungary

The research has been supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.2-16-2017-00013).

[1]axx6v4, [2]anna.reale, [4]istenes at [inf.elte.hu]

[3]ferrari at [caesar.elte.hu]

emerging Artificial Intelligence (AI)-based technologies that are designed precisely to cope with similar challenges.[30]

Our main contribution in this paper is to summarize and redefine requirements for this new IoT networks and computing systems, in particular the need for a platform on top of 5G edge computing and computational offloading paradigms. Compared to previous work we go into more details regarding the possible technical solution that may be used to satisfy the requirement of this IoT platform, in particular we give an overview of the different choices for modeling and partitioning application and we individuate some progress in clustering and AI that can support the development of such complex solutions.

In our effort to summarize the process needed for building an offloading framework for arbitrary task of IoT devices we specified five main parts:

- Discovery and modeling
- Planning and optimization
- Execution
- Monitoring and performance maintenance
- Learning and predicting

The paper is structured as follows: Chapter II offers a background on technologies that can enable building an efficient task offloading framework for IoT devices as 5G, Edge Computing and also presents basics of computation offloading, clustering and Artificial Intelligence. In Chapter III we suggest a collection of requirements for the framework and the steps to be followed for a future iterative implementation. Finally in Chapter IV we discuss how 5G technologies, Clustering and Autonomic Computing Architecture concepts [35] can benefit from our IoT framework.

II. BACKGROUND

In this section we give some basic definition for the involved technologies and paradigms and try to summarize the mainstream directions of the literature.

A. EDGE COMPUTING

As the number of mobile devices and services requiring computing or storage capabilities that significantly exceed their own capacities has exploded, the paradigm of Cloud Computing (CC) had arisen and gained a continuously increasing importance. The concept of CC is based on Data Centers (DC), which are capable of coping with storage and processing requirements of tasks involving large scale of data. Moreover, data centers are usually connected between each other over optical cable building up Data Center Networks (DCNs) that, due to extremely low internal communication costs, appears to the outside world as a single entity. When facing a problem that outruns available local resources one can offload the code and the data to the cloud, then - when computations are done - receive the results back.

The paradigm of CC has given a solution to the scalability problems related to inefficient resources, the code and data migration, however, may involve a significant latency or cause congestions in the network. The root of the problem can be summarized as location unawareness of the CC paradigm and is getting severe with time as more and more (semi-)intelligent devices will attempt to connect to DCNs.

The concept of Edge Computing has emerged to leverage the storage and computation capabilities of the edge devices that are connected to the Internet and meant to be an intermediate layer between the devices. They are able to handle a subset of requests that would be usually sent to the cloud, but which in fact do not need its real involvement because of the diminished resource requirement or no need for the DC to be involved. Thus, thanks to the presence of this type of device the computation load of DCs is reduced, similarly the latency of responses when an application needs to have real-time or almost real-time responses. Moreover, due to its geo-distributed nature and high availability, the Edge layer is apt to handle the challenges of mobility e.g. serve the requests of moving users as it is necessary in cases of autonomous cars or streaming and real-time gaming in a high speed vehicle.

1) Comparison of Edge Computing Implementations: Fog Computing, Cloudlet and Mobile Edge Computing: In the following paragraphs we present a distinction of Edge Computing into three categories, as previously proposed [28]: Mobile Edge Computing (MEC), Fog Computing (FC) and Cloudlets.

a) Fog Computing: in this paradigm Fog Computing Nodes (FCN)[17] can be placed at any point of the architecture. FCNs are highly heterogeneous; they can be built on various devices as routers, switches, IoT gateways, set-top boxes and so on. Heterogeneity of devices leads to ability for working with different protocols as well as with non IP-based technologies in communication between FCNs and the end-devices. Since Inhomogeneity of the edge naturally should stay hidden from the user devices, FC system exposes a uniform interface containing storage and computational services along with monitoring security and device management facilities. On top of this abstraction layer an orchestration layer organizes resource allocations according to the users' requests.

b) Cloudlet: The term Cloudlet[16] can be defined as a trusted set of computers having a good connection to the Internet and making their resources available to nearby mobile devices. The Cloudlet runs a virtual machine that is capable of provisioning resources to the connected users near real-time over a WLAN network in one-hop distance and with a high bandwidth. Above provisioning the infrastructure, Cloudlet architecture also provides a middleware framework support to component based applications designed with a focus on applications with strict real-time requirements such as augmented reality.

c) Mobile Edge Computing(MEC): MEC[15] is responsible for bringing storage and computational resources to the edge of Radio Access Network which enables us to reduce latency and improve location awareness. A clear choice for deployment of MEC nodes is to collocate them with the Radio Network Controller or a macro base-station. Servers of the MEC node run multiple MEC host instances with their own computational and storage resources. A MEC

orchestrator is in charge for monitoring hosts and state of the network, keep track of which host offer what services, track the available resources as well as information on devices that are connected to the servers including location and routing informations.

B. 5G

The Fifth Generation of mobile network is still under standardization process but its main general characteristics are more or less publicly available. As explained in the most recent Europeans actions plans on 5G [31], the technology will be based on:

1) *Economic fibre-like radio access* reaching data rates beyond 10 Gb/s, through the usage of higher frequency bands above 6 GHz and related technologies.
2) *Network Function Virtualization (NFV)*, will allow implementing specific network functions in software running on generic hardware without the need for costly hardware-specific machines. Reducing implementation, management and operational costs; Allowing reuse and sharing of the same functionality between customers;
3) *Software Defined Networking (SDN)*, will allow the control of network resources to be opened to third parties, Flexibility to accommodate demanding professional-grade applications.

Core 5G applications requirements that are out of the current 4G technology's abilities will be:

1) low latency of 1ms (10 to 20 ms for 4G),
2) serving 1 million of devices/km^2 (about 1000 device/km^2 for 4G)
3) fast deployment of new services in 1 hour time (done in days with current technology)

C. OFFLOADING AND PARTITIONING

We refer to computation offloading as the act of transferring a certain computing tasks to an external platform [13]. Historically the main motive for offloading is to augment the mobile systems' capabilities trough code migration which enables these systems to save energy and improve their performance[14]. Offloading, can be particularly useful in case of computation and data intensive applications such as AI, artificial vision and object tracking[12].

Offloading should occur when it is beneficial to the mobile application, that is, in most cases when outsourcing the computation can improve response time or saves energy. Calculating this, however, can be complicated and depends on several parameters. The determinations of whether and what to offload are most frequently based on parameters of the hosting device, the network, and the cloud infrastructures. This includes bandwidths, server speeds, available memory, server loads, and the amounts of data exchanged between servers and mobile systems. Computing benefits in time, as an example, should include time lost for the transmission and time gained on the faster cloud/edge computation.From the perspective of total energy consumption an important factor is that the edge/cloud is relatively more energy efficient than the smaller device which is a crucial consideration since the device can have energy constraints (battery)). All of these should be taken into account with the priority specified based on the "optimization" criteria. However heavy loaded network can inhibit the offloading.

The techniques of splitting up an application into separate components, while preserving the semantics of the original application, are referred as Application Partitioning (AP)[11]. The objective of AP algorithms is to divide the code into logical units or tasks (candidates for offloading), and between them to clearly specify the interfaces of interaction.

AP algorithms in literature[11] can be classified according to various parameters: task granularity, partitioning objective, application model, language support, profiler used, allocation decision, analysis technique, and usage of annotation (automatic or manual).

Partitioning objectives are usually one or two between the following: increasing application performance, reducing memory constraints, reducing network overhead, reducing friction of adoption for the programmer, saving energy.

Models used to represent and optimize the program are Graph-based, Linear programming-based or some hybrid in-between solution.

Allocation decision may be made either online (at runtime) or offline (at deployment time). The selected parts can be statically assigned (thus remain unchanged during the whole life of the application) or they could be dynamically modified.

Static partitioning of code assumes the program being divided either during development or at first deployment. This allows a low overhead during execution; however, this approach is valid only when the influencing parameters can be accurately predicted in advance and are not expected to vary drastically in time[14]. Dynamic partitioning, on the other hand, allows application deployment to adapt to changes in the environment (ex. bandwidth) at run-time. This makes the application execution performance higher at the price of additional latency and resource usage. In fact, in dynamic offloading every time a new optimal solution is found there is a need for coordination and redeployment of application among involved nodes. In case of a rapidly evolving environment, this translates into a challenge to identify an adaptable partitioning of the application to offload which is still advantageous and fast enough to justify the offloading itself.

D. CLUSTERING FOR WSN AND IOT SYSTEMS

Various clustering techniques[7] in Wireless Sensor Networks as well as in IoT networks are addressing connectivity and bandwidth problems of the very big - and continuously growing number of devices - attempting to connect to the Internet.

1) Clustering in WSNs: Clustering techniques in IoT systems refer to topology control methods aiming at a more optimal usage of resources as energy, bandwidth and latency where it is necessary. Clustering methods achieves

these purposes through building up groups of devices to be connected so that those groups can manage collaboratively the resource usage. These methods in recent Wireless Sensor Networks (WSNs) build on a rather static approach in terms of membership of the constituting devices. The most important role in WSN clustering is that of the Cluster Head (CH), a device in the network that is responsible for transmission scheduling. CH gathers information from all sensors of the cluster and after execution of some assembly methods forwards the preprocessed data to the gateway. Responsibilities of CH includes optimization of their work in terms of energy efficiency i.e. prolonging network lifetime through partially switching off redundantly deployed sensors when their work is not necessary to maintain QoS. This optimization brings two advantages:

1) scheduling duty cycles cuts consumptions of sensors whose work is not necessary at the given moment,
2) reducing number of sensors trying to communicate lowers the probability of conflicts and leads to smaller latency (and again, naturally reducing energy consumption caused by retransmission attempts).

Forming clusters of devices can be carried out through the following steps:

1) Election of Cluster Head(CH). Since CH is in charge for organizing work and communication of the cluster, it will consume way more energy than the others. Election therefore should be carried out after careful investigation. Optimizing the lifespan of the network also requires a frequent rotation between nodes. In recent systems CH assignment is usually executed applying the following strategies:
   a) Deterministic methods mostly in case of more powerful supernodes available (having superior processing capabilities or better energy supply), they will be chosen for taking care of coordination in the network.
   b) Random election, e.g. picking CH based on assignment of random values suits well for sensor fields of homogeneous devices and balanced workload.
   c) Adaptive election refers to methods that take into account specified parameters of consisting nodes such as residual energy or distance to BS.
2) Cluster formation. After some CHs have been selected, they announce their role to neighboring nodes, wich will make decisions about what cluster to join according to what is the most beneficial to accomplice their own tasks and also based on their communication range. Determining parameters can also be communication distance, number of hops, physical distance or, in some cases, size of the cluster and many others.

E. AI IN NETWORKING

The evolution of communication and computer networks caused a shift from static or deterministic management techniques to more flexible, robust and adaptable self-organizing technologies. These new methodologies are able to cope with heterogeneity and complexity growth[29]. They aim at regrouping resources, re-optimizing communication channels in case of radical requirement changes, or at restoring capabilities in case parts of the networks falls out due to malfunctions.

Due to the number of different possible configurations to be taken into consideration it was an obvious choice to start applying AI techniques to handle such an exploding complexity in network management.

To have a general understanding of AI-managed systems, we can schematize their action with the following cycle aimed at maintaining and/or developing the quality of a given system[30]:

1) By monitoring the environment builds up a model of the system,
2) classifies the detected problems, then
3) advocates solutions to adapt to the circumstances in order to achieve a better quality of service.

III. REQUIREMENTS FOR A MOBILE IOT FRAMEWORK

In this section we will describe some of the requirements linked to IoT, IoT applications and their deployment.

A. General IoT Requirements

The main challenges in IoT has always resided in handling a considerable amount of data and defining services on top of those [18]; therefore, IoT needs a common high performance network with a common uniform architectural base as addressed previously [5] in the review of the current IoT technologies.

As the range of possibilities for IoT increases this network gains a more heterogeneous and complex shape [5].

In some scenarios, sensors spend a large part of their time in a sleep mode to save energy and cannot communicate during these periods.

However, the new smart devices produce huge volumes of real-time streaming data, generating a need for effective techniques to transmit and process data streams and to gain insights and actionable information from real-world observations and measurements [8].

In contrast, some sensors may need to apply ad hoc communication patterns, for example if they are designed to communicate only if certain rules are triggered.

In addition, these communicating devices operating with different networking standards may experience intermittent connectivity with each other, some could have limited transmission range and many of them will be resource constrained. These characteristics open up several networking challenges that traditional routing protocols cannot solve.

It is important to state that the existence of Internet of Things devices is justified only if there are applications exploiting their abilities. In fact, to stimulate usage of their products, manufacturers are creating software ecosystems that enable third-parties to develop applications for their devices. Since such applications are developed by third

parties, their seamless integration into the platforms is a significant challenge [2].

Furthermore, since IoT applications often do not provide complex user interfaces, their requirements should be modeled and resolved before their integration in the platform, thus descriptions of applications and environment specifics should be built and matched in advance [2].

Due to limitation on the computing resources of some IoT devices to run the application with a reasonable performance, connecting to additional computational capabilities like Edge servers can become necessary.

All these heterogeneous and fluctuating requirements must be satisfied trough dynamic routing and coordination, a good number of access, transmission and service provisioning mechanism and by Service Provisioning Management and orchestration.

Many works in the recent years have tried to resolve this issue trough various architectures and orchestrator definitions; the two most similar to this work will be reviewed in the following section.

B. RELATED WORKS ON IOT INFRASTRUCTURES

Applications in the Internet of Things (IoT) domain need to manage and integrate great amounts of heterogeneous devices. To facilitate such a task, IoT software ecosystems may be using an architecture that exploits semantically enriched applications[4]; the disadvantage of this approach results in more complex descriptions of applications and an increase of the developer burden.

To support IoT applications, it is necessary to have efficient task management. In[9] the authors propose a method to periodically distribute incoming tasks to increase the number of performed ones while still satisfying the quality of- service (QoS) requirements of the tasks. The approach seems to have better performance when the number of input tasks is large, data size of input tasks is large, or the connectivity of the edge network is high. But as the author states it needs to consider different level of prioritized inputs.

In [10] the author present their vision and initial design efforts towards a distributed IoT orchestration architecture, but no working implementation is presented.

The authors concentrate as a first challenge on the locality and workload aware computation partitioning between midway servers and edge gateways and between edge gateway and edge clients running on the corresponding edge device. They refer to the need of an intelligent task partitioning mechanism to enable real-time services provisioning with high scalability, but no proposal is done in this sense.

They advocate that their orchestrator should allow intelligent partition of a real time IoT computing task into an optimal coordination of server-side and IoT object processing. This makes it possible to scale in real time when objects are moving, as well as enables the system to continously take into consideration the changing resource availability and workload at the distribution of computation.

The author stresses the need for a resource-aware allocation model that can dynamically schedule and allocate resources and a workload-aware resource scheduling of multiple services ensuring that the tasks run concurrently while taking into account of the object side processing workloads. Finally a resource-aware selection of computation and an execution environments for a collection of IoT service provisioning requests are mentioned. The authors also refer to the decision on how and what to offload from edge devices to a network of IoT stations (servers) or to a cloud data center.

In [4], the authors explore the requirements of a platform for IoT through semi-structured interviews with employees with roles in software architecture, engineering and management at their industrial partner. We can summarize their findings in 3 main models:

1) Contextual Variability: the software should adapt to the context of the device. The context to be modeled consists of connected devices (e.g. sensors and actuators), sensor readings (e.g. chemicals in water) or features of the installation (e.g. location, temperature).
2) Modeling Functionality of Applications to enable a more efficient and goal-driven user interaction
3) Model of Deployment Architecture to express contextual and system constraints of the applications.

Both approaches still do not consider the need of a specific modeling of the movement in order to forecast and predict future network topology and answer to situations in which a fast reaction to context changes is needed. A semantic description of the devices would also be useful for such an ending.

IV. BUILDING A MOBILE IOT FRAMEWORK

Given the analysis of the previous chapter, and the background knowledge on previous efforts on computation offloading and application partitioning, we believe that the a framework designed for offloading arbitrary tasks of IoT devices in a 5G Edge Computing environment, should implement the following five main parts:

- Discovery and modeling
- Planning and optimization
- Execution
- Monitoring and performance maintenance
- Learning and predicting

a) Discovery: The first step in finding the optimal way for executing a task should be profiling the context and the task itself. Profiling of a given task can be carried out in several ways. The essence of this phase is to discover locations of the code at which the execution can be distributed and/or parallelized. Having an enriched description or abstraction of the different resources available could help with the integration from different vendors HW and SW and with fastening this profiling. Developers may annotate an IoT application to influence the result of the profiling and the consequent partitioning decision. An offline static analysis may help separate monolithic applications into candidates for tasks, for example identifying part of the code that can be run in parallel or that are independent from each other

and therefore can be deployed on different machines. The scope of the context discovery is to build up a model or representation of an available network slice for which the deployment of the subtasks is reasonable. In this analysis we have to gain information on costs of transmission, available resources and the capabilities and current workload of reachable nodes in the network. A combination of static invariable knowledge and dynamic collection of this data through simulation and estimation models will be needed. For example measurements of the average consumption of the battery per instruction or task size could be modeled, while total memory is a static data, finally the available resources on a node of the network change during the time and need constant monitoring. Based on previous works [11] we believe a graph representation of the task connections and the cost associated to run them in different available nodes would be more performing and less resource intensive than a linear programming (LP) model.

b) Planning/Optimization: Thus the results of the Discovery phase should be applied to elaborate a sort of place-and-route graph, as a plan to deploy our subtasks in the available network. Making deployment plan is very similar to a *path computation and function placing* problem; already well known task must be solved by network manger in NFV/SDN settings. Even et al. in [22] described an approximation algorithm for addressing this problem in an efficient and flexible way. Integrating this solution with methods for faster Pareto-optimal solutions may allow to comply to a more strict real-time requirement. Further restriction to the slice of the network to consider may be posed by the application of clusters as described in the previous section.

c) Execution: At the execution phase first we have to migrate codes of subtasks to their execution location according to the execution plan built up in the previous phase. After all parts are on their dedicated location the system can finally start the required computations. The challenge that the offloading framework will face at this point is the orchestration of the collaboration of the resulting micro-services, among others, the efficient scheduling and seamless data transmission between the nodes. This issue should definitely be addressed using Clustering techniques - even though there is still not a clear way to keep into considerations the mobility of the whole system - clustering will help reduce the managing complexity and also improve the energy saving and the communication.

d) Monitoring: following from the fact that the whole infrastructure will not be dedicated to one single software of interest, other services are going up and down with time. That makes necessary to monitor the performance of our system and execute reorganization of the partitioning of some pre-deployed applications to ensure a better global performance.

e) Learning and Predicting: To follow the requirement of self-adaptation and self-configuration, the framework should have an ability to estimate and predict the context and the task requirements. In this the AI technology comes to help in various fashions, as explained previously.

In the next sections we specify how the mentioned technologies can contribute to realize the described offloading and execution of an arbitrary IoT application.

A. 5G EDGE FOR IOT

Systems made of resources constrained devices such as IoT sensors can rely on offloading to handle a particularly computation intensive task. Edge Computing supports the IoT devices with a cloud closer to the edge of the network allowing for some of their task to be offloaded.

Also the Edge servers can be part of the IoT network where the management and orchestration of the IoT devices is performed, for example electing every Edge server as a Cluster Head. Adding Edge Computing servers alone complicate the network infrastructure and still does not solve completely latency related issues. Finally, the Edge servers should be flexible and easy to reconfigure benefiting from SDN and VFN. All of these observations justify the need for integrating the Edge Computing paradigm with a 5G architecture. In this way, we can have high computations abilities and an almost real time response of the system, regardless to the physical position and to the capability of the involved IoT objects.

In the literature the Edge Computing concept and 5G have been already suggested as a solution to IoT.

Among the applications proposed by [20], Smart Trains should be mentioned which needs a sensors network providing the carriage information into a central unit; in this case, simply using local area network technologies. Some more hybrid and composed approaches are, for instance, in [19], where the authors use Personal Area Network to collect sensors information and retrieve those to the Edge using a more robust connection.

As a further justification for implementing IoT on top of 5G, there is the fact that such infrastructure would allow the capabilities of machine to machine (M2M) communication to be inherited by IoT devices.

In classic M2M communication there is a lack of trustworthy communication channel, the bandwidth is limited and there is a strong dependence on the (usually slow) response from the server. As explained by [6], Smart Parking, Augmented Parks, Logistics and Vehicle to Everything Communication (V2X) are domain areas where the IoT can take advantage of a flexible real time and secure M2M communication. Such results will be achievable if supported by Edge servers and by a 5G infrastructure.

B. CLUSTERING FOR IOT

The Clustering strategies described in Chapter II are based on methods primarily developed for WSNs; therefore the majority of those assume that sensors in the networks are homogeneous. The emerging concept IoT systems, however, differs from this type of sensors-fields as it is aiming to serve the big variety of devices (including fat clients) that additionally can continuously change their position. Building clusters in this new environment necessitates therefore an updated resource optimizing process that is able to adapt to this dynamically evolving circumstances. The IoT backhaul

issues we face using 3rd Generation Partnership Project (3GPP) standard communication can be grouped as follows:

1) Energy efficiency is still naturally a problem, in static scenarios there are efficient methods to maintain QoS by shifting focus from longevity of individual sensors to the question of keeping the coverage of the network [23].
2) Management hierarchy Compared to WSNs, IoT has a heterogeneity of devices and subsequently a number of different services that those require in the network.
3) Data processing The exploding amount of data generated by increasing number of gadgets is treated as one of the most valuable asset in the time of Big Data revolution. However, the tremendous number of records poses a big challenge for 3GPP networks because it tends to overload links when in general the transmission of entire datasets are unnecessary and highly redundant. This boom of generated data makes inevitable to screen and assemble before sending on to Base Stations (BSs).

1) CHALLANGES IN CLUSTERING FOR 5G AND IOT: Comparing to WSNs, applying clustering on IoT over 5G raises some additional challenges. The first difference to WSNs systems is the vast heterogeneity of devices. In IoT networks a large number of transmit-only devices are present[24] along with super sensors(or actuators), e.g more powerful nodes that are in charge in collecting and transmitting data and in some cases due to their computation capabilities some preprocessing or even organizing tasks[25]. It is natural to use these super sensors to be in charge of coordination of work of the cluster as Cluster Head (CH). It is worth to emphasize again that in IoT systems due to over-the-top applications, the task of CH become more comprehensive and complex.

Another point to consider is the cost implication of transmission. One part of these costs are still stems from the energy costs as in WSNs but since in 5G architectures mobile network is also involved we need to keep a strict control over the usage of LTE infrastructure[25].

It is an important issue how to exploit the presence of intelligent components of the core network through dynamically organizing the routing of the requests of IoT devices based on the congestion level of a given channel. In basic scenario, if it is available the most obvious choice is using wireless for transmission, but if an application has specific bandwidth requirements we have to enable the system to use assistance of LTE in order to provide the needed Quality of Experience (QoE)[26].

While clustering in 5G network we should take into consideration the needs of a given user/device and based on the result of profiling clustering should consider how to improve user utility. For example if we use TDMA MAC protocol to organize the access to the network for the members of the cluster we should avoid grouping users that require a low latency. On the other hand, grouping together devices with similar usage could enhance processing of transmitted data at the edge since CH would reduce redundancies before transmitting.

One of the most important and relevant issues in case of future IoT networks is how to handle the connectivity problem of fast moving objects complying the possibly high communication requirements. Previous works proved that it can be very advantageous having actuators being able to change their locations[27], [25], but how to organize clusters of moving objects with supernodes that change their position independently not driven by maintenance of QoS and QoE of IoT network can be a challenging question.

C. AI IN 5G

5G cellular networks encompass a good number of access, transmission and service provisioning mechanisms. These new technologies cover topics from Radio Resource Management through Mobility- and Service Provisioning Management to orchestration techniques [30].

The primary motives for using AI related technologies in 5G infrastructures is to enable the network to intelligently adjust its configurations as the requirements or parameters of the environment change.

The new 5G network should be able to provide efficient solutions for radio resource management (RRM), mobility management (MM), management and orchestration (MANO), and service provisioning management (SPM), making dedicated purpose networks no longer necessary supporting instead dynamic reconfigurations of networks as the concept of NS [21].

Comparing to typical 4G node, the number of configurable parameters is expected to increase up to 2000 or more from 1500. It is critical in the new architecture to enhance intelligence to enable it to cope with new demands for self-organizing features (e.g. self-configuration, self-optimization, and self-healing).

Service types (e.g., eMBB, URLLC, mMTC) that are defined in the context of 5G are static, but as new type of services arise continuously or patterns in existing services evolve, the system should be able to recognize these new type of service and infer the fitting provisioning mechanism for it consequently building up the required network slice.

The SDN architecture on 5G, with its centralized logic, will provide the possibility to completely reorganize routing and network function placement. However, keeping track of the state of several network devices and updating policies becomes even more difficult when increasingly sophisticated policies are implemented and only low-level configuration commands are available on the networking hardware[34]. To reconfigure such a network with the tools currently available, repeated manual interventions would be needed. In this sense, the developing of a management framework on top of 5G and SDN presents many challenges.

Such challenges can be summarized as the need for self-organization of the network which involves issues of: monitoring environmental changes, learning uncertainties, elaborating how to react to new circumstances, and setting up new configuration of the network to maintain efficiency.

Various multidisciplinary techniques (machine learning, optimization theory, game theory, control theory, and metaheuristics) that can be grouped under the term Artificial Intelligence are coping precisely with these topics.

In [30] authors describe a possible AI-supported cellular network architecture, where an AI-controller is deployed on top of the Open Network Operating System (ONOS), or an independent network entity. The controller communicates with Radio Access Network (RAN), Core Network (CN) and SDN controllers, and access service level agreements, informations on connected user devices from SDN controllers, and collects traffic informations from RAN. The AI-controller can consist of four logical module, corresponding the MAPE (Monitor, Analyze, Plan, Execute) loop [35]:

1) Sensing module collects all the relevant information on state of the network,
2) Mining attempts to discover patterns in the collected data,
3) Prediction makes predictions on the future state of the system,
4) Reasoning adjust parameters of the network to achieve a better performance.

An example of the application of AI controller relate to mobility issues. Sensing module tracks location of User Equipments (UEs), then using functionality of Prediction module makes guesses on the future whereabouts of the user based on the mobility patterns developed in the mining phases. At the end, according to decisions of Reasoning module that asks for location record updates and prepares the infrastructure to handover resources to serve the predicted requests of UEs.

A possible other application of AI is concerned with implementation of 5G networks, is shown by the CogNet[32] project. It is an architecture of an autonomic self-managing network extending Network Function Visualization management with Machine Learning-based decision making mechanism. The reason behind deploying a more adaptive controlling mechanism next to base NFV functionalities is the pursuit to reduce the costs of the system, whilst keeping QoS on a competitively high level.

The architecture applies the MAPE paradigm, whose functionalities are implemented by three building blocks: data collection and storage, Cognitive Smart Engine and Policy manager. [33]. Agents are used to continuously monitor the state of the network collecting records on the state of components, resource consumption of clients, and other relevant events. These records are then forwarded to Cognitive Smart Engine (CSE), where the intelligence of the system resides. The role of the CSE is to process those information to decide whether and how to allocate resources, to identify performance issues, to optimize network capacity, and to secure the network. The engine supports several Machine Learning modules to provide these services.

CSE selects relevant parts of data (feature selection), deliver it to the real-time time and batch processing engines. The engines score the incoming records, then based on scoring make conclusions or predictions on the state of the system.

According to the output of CSE the Policy Manager generates new control policies for MANO components if that is necessary. Control policies are defining some rules on how to adjust network configuration in case of they match the defined condition.

V. CONCLUSIONS AND FUTURE WORKS

In this work we addressed the need for Internet of Things to move towards more autonomous, scalable, connected and location independent infrastructures.

Our contribution in this work was defining the current possibilities of development of IoT services for the network and defining and collecting the requirements for a fully trustworthy combination of IoT technologies and 5G that allows to exploit the most of both.

We brought together relevant background knowledge on previous efforts on computation offloading and application partitioning, with state of the art challenges for IoT. We identified possible solutions to those trough complementary technologies and paradigms such as 5G Edge Computing and the integration of Clustering and AI methods.

5G Edge computing architecture responds to the needs of IoT networks of a fast reliable communication with less overhead. It facilitates computation offloading by reducing to the minimum link costs and it allows better management of the whole infrastructure adding decentralization and reducing the need to communicate with Central Data centers.

Clustering techniques can help to reduce the communication loads at the edges of the network to save energy and simplify the network management in a *divide et impera* fashion.

On the other hand application of AI based technologies enhance the infrastructures ability to adapt to continuously changing requirements and to reorganize itself when it is necessary.

Our further studies will be focused on creating a resource-aware framework based on this study, able to collect and distribute tasks across heterogeneous, vertically and horizontally distributed services, making the best usage of the capabilities of 5G Edge and Cloud Computing.

REFERENCES

[1] C. Meurisch, J. Gedeon, T. A. B. Nguyen, F. Kaup and M. Muhlhauser, "Decision Support for Computational Offloading by Probing Unknown Services," 2017 26th International Conference on Computer Communication and Networks (ICCCN), Vancouver, BC, 2017, pp. 1-9.

[2] M. Tomlein and K. Grnbk, "Building Models of Installations to Recommend Applications in IoT Software Ecosystems," 2016 IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud), Vienna, 2016, pp. 9-16.

[3] M. Vgler, J. M. Schleicher, C. Inzinger and S. Dustdar, "DIANE - Dynamic IoT Application Deployment," 2015 IEEE International Conference on Mobile Services, New York, NY, 2015, pp. 298-305.

[4] M. Tomlein and K. Grnbk, "Semantic Model of Variability and Capabilities of IoT Applications for Embedded Software Ecosystems," 2016 13th Working IEEE/IFIP Conference on Software Architecture (WICSA), Venice, 2016, pp. 247-252.

[5] P.P. Ray, "A survey on Internet of Things architectures." Journal of King Saud University-Computer and Information Sciences, 2016.

[6] S. H. Shah and I. Yaqoob, "A survey: Internet of Things (IOT) technologies, applications and challenges," 2016 IEEE Smart Energy Grid Engineering (SEGE), Oshawa, ON, 2016, pp. 381-385.

[7] L. Xu, R. Collier and G. M. P. OHare, "A Survey of Clustering Techniques in WSNs and Consideration of the Challenges of Applying Such to 5G IoT Scenarios," in IEEE Internet of Things Journal, vol. 4, no. 5, pp. 1229-1249, Oct. 2017.

[8] D. Puschmann, P. Barnaghi and R. Tafazolli, "Adaptive Clustering for Dynamic IoT Data Streams," in IEEE Internet of Things Journal, vol. 4, no. 1, pp. 64-74, Feb. 2017.

[9] Y. Song, S. S. Yau, R. Yu, X. Zhang and G. Xue, "An Approach to QoS-based Task Distribution in Edge Computing Networks for IoT Applications," 2017 IEEE International Conference on Edge Computing (EDGE), Honolulu, HI, 2017, pp. 32-39.

[10] Applications 48 (2015): 99-117. E. Yigitoglu, L. Liu, M. Looper and C. Pu, "Distributed Orchestration in Large-Scale IoT Systems," 2017 IEEE International Congress on Internet of Things (ICIOT), Honolulu, HI, 2017, pp. 58-65.

[11] J. Liu, E. Ahmed, M. Shiraz, A. Gani, R. Buyya and A. Qureshi, 2015. Application partitioning algorithms in mobile cloud computing: Taxonomy, review and future directions. Journal of Network and Computer Applications, 48, pp.99-117.

[12] Y. Yifan Mobile Edge Computing Towards 5G: Vision, Recent Progress, and Open Challenges Intel Labs China, Haidian District, Beijing, 100086, Chin

[13] S. Gurun, R. Wolski, C. Krintz and D. Nurmi, 2008. On the efficacy of computation offloading decision-making strategies. The International Journal of High Performance Computing Applications, 22(4), pp.460-479.

[14] K. Kumar, J. Liu, Y.H. Lu and B. Bhargava, 2013. A survey of computation offloading for mobile systems. Mobile Networks and Applications, 18(1), pp.129-140.

[15] M. T. Beck, M. Werner, S. Feld, and S. Schimper, Mobile edge computing: A taxonomy, in Proc. of the Sixth International Conference on Advances in Future Internet. Citeseer, 2014.

[16] T. Verbelen, P. Simoens, F. De Turck, and B. Dhoedt, Cloudlets: Bringing the cloud to the mobile user in Proceedings of the third ACM workshop on Mobile cloud computing and services. ACM, 2012, pp. 2936.

[17] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, Fog computing and its role in the internet of things, in Proceedings of the first edition of the MCC workshop on Mobile cloud computing. ACM, 2012, pp. 1316.

[18] O. Vermesan, P. Friess, P. Guillemin, S. Gusmeroli, H. Sundmaeker, A. Bassi, I.S. Jubert, M. Mazura, M. Harrison, M. Eisenhauer, M. and P. Doody, 2011. Internet of things strategic research roadmap. Internet of Things-Global Technological and Societal Trends, 1, pp.9-52..

[19] D. Singh, G. Tripathi, A. M. Alberti and A. Jara, "Semantic edge computing and IoT architecture for military health services in battlefield," 2017 14th IEEE Annual Consumer Communications and Networking Conference (CCNC), Las Vegas, NV, 2017, pp. 185-190.

[20] S. Tata, R. Jain, H. Ludwig and S. Gopisetty, "Living in the Cloud or on the Edge: Opportunities and Challenges of IOT Application Architecture," 2017 IEEE International Conference on Services Computing (SCC), Honolulu, HI, 2017, pp. 220-224.

[21] X. Zhou et al., Network Slicing as a Service: Enable Industries Own Software-Defined Cellular Networks, IEEE Commun. Mag., vol. 54, no. 7, Jul. 2016, pp. 14653

[22] G. Even, M. Rost, and S. Schmid."An Approximation Algorithm for Path Computation and Function Placement in SDNs". 2016

[23] L. Xu, G. M. OHare, and R. W. Collier, A smart and balanced energy-efficient multihop clustering algorithm (smart-beem) for mimo iot systems in future networks, in MDPI Sensors, vol. 17, 2017

[24] J. Zhao, C. Qiao, R. S. Sudhaakar, and S. Yoon, Improve efficiency and reliability in single-hop wsns with transmit-only nodes, IEEE Transactions on Parallel and Distributed Systems, vol. 24, no. 3, pp. 520534, 2013.

[25] M. F. Munir and F. Filali, Increasing connectivity in wireless sensoractuator networks using dynamic actuator cooperation, in Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE. IEEE, 2008, pp. 203207.

[26] L. Xu, J. Xie, X. Xu, and S. Wang, Enterprise lte and wifi interworking system and a proposed network selection solution, in Proceedings of the 2016 Symposium on Architectures for Networking and Communications Systems, ser. ANCS 16. New York, NY, USA: ACM, 2016, pp. 137138.

[27] O. Banimelhem, M. Mowafi, E. Taqieddin, F. Awad, and M. Al Rawabdeh, An efficient clustering approach using genetic algorithm and node mobility in wireless sensor networks, in Wireless Communications Systems (ISWCS), 2014 11th International Symposium on. IEEE, 2014, pp. 858862.

[28] K. Dolui and S. K. Datta, "Comparison of edge computing implementations: Fog computing, cloudlet and mobile edge computing," 2017 Global Internet of Things Summit (GIoTS), Geneva, 2017, pp. 1-6.

[29] J. Qadir et al., Artificial Intelligence Enabled Networking, IEEE Access, vol. 3, 2015, pp. 307982.

[30] R. Li et al., "Intelligent 5G: When Cellular Networks Meet Artificial Intelligence," in IEEE Wireless Communications, vol. PP, no. 99, pp. 2-10.

[31] "European Comission, 5G for Europe: An Action Plan", Communication from the Comission to the European Parliament, The Council and Regions and Economic Committees. Brussels, 14 SEP 2016 , Available: https://ec.europa.eu/digital-single-market/en/news/communication-5g-europe-action-plan-and-accompanying-staff-working-document, Accessed 30-OCT-2017

[32] I. G. Ben Yahia, J. Bendriss, A. Samba and P. Dooze, "CogNitive 5G networks: Comprehensive operator use cases with machine learning for management operations," 2017 20th Conference on Innovations in Clouds, Internet and Networks (ICIN), Paris, 2017, pp. 252-259.

[33] L. Xu et al., "CogNet: A network management architecture featuring

cognitive capabilities," 2016 European Conference on Networks and Communications (EuCNC), Athens, 2016, pp. 325-329.
[34] T. Bakhshi, 2017. State of the Art and recent research advances in software defined networking. Wireless Communications and Mobile Computing, 2017.
[35] J. O. Kephart and D. M. Chess, "The vision of autonomic computing," in Computer, vol. 36, no. 1, pp. 41-50, Jan 2003.