Fog-Enabled Multi-Robot Systems

Nader Mohamed[1], Jameela Al-Jaroodi[2] and Imad Jawhar[3]

[1]Middleware Technologies Lab., Pittsburgh, Pennsylvania, USA
[2]Department of Engineering, Robert Morris University, Moon Township, Pennsylvania, USA
[3]Midcomp Research Center, Saida, Lebanon
nader@middleware-tech.net, aljaroodi@rmu.edu, imad@midcomp.net

***Abstract*— Fog computing has been proposed lately to offer services for the Internet of Things (IoT) thus adding many benefits for IoT applications. These benefits include support for low latency needs, mobility, location awareness, scalability, and efficient integration with other systems such as cloud computing. This paper investigates employing fog computing to enable multi-robot system (MRS) applications. It also discusses the potential services fog computing can offer MRS applications. Furthermore, the paper categorizes MRS applications highlighting the different fog support needed for each category. The paper classifies MRSs into five different categories based on the type of network used. This classification simplifies the process of understanding the issues involved in utilizing fog computing for different types of MRS applications. This classification also helps identify the most suitable fog computing architecture for each category. Then the paper proposes a service-oriented platform to support fog-enabled MRS applications and discusses its prototype implementation.**

Keywords- Fog Computing, Multi-Robot Systems, Cloud Computing

I. INTRODUCTION

Various innovative robots were created and more will be created in the coming years to partially or fully automate many activities and processes in different applications. These robots are designed to enhance productivity, accuracy, quality, reliability, flexibility and cost-effectiveness of these applications. Robots come in different types and have different capabilities. Some of the common types of robots include Humanoids, on-ground fixed robots, Unmanned Ground Vehicles (UGVs), Unmanned Aerial Vehicles (UAVs), Unmanned Surface Vehicles (USVs), and Unmanned Underwater Vehicles (UUVs). They can also be stationary and do not change their locations or mobile to be available in different locations to serve certain objectives. They can have varying actions, sensing, processing, communication, autonomy, and learning capabilities.

There are numerous applications that can use multiple robots to accomplish their objectives efficiently and cost-effectively. Multiple robots used for these applications are connected in what is called a Multi-Robot System (MRS) and work together to complete a certain task. These robots can all be of the same type or of different types. For example, they can all be UAVs with the same capabilities or they can be a collection of UAVs, USVs, and UUVs in one system. They can all be fixed, mobile, or a mix of both and may be homogenous or heterogeneous in their equipment, operations and capabilities. One of the main benefits of using MRSs is lowering the time needed to complete a task. One example application is using collaborative UAVs for search and rescue operations to significantly reduce the time needed to complete some lengthy tasks like searching for survivors across large geographical areas. In this case, for example, using 20 well-coordinated UAVs will cut the required time for search operations to around 5% of that required by one UAV [1].

Utilizing MRSs in various applications offers many additional benefits; however, designing, implementing, and operating such systems is not trivial as there are development and management issues involved including synchronization, coordination, resource allocation, and planning. These are generally not easy to address due do the diverse system and network challenges. Furthermore, there are high needs to integrate robots with other systems such as cloud computing [2][3] or other external systems such as remote storage systems. This integration is needed for many applications to utilize scalable, flexible, and cost-effective services provided by the cloud or other systems. However, such integration is also not trivial due to many issues [4]. One of the most challenging issues in this integration is the high communication delays between the MRS and the cloud that makes the utilization of cloud services less effective for some applications and situations. In addition, there are security worries in integrating robots with the cloud. This type of integration requires maintaining suitable security measures for the applications, the cloud infrastructure, the MRS, and other connected systems.

Meanwhile, fog computing has recently been proposed to provide solutions for many challenges facing many IoT applications. Fog computing is a model that offers small and distributed platforms placed at the network edges to provision diverse types of services for IoT applications [5]. These small platforms are placed at the edge of the network to be closer to where the IoT applications data is produced and where IoT applications tasks should be achieved. Comparably, fog computing can also provide similar benefits for MRSs. The main goal of this paper is to investigate and discuss the services and benefits that can be offered by integrating fog computing with MRS applications. We also investigate different possible fog computing architectures to support different types of MRS applications. Our research approach in this regard is to classify the MRSs in different categories based on their network types. Based on these categories we investigate different possible fog computing architectures that can be used to provide suitable support for the different MRS applications within each category. In addition to investigating possible services, benefits, and architectures of fog computing for MRSs, we also discuss the different issues involved in employing fog computing for MRSs and propose a service-oriented platform to support fog-enabled MRS applications and discuss its prototype implementation.

The rest of this paper is organized as follows. Section II discusses background information and related work about MRSs and fog computing. Section III lists and discusses potential fog services and their benefits for MRS applications. Section IV classifies MRS applications and discuss potential fog architectures and services for the different categories. Section V describes a service-oriented platform for fog-enabled MRS applications, while a prototype

978-1-5386-6488-9/18/$31.00 ©2018 IEEE

implementation of this platform is discussed in Section VI. Some discussion of the issues involved in employing fog computing for MRSs is offered in Section VII and Section VIII concludes the paper.

II. BACKGROUND

To be able to discuss the opportunities and benefits of utilizing fog computing to support MRS applications, it is necessary to identify the main characteristics of both.

A. Multi-Robot System (MRS)

Although there are lots of tasks where a single robot can be utilized to provide effective automation, there are some tasks where it is inefficient or sometimes impossible to use a single robot to provide effective automation [6]. This is due to the characteristics and requirements of these tasks such as:

- The task involves many functions requiring different capabilities from the robots. As a result, it is difficult to be completed by a single robot. Advanced robots with multiple functions can be designed for these tasks, however this can be very costly.
- The task is widely distributed making it ineffective or difficult to achieve using a single robot. In this case, a single robot will not have full coverage of the whole area or will require a long time to complete the task. A highly mobile robot may have the capability to cover larger areas or work faster, yet it will need extended time to complete the tasks and it could also be very expensive.
- The task is considered a critical mission task and needs very dependable operations to be achieved within the defined operational parameters. Employing a single robot to achieve this type of task creates a single point of failure, which may lead to delays or failure of the mission.

Multiple robots can collaborate to effectively accomplish these types of tasks. The collective robots used to achieve these tasks represent a MRS, which offers the following benefits:

- Multiple robots with the same capabilities can simultaneously work on a task to finish it faster.
- Multiple robots heterogeneous in their capabilities and functions can deliver a cost-effective solution to achieve a task requiring different functionalities. This can eliminate the need to build costly powerful multi-function robots.
- Multiple robots can effectively and efficiently execute a task that is widely distributed by dividing the work areas among them.
- Multiple robots working on a task provide fault tolerance to compensate for the loss of one or more robots in the system. A faulty robot can be substituted by standby robots in the system or by reorganizing and reconfiguring the other robots to take over the responsibilities of the faulty robot. This permits the MRS to continue working to achieve its tasks with minimal interruptions.

There are many real-life applications that can benefit from MRSs like search and rescue [7][8], detecting and fighting forest fires [9], hazardous waste removal [10], agriculture [11], mining [12], construction [13], disaster management [14], security applications [14][15], warehouse management [16], moving containers in harbors and airports [17], and playing soccer [18]. MRSs can generally be divided into two types, collective systems and intentionally cooperative systems [6]. Robots in collective systems perform their required tasks with minimum interactions with each other. Therefore, there is no or minimum communication among the robots in the system as they operate. The robots in a collective system are usually homogeneous. In contrast, robots in an intentionally cooperative systems, interact and communicate with each other to coordinate their tasks towards a common goal. In this type of systems, knowledge about the states, actions, and capabilities of the robots in the system as well as the progress towards the common goal is maintained. The robots in this system can be homogenous or heterogeneous in their design, capabilities and functions. There are some common and important features required by many MRS applications including:

- Cooperative manipulation: multiple robots capable of working together to achieve a specific action.
- Multi-target observations: utilizing multiple robots to monitor multiple related targets.
- Exploration: using multiple robots to discover and learn about the operational environment rapidly.
- Formation control: sustaining certain relative positions among the robots as they move through the operational environment.
- Motion planning: creating optimized plans for multiple robots' movements and navigation.
- Flocking control: allowing multiple robots to travel together within in specific formations and on defined paths.
- Traffic control: preventing collisions and deadlocks among the robots.

There are several methods suggested for these essential requirements [6]. Generally, all these requirements need suitable task allocation mechanisms [19]. Moreover, some of these requirements should have innovative leaning algorithms to permit the MRS to learn from former experiences to improve its actions [20].

B. Fog Computing

Many IoT applications need to connect to the cloud and other systems for various reasons. One of these reasons is to utilize functions and services offered by these systems such as elastic resources at low cost. IoT applications can use the cloud for data storage, powerful processing, or to conduct other advanced services that apply smart algorithms for new observations, optimization, forecasting, scheduling, and planning. Nevertheless, linking IoT applications to the cloud has many restrictions as the cloud cannot provision for some of the essential requirements and characteristics of IoT applications such as highly heterogeneous devices, low-latency responses, mobility, and location awareness.

To overcome these restrictions, Cisco introduced the concept of fog computing [5]. Fog computing can improve the cloud computing paradigm by offering small platforms placed at the network edges closer to the IoT. These platforms offer cloud-like services to provision different IoT operations. The services can be control, storage, communication, processing, configuration, monitoring, measurement, and management services. Utilizing fog computing, an application in a certain area can use an architecture that has a dedicated computer available locally, or one or more clients or nearby edge devices. The Fog platform facilitates executing services geographically close to the IoT applications and at the same time it can use services provided by the cloud. This provides several benefits for IoT applications including [5]:

- Offering low latency services
- Offering location aware services
- Offering better scalability for geographically widely distributed applications

- Supporting streaming communication and processing
- Supporting better mobility and access control
- Offering better Quality of Services (QoS) support
- Offering more efficient communication with other systems

These benefits allow for creating solutions to many challenges that IoT applications face and enable the creation of higher quality and more controllable services to achieve the vision of IoT. The architecture of integrating IoT, fog computing, and cloud computing is shown in Figure 1. In this architecture, the fog will provide more localized real-time monitoring, control, and optimization for the IoT applications while the cloud will provide global monitoring, control, optimization, future planning, elastic resources, and other advanced services for these applications.

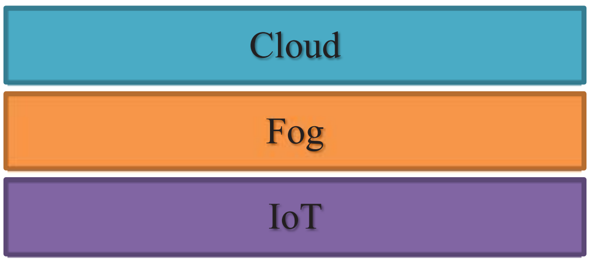

Figure 1. Three-layer architecture (IoT, fog, and cloud).

There is some research investigating and reviewing the use, design and experiments on using fog computing. Stojmenovic and Wen [21] and Shanhe et al. [22] reviewed fog computing advantages and issues. Satyanarayanan et al. [23][24] introduced Cloudlets, a platform that has similar roles to fog computing. Hong et al. [25] introduced a high-level programming model for fog computing platforms. Willis et al. [26] introduced ParaDrop, a fog platform provided over a wireless router. Ha et al. [27] applied Cloudlets to design and implement a real-time wearable cognitive assistant on Google Glass. Zhu et al. [28] introduced utilizing a fog to enhance websites performance. Cao et al. [29] employed fog computing to allow analytics for pervasive health monitoring such as timely fall detection. Hassan et al. [30] examined fog computing for processing offloading and storage expansion for mobile applications. Security and privacy issues in fog computing have been reviewed by Stojmenovic et al. [31] and Dsouza et al. [32].

III. Fog Services for MRS

MRS present some similar features and requirements as IoT systems in addition to their unique aspects. Just like its support for IoT applications, fog computing can provide many services for MRS applications in a similar manner. When a MRS is deployed, fog nodes in nearby locations can be used to provide the necessary functions supporting the MRS operations. Depending on the needs of the MRS, one or more fog nodes can be made available at specific locations that would offer the best localization and benefits. As such, the fog nodes will provide multiple services for the MRS. Many of these are very similar to services needed by other environments like IoT applications. However, the following discussion is focused mainly on MRS applications.

- **Computational Services:** As most robots may not have high processing power, they can offload their processing needs to nearby fog nodes. Locally collected data may require some compute intensive analysis or need specialized algorithms. In this case, the fog nodes will perform these operations and send back the results to the robots. In addition, multiple fog nodes may be utilized to provide concurrent processing for the MRS when necessary.
- **Storage Services:** Robots usually have some storage capacity; however, it is also limited in most cases. Some MRS applications require gathering huge amounts of data in the form of video recordings, high definition imaging, and continuous sensory data collection. Over time it becomes necessary to transfer this data to other platforms to free up the space for more data. If the data is needed in real-time it will be more practical to store it nearby. Therefore, fog nodes can provide the storage capacity needed for this data. In addition, with multiple fog nodes it will be possible for each robot to transfer its data to the closest node, which also reduces energy consumption.
- **Communication Services**: Robots may need to exchange information with remote locations, which can strain their power sources as they frequently use long range transmissions. Instead, the robots can find a nearby fog node to send their data to and that fog node then can relay the data to the required destination. Fog nodes can also pass data back to the nearby robots from these remote locations. This reduces the energy use in the robots and can offer additional benefits such as Quality of Service (QoS) provisioning, streaming communication, and data pre-processing, filtering and fusion.
- **Caching Services:** It may be necessary for some MRSs to retrieve and use large data sets from the cloud frequently. Given the long distance between the cloud and the MRS, the unpredictable network conditions between them, and the possible delays in transmissions getting the required data will take a long time. This can negatively impact the performance of the MRS especially if the data sets are very large or they are repeatedly needed and can't be fully stored on the MRS robots. Fog nodes can download the required data sets for the robots and then act as cache units to hold the data sets the MRS needs over time. Therefore, when a robot in the MRS needs a specific part of that data set, it should find it in the cached copies on the fog nodes. This reduces the time needed to retrieve the data and allows the MRS to operate faster.
- **Security Services:** The frequent exchange of data between MRS robots, the cloud and other connected systems create the need to protect and secure this communication. When fog nodes are used to support these communications, they can also be designed to add security features to these communications. Different services offering various types of secure communication and different levels of security can be made available on fog nodes. This allows the MRS applications to incorporate the necessary security services into their system based on their requirements.
- **Monitoring and Control Services:** An MRS application needs to keep track of its robots as they operate. Information about their tasks, locations, progress, etc. is continuously recorded and analyzed. In addition, based on the analysis and also on the defined goals of the MRS application, the MRS components must be controlled and guided to successfully accomplish their tasks. One example of controls is task allocation and re-allocation as the robots operate. Fog nodes, being very close to the MRS components can provide monitoring and control services that allow monitoring data to be collected and analyzed on the fog node and relay control information to the MRS components when needed. Several levels of analysis can be

supported on the fog nodes from simple sense and respond types to high level intelligent analysis and learning models.

- **Global Reference Time (GRT) and Synchronization Services:** An MRS having multiple components operating, usually in different locations, requires a common time reference to ensure accurate operations. A global reference time needs to be maintained for all MRS components. A fog node close to the MRS can provide this GRT for the robots to allow them to coordinate their actions. In addition, as a collective, the robots in a MRS application need to synchronize their actions for accurate and safe operations. For example, it is sometimes important to ensure that certain tasks are completed before others can start. With the GRT and synchronization services available on connected fog nodes, these operations can be orchestrated by services on the fog nodes.
- **Mobility Support:** Some robots in a MRS may be mobile and change location as they operate. Fog nodes available in different areas in the operating environment can help locate and keep track of moving robots. They can also provide the required services to these robots based on their location. In a very similar manner to cellular networks, fog nodes will become like the base stations for those moving robots thus facilitating tracking, handoff, and continuous connectivity.
- **Navigation Services:** with mobile robots in a MRS, we need to ensure correct navigation, collision avoidance and deadlock among these robots. As fog nodes can keep track of each moving robot and their location and status information, they can also provide services that will monitor and direct these robots. Services to create and follow specific paths, avoid collisions with other robots and resolve deadlock situations can be built and deployed on the distributed fog nodes. These services usually require high processing power and knowledge about all robots, thus it is not feasible to rely on the robots' onboard computing facilities to do them.
- **Scalability Support:** In many situations, applications may expand rapidly and require the use of more robots, services and other components in the system. The operating environment may also grow and require more resources to cover effectively. This leads to higher complexity and higher demand on storage and compute resources to manage and control the MRS. Fog nodes distributed across the operating environment can be organized in a hierarchical structure to provide these resources and facilitate the required services for the whole system. In addition, it also becomes possible to add more fog nodes to the hierarchy to extend their capabilities and maintain the required levels of performance. For example, with this structure, hierarchical control services can be provided [33] and as illustrate in [34] this structure can enhance efficiency and scalability.
- **Integration Services:** Generally a MRS application will not operate in a vacuum. There is always the need to connect with other systems that will either provide services or data to the MRS or use results generated by the MRS. Therefore, services to integrate the MRS applications with other systems are necessary. Fog nodes are a good candidate to offer the integration services that act as the gateway between the MRS and the other systems. Examples of integration services are interoperability, communication filtering, and security.

IV. Fog Architectures for MRS Categries

Since we have different types of services fog computing can provide to support MRS applications, it is necessary to have suitable fog architectures to efficiently support these different types. Different fog architectures have different performance impact for different types of MRS applications. To identify the suitable architectures, we first provide a classification for the MRS applications based on the network types they use. This will help understand the common characteristics of the MRS applications within each category and highlight the common challenges facing each category to help address them using fog computing solutions. Our classification defines five MRS categories: Nanoscale MRS (NMRS), Body Area MRS (BMRS), Local Area MRS (LMRS), Mobile Ad Hoc MRS (MMRS), and Wide Area MRS (WMRS).

A. Nanoscale Multi-Robot Systems (NMRS)

Nanoscale robots are systems at the nanoscale level measuring between 1 and 100 nanometers [35] and constructed of nanoscale or molecular components [36]. These robots have nanosensors, nanoauctotors, and other nanocomponents that are connected using nanonetworks. Nanorobots are usually designed to interact with nanoscale objects. When multiple nanoscale robots collaborate to achieve a specific task, we get the Nanoscale MRS (NMRS). These nanorobots have many useful applications many areas like medicine for example. Examples of these applications are to recognize and terminate cancer cells [37], to deliver drugs to a targeted area [38], and to help control diabetes [39]. Fog computing can provide several services for NMRSs; however, in this paper, we will not discuss this category further since it has some unique characteristics that need to be addressed separately. In addition, nanoscale robot technologies are still under research and development. Nevertheless, we mentioned it here to have a comprehensive classification for all MRSs categories.

B. Body Area Multi-Robot Systems(BMRS)

This category of MRSs is used within a single mobile body such as a human or a mobile structure. A mobile structure can be a ship, a plane, a submarine, or a food truck. This configuration of connected robots for a Body area MRS (BMRS). All robots in a BMRS are always available within their corresponding mobile structure. A BMRS in this category consists of multiple robots and other devices connected through mobile networks available within the body they operate within. Robots can communicate and collaborate among themselves and with other available devices. They can communicate using wired or wireless networks or using both. One example of a BMRS is using multiple wearable robots connected as a system to help disabled people with multiple disabilities to move and function. In this application, multiple wearable robots can be attached to different body parts to facilitate movement or allow for better body functions. These wearable robots need to communicate, synchronize, and collaborate to function correctly. Another example is a BMRS on a mobile structure like an advanced warship equipped with multiple offense and defense systems supported by multiple machines, robots, and other devices. In various occasions, these robots will need to coordinate their operations to achieve the desired task.

In this type of system, a BMRS, one fog unit installed within the body can usually provide most needed services for the BMRS. This fog unit connects to the BMRS components through the available network technology and provide the needed processing, storage, communication, security, control, synchronization, and monitoring services. In addition, the fog node can simplify the integration of the BMRS with other systems within the body or outside the body such as cloud computing services or specialized systems like a medical system in the case of a health related BMRS. The fog node will be connected wirelessly with the external systems through Wi-Fi, GSM, Wi-Max, or Satellite communication. Usually a single fog node is sufficient to cover the required services for a BMRS. However, it is recommended to have multiple fog nodes for better performance, reliability, fault tolerance and scalability. This also allows for avoiding a single point of failure at the fog node.

C. *Local Area Multi-Robot Systems (LMRS)*

This category is available in a limited geographic area such as within a building, a house, a soccer field, a farm, or other small facilities. When multiple robots are connected using a local area network (LAN) and collaborate within this area to achieve a given task, we have a Local Area MRS (LMRS). The robots can be fixed or mobile and may use wired, wireless or both networks to communicate. In addition, these robots can communicate among themselves and with other devices like sensors, servers, multimedia systems that are available in the area. Examples of a LMRS are manufacturing robots, soccer playing robots, and multiple collaborative agricultural robots. These LMRSs can consist of homogenous robots such as in the soccer playing robots or can have heterogeneous robots such as in the manufacturing robots, where multiple robots with varying capabilities and functionalities collaborate to build a product in an assembly line.

In this category, one or more fixed fog nodes can be used to support a LMRS. The number of fog nodes used depends on the size and requirements of the LMRS application. Multiple fog nodes can be used to provide better performance, reliability, fault tolerance and scalability. These fog nodes can provide several services for processing, storage, communication, security, control, synchronization, monitoring, and integration. The multiple fog nodes can also be homogenous or heterogeneous in their capabilities to support different types of functionalities and requirements for the LMRS. For example, only one node may be needed to provide integration with external systems, while all other nodes are used for localized operations and controls. In addition, some nodes may need to have specific capabilities based on their location or nature of services they offer to the robots. The mobile robots of the LMRS can be connected wirelessly to one or multiple fog nodes, while fixed robots and other devices can be connected through wireless or wired networks.

D. *Mobile Ad Hoc Multi-Robot Systems (MMRS)*

A MMRS is formed when some or all the robots in the MRS are mobile and connected wirelessly in a mobile ad-hoc network (MANET) using direct or multi-hop communication. Nodes in this category can be mobile robots, mobile devices, or BMRSs: mobile structures that have several connected robots within the structure. There is usually no fixed structure that can be utilized to provide communication and other needed functions for the MMRS since the communication links and network topology used are usually dynamic. This means that direct links between any two nodes in the system can be available for some time but not all the time as nodes are moving and their limited wireless communication range will not allow them to maintain continuous connections with remote nodes. Robots can communicate directly if they are within their wireless communication ranges, yet as they move around, they go in and out of this range, thus requiring multihop communication. In some cases, MMRSs can be supported by geographically fixed facilities to support their communication needs. These fixed facilities can be road side units (RSU) for vehicular networks [40], GSM base stations enabling cellular communication, or any other fixed facilities. In other situations, with the absence of such facilities, the robots in a MMRS will need to form and maintain their own ad hoc network. Application examples of a MMRS that only consist of multiple mobile robots are collaborative UAVs, collaborative AUVs, collaborative autonomous boats, and collaborative self-driving cars. An application example of a MMRS that consists of mobile robots and fixed structures can be found in war situations where multiple UAVs, AUVs, and robot-equipped warships need to collaborate to achieve certain missions.

Fog computing can be utilized to provide several services for MMRS applications. However, unlike in a BMRS or LMRS, we need different fog architectures to better support the applications of a MMRS. This is mainly due to the high mobility and the dynamic characteristics of the communication connections. To understand these architectures, we further classify MMRSs into three types:

- Type 1 - All nodes in a MMRS can communicate with a fixed communication infrastructure and they can also communicate among themselves through their MANET. Here, one or more fog nodes can be placed in these mobile robots/structures and/or at the edges of the communication infrastructure. The number of fog nodes and their locations depend on the type of the MMRS applications and the capabilities of the MMRS robots.
- Type 2 - Some of the nodes in a MMRS can communicate with a fixed communication infrastructure while all nodes can communicate among themselves through their MANET. In this type, one or more fog nodes can be placed on any of the mobile robots/structures or at the edges of the communication infrastructure. However, if the fog node is placed at the fixed points, not all robots/structures will be able to communicate directly with them. The number fog nodes and their locations depend on the type of the MMRS applications.
- Type 3 – None of the nodes in a MMRS can communicate with any fixed communication infrastructure while they can communicate among themselves through their MANET. In this type, selected mobile robots/structures can have fog nodes to support the MMRS applications. The selected numbers of fog noes and their location depend on the application, the number of used nodes, the number of mobile robots/structures, and the locations of the nodes.

In all these types, fog computing can provide several services for MMRSs including processing, storage, communication, multihop routing, security, control, synchronization, monitoring, and integration services. In addition, it can also provide services for navigation. However, depending on the type of MMRS being supported, specific requirements are imposed on the fog nodes and their hosts due to the limitations these robots may have in terms of payload, energy and processing and communication capabilities.

E. *Wide Area Multi-Robot Systems (WMRS)*

In this category, we are dealing with application components that extend over a widely distributed geographical area. Multiple robots and devices are distributed over the large area to accomplish their objectives. Thus, they require wide area communication coverage resulting in a WMRS. A WMRS can consist of many robots and/or other types of MRSs such as BMRSs and LMRSs connected through a wide area network (WAN). A WMRS can be viewed as a network of networks or a network of systems. One application example is geographically distributed, fully automated collaborative manufacturing plants. Another example is a city-wide sky defense system that utilizes multiple collaborative military robotics and machines working together to provide complete protection for the city from missile and other air attacks. In addition, applications that involve widely distributed multiple robots that can be controlled through the cloud or the Internet can also be considered good examples of a WMRS.

In a WMRS, multiple fog nodes distributed in a hierarchical structure can provide better operations and scalability. In addition, different fog services can be more efficiently offered for WMRS applications. For example, if a WMRS application consists of several connected LMRSs, then each LMRS will have its local fog node(s) to provide local support. At the same time, other fog nodes can be made available to provide links between the different LMRSs and the services to support the whole system. These fog nodes can provide processing, storage, communication, security, control, synchronization, monitoring, integration, and other services to the WMRS applications.

V. Proposed Fog Platform

In this section we propose a service-oriented platform for MRSs named MRSFog. The key aim of MRSFog is to offer a software environment to be utilized to implement and execute MRS applications supported by fog computing. In this platform, all functions are seen as a collection of services to be used for implementing and provisioning the execution of diverse MRS applications. These services are divided into two sets: core services and environmental services. Core services are those designed precisely to achieve the fundamental operations of the platform. Examples are the broker, security, invocation, and location aware services. Environmental services enable access to services offered by the cloud, distributed fog nodes, and robots making up the MRS applications. Cloud services examples are IaaS (Infrastructure as a Service), PaaS (Platform as a Service), and SaaS (Software a Service), which offer advanced software services for MRS applications such as machine learning, planning, simulation, data mining, big data analytics, and optimization. Fog services are more localized and can offer monitoring and control, data caching, communication, streaming, computation, and management services. Robot services provide interfaces to robots functionalities like sensing, actions, and navigation. The environmental services can either offer direct interfaces to apply the original services offered by the cloud, fogs, or robots or they add values to the original services such as adding dependability and security features. While some robots can perform on-board processing to offer some services, other robots will be controlled mainly by an external computer or fog nodes that will have services that provide interfaces to utilize these robots' functionalities. The proposed platform services can be used by applications available on the cloud, fog nodes, or robots such as a UAV asking for a specific service from an application offered on the cloud.

The key functions of the MRSFog platform are to allow the smooth integration and operations among all local robots' services, local MRSFog services, remote MRSFog services, and cloud services to effectively support MRS applications. It should allow any service or application available on the cloud or on one of the MRSFog nodes to utilize all services available in the environment. Each service defines its own simple interfaces that make it available to other services. Using the service-oriented architecture for the MRSFog platform provides a flexible approach to link available services and to develop new ones following the same standards. A selected subset of available distributed services can be integrated to accomplish a specific MRS application. The MRSFog platform helps achieve loose coupling among interacting services. It also manages service advertisement and discovery, communication, and invocation. In addition, it can be used to implement collaborative services across multiple robots and with other service-oriented systems.

The MRSFog platform can have many core services; however, the most essential services for its operations are robot resource discovery and integration, broker, invocation, location-based and security services. These five service types are important to operate the platform and support the creation of MRS applications as more environmental services are introduced.

A. *Local Robots Discovery and Integration Services*

An MRSFog node can be part of powerful mobile robots that can move and relocate at different locations to support different MRS applications, where other robots and IoT devices are also present in the same area. In this case, it is important that the MRSFog node is able to discover the available local robots and IoT resources, their locations, and their capabilities to integrate and operate with them effectively. Different approaches can be used to simplify the discovery, configuration, and integration with the available robots and IoT devices. One of the best approaches is the Universal Plug and Play (UPnP) approach [41]. With this approach, discovery, configuration, and integration of all available devices can be easily automated. The locations of other robots and IoT devices can be provided using the devices' GPS capabilities or estimated using different location estimation techniques [42].

B. *Broker Services*

Each MRSFog node has a broker and a set of broker services that are responsible for the MRSFog, local robots, and IoT devices services advertisement, registration, and search. All discovered local robots and IoT devices will have corresponding services that enable their utilization, which must be registered with a broker. The broker of each MRSFog node is considered a local broker and only maintains information about the current available services within its node and the services provided by associated robots and IoT devices. This approach is used to allow applications and services within a MRSFog node to use the offered services and resources and deliver low latency responses. Consequently, the time required to realize services is minimized to efficiently use the services. The MRSFog broker will also keep information about other available brokers in the environment. These brokers may belong to other MRSFog nodes or the cloud. All MRSFog brokers will maintain information about their local resources associated with them, while the cloud broker will maintain information about resources and services provided by the cloud. Any application or service requiring to use a certain service will lookup that service at the local broker. If the local broker does not have the service, it will forward the lookup request to other known brokers in the environment. In some cases, the MRSFog node may need to forward the request to the cloud broker to access services on the cloud. Each of the brokers maintain the description information using Web Service Description Language (WSDL) for each defined service. Each service description includes the functions the service can achieve, the detailed types and formats of the messages it exchanges with the service providers and consumers, and the network addresses of these entities.

C. Invocation Services

Invocation Services, local and remote, are initiated by services consumers with the MRSFog support. Remote service invocation can be between a MRSFog service and another MRSFog service, between a robot and another robot, between a robot and an IoT device service, between a robot and a MRSFog service, between a MRSFog service and a cloud service, or between a cloud service and a MRSFog service. The MRSFog platform handles message addressing, data marshaling and demarshalling, establishing communication connections between services consumers and producers, delivere requests and responses, and execute services.

D. Location Based Services

The MRSFog platform offers location-based services by relying on the brokers to keep additional location information about the MRS components. Unlike regular service brokers on the Internet, the MRSFog brokers record the current locations of presently associated mobile robots and mobile IoT devices. This information facilitate services that can only be active or useful if available at a specific location. An example is utilizing a sensor service on a robot if the robot is positioned within a given location. A service consumer in a MRSFog node can look up a service based on location through its MRSFog broker. If this service is accessible within the MRSFog node's range, then WSDL information about the service is directed to the consumer to utilize the service using local service invocation. Otherwise, the MRSFog broker will forward the lookup request to other brokers on the cloud or other MRSFog nodes. If the service is present on any of these within the required location, the service consumer can use it with remote service invocation. The cloud broker records location information and available services of all MRSFog nodes to find specific location-based services anywhere.

E. Security Services

Several security features can be applied by different clouds, MRSFog nodes, robots, and IoT devices in MRS applications. The major roles of security services in the MRSFog platform are to integrate and normalize security features among all these systems and guarantee that the needed security measures are applied properly to protect the MRS applications, used devices, offered services, and the physical environments. The security services consist of authorization and authentication services and access control services for the MRS applications, MRSFog services, cloud services, robots' services, and the physical environment. These services can be offered with different levels of protection measures such that diverse applications can utilize the appropriate set for their security needs.

VI. PROTOTYPE IMPLEMENTATION AND EXPERIMENTS

A MRSFog prototype was implemented using a distributed Java agents environment [43][44]. This environment provides a middleware infrastructure that enables implementing different distributed programming models and advanced platform services for distributed heterogeneous environments. These heterogeneous environments may consist of multiple heterogeneous systems and nodes with different capabilities. In addition, this middleware infrastructure provides scalability solutions to deploy, operate, monitor, and control large-scale distributed applications. Each agent in the environment can provide runtime support for securely executing needed functions. This makes this middleware infrastructure very suitable to build advanced platform services for integrating the cloud with its powerful resources with one or more MRSFog nodes that usually have limited resources. For the prototype implementation, the core services of the MRSFog platform were built. These consist of robots' discovery and integration services, broker services, invocation services, and location-based services. In addition, a function is added to enable a local MRSFog broker to forward a service lookup request to other available brokers if it does not have the requested service. Both local and remote service invocations were also included.

For the robots side, we employed the Arduino board [45], an open source hardware used for embedded system applications prototypes. For our experiments, the Arduino was used as the robot payload subsystem that may request some services form external systems. Coupled with the Arduino are some sensors and actuators such as the DHT11 sensor [46] and LEDs and a buzz. Moreover, an Adafruit CC3000 Wi-Fi board [47] was installed to enable communication between the Arduino and the local area network connecting the MRSFog. The Arduino code was developed using the Arduino integrated development environment (IDE) [48] supported by the Adafruit CC3000 library [49]. Each robot service was built with a RESTful API.

At the MRSFog side, there is a service representing each sensor or actuator associated with the Arduino. The main role of these services is to map a call from the SOAP APIs to RESTful APIs. All sensor and actuator services are registered with the local MRSFog broker. For the experiments, we utilized three machines; one represents the cloud while the other two represent two MRSFog nodes. Furthermore, we utilized WAN emulators among the machines to introduce the effects of applying long distance and/or the Internet to link the machines. Experiments were conducted for diverse configurations:

- Call1: a local robot service call within the local corresponding MRSFog node.
- Call2: a remote robot service call from the cloud.
- Call3: a remote robot service call to another MRSFog node, where both MRSFog nodes are connected with a WAN and not involving the cloud.
- Call4: a remote robot service call to another MRSFog node through the cloud.

The experiments were repeated for two types of services. The first is to obtain the current temperature reading (CurTemp) and the second to switch on the LED (LEDon). The average results of 10 calls are presented in Figure 2. These average times do not contain the service search times. The response time for a service call from a MRSFog node to another MRSFog node and from the cloud to a MRSFog node are similar as both types have to travel through the WAN in a similar manner. However, the service call from one MRSFog node to another through the cloud adds extra time due to the indirect communication.

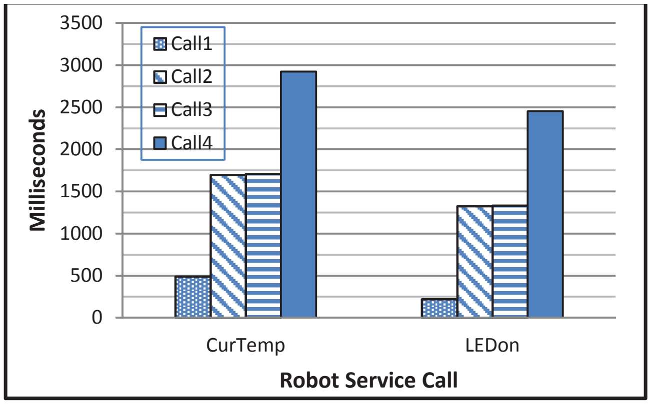

Figure 2. Robot service calls response times.

The average service search time for local services (Local), remote services between MRSFog node and the cloud or between MRSFog node and another MRSFog node linked by a WAN (Remote1), and remote services between MRSFog node and another MRSFog node through the cloud (Remote2) are presented in Figure 3. There is a major variance between Local and remote (Remote1 and Remote2) service search times. The local services can search for and use the local MRSFog node services and local robot services quicker. This allows having low latency services maintained by the available MRSFog nodes for the MRS applications. Simultaneously, local services can utilize services at the cloud or at other MRSFog nodes and vice versa when needed. However, at higher latency levels.

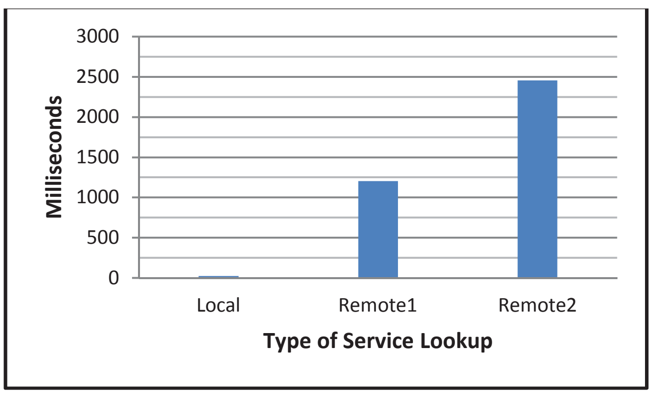

Figure 3. Service lookup times.

Another experiment was organized to measure and compare the response time of retrieving stored small-sized information by a robot from a local MRSFog node and from the cloud. The average results of 10 retrievals response times were recorded as shown in Figure 4. Here, the average response time for retrieving the information form the local MRSFog node is significantly shorter than the average response time of retrieving the same information from the cloud. This experiment demonstrates the benefit of utilizing local MRSFog nodes to provide temporary storage services for robots and MRSs for faster access.

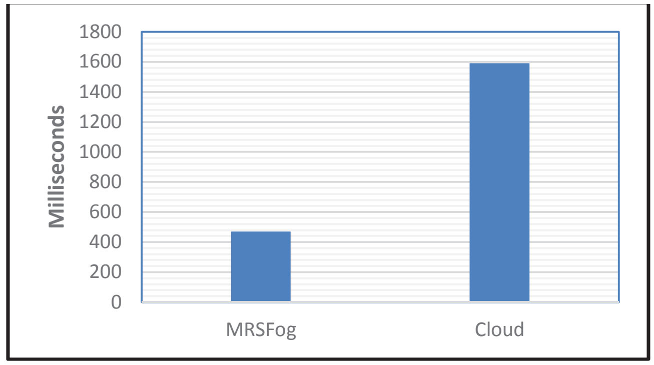

Figure 4. Information retrieval time.

VII. DISCUSSION

Fog computing can provide many useful services to support the operations of MRS applications. As we discussed, fog computing can have different forms and architectures to support different MRS types. Fog nodes can be made available on advanced network routers, communication infrastructure base stations, microcontrollers, mobile robots, mobile structures, or dedicated computers. The choice of number, type and location of fog nodes is highly dependent on the category of the MRS application. In addition, the fog nodes can take different organizational structures such as a flat or hierarchical structure. This depends on the size of the MRS application and the application category as discussed in Section IV. Generally, the flat structure is good for small MRS applications where a small number of robots are used and a limited number of fog nodes are needed. However, for large-scale MRS applications where many robots are used, a large geographical area is covered, and several fog nodes will be needed at different level. Table 1 provides a summary of different MRS categories, their applications, and suitable fog node types and architectures.

A lot of research is being done on utilizing cloud computing to support robots, MRSs, and their applications [2][3][4]. The efforts show the important role of cloud computing for progressing in the robotics field. Ultimately, cloud computing alone may not be enough to bring out the full potential in MRS applications. Many MRS applications require low latency and location aware functionalities to achieve their goals effectively and within required time frames. As a result, an additional component is necessary to augment the gap between the MRS components and the cloud. In this paper, we highlighted the importance of fog computing and its services for MRS applications. Fog computing will not replace cloud computing to support robots and MRS applications, but will play an important role complementing and supporting cloud services in MRS applications. Fog computing will provide solutions for some challenges facing cloud services supporting such applications. For example, supporting the required processing and storage needs to offer low-latency services locally rather than going to the cloud. At the same time, fog nodes will still be unable to offer the powerful processing and huge storage capabilities of the cloud, yet it will create an effective model to connect these resources to the MRS components when needed. Table 2 provides some comparisons between fog and cloud computing including the type of suitable services they can provide for MRS applications.

Table 1. MRS categories, applications, and fog Support.

MRS Category	MRS Applications	Fog Support
Nanoscale Multi-Robot Systems (NMRS)	• Recognize and terminate cancer cells robots • Targeted drug delivery robots • Diabetes control robots	One fog node is generally enough. Special technological development is needed to provide such fog platform and support.
Body Area Multi-Robot Systems (BMRS)	• Wearable robots • Warship robots • Submarine robots	One or more fog nodes attached to the mobile body can be used. Fog nodes can be executed on microcontrollers and/or advanced mobile routers attached to the mobile body.
Local Area Multi-Robot Systems (LMRS)	• Soccer playing robots • Farm operations robots • Warehouse management robots • Construction robots • Manufacturing robots • Container transportation robots	One or more fixed and local fog nodes can be used. Fog nodes can be on computers or advanced network routers. Fog computing and supporting services can be provided within the same LAN.
Mobile Ad Hoc Multi-Robot Systems (MMRS)	• Collaborative UAVs • Collaborative UUVs • Collaborative USVs • Collaborative self-driving cars • Collaborative heterogenous unmanned vehicles	Multiple fog nodes installed on some of the mobile robots/structures and/or in the fixed communication infrastructure base stations. Limitations may apply on fog nodes on the mobile robots/structures depending on their capabilities.
Wide Area Multi-Robot Systems (WMRS)	• Widely distributed collaborative manufacturing • City-wide sky defense systems • Cloud/Internet multiple distributed robots	Multiple fog nodes organized in a hierarchical structure for large-scale systems should be used. Access to a wide selection of equipment and computing facilities allows for sophisticated capabilities on the fog nodes used.

Table 2. Differences in support provided by cloud and fog computing for MRS applications.

Aspect	Cloud Computing	Fog Computing
Communication latency	Comparatively high and unstable	Low and stable
Computation power	High and scalable	Low and limited
Data storage capacity	Large and scalable.	Small to medium
Appropriate services	• High computational analysis • Long term storage • Maintaining knowledge-based systems • Planning • Optimization • Large-scale scheduling • Learning • Large-scale resource management • Unknown fault detection • Large-scale video processing	• Computation offloading • Temporary storage • Communication • Caching • Security • Monitoring and control • Navigation • Mobility support • Data streaming • Synchronization • Integration with other systems • Small-scale local resource management Rescheduling

VIII. CONCLUSION

Fog computing emerged as a way to augment the functionality and services of cloud computing by offering more localized services closer to the operational components of the applications using them. Fog nodes are deployed at the network edges, placing them at close proximity to the actual system, thus improving the support for low-latency requirements, localization and mobility. For MRS applications, such support is usually important as these applications rely heavily on localized interactions and collaboration. However, the varying types of MRSs and the different types of their applications impose some challenges to fog utilization. In this paper, we attempt to address this issue by first identifying the main services fog computing can offer MRS applications. These include processing, storage, communication, security, monitoring and control, navigation, scalability and synchronization services. The next step is categorizing the MRSs into five categories based on the type of network used. This categorization leads to a better view of the different MRS applications and their needs. With this knowledge, it becomes less complicated when trying to design and develop a suitable fog computing architecture to serve each category. We also proposed a service-oriented platform for fog-enabled MRSs. In this platform, all robots in an MRS and fog nodes resources are viewed as a set of services to be used to develop the MRS applications. Hence access and utilization of these services becomes easier and can support integrating services from different systems as well.

REFERENCES

[1] N. Mohamed, J. Al-Jaroodi, I. Jawhar, and S. Lazarova-Molnar, "Middleware Requirements for Collaborative Unmanned Aerial Vehicles," in proc. Int'l Conference on Unmanned Aircraft Systems (ICUAS'13), IEEE, pp. 1040-1049, 2013.

[2] Y. Chen, Z. Du, and M. García-Acosta, "Robot as a service in cloud computing," In Fifth IEEE International Symposium on Service Oriented System Engineering (SOSE), pp. 151-158, IEEE, 2010.

[3] S. Mahmoud and N. Mohamed, "Collaborative UAVs Cloud," In 2014 International Conference on Unmanned Aircraft Systems (ICUAS), pp. 365-373, IEEE, 2014.

[4] B. Kehoe, S. Patil, P. Abbeel, and K. Goldberg, "A survey of research on cloud robotics and automation," IEEE Transactions on automation science and engineering, 12(2), pp.398-409, 2015.

[5] 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, pp. 13-16, ACM, 2012.

[6] L.E. Parker, D. Rus, and G.S. Sukhatme, "Multiple Mobile Robot Systems," In Springer Handbook of Robotics, pp. 1335-1384, Springer International Publishing, 2016.

[7] R.P. Murphy, "Marsupial and shape-shifting robots for urban search and rescue," IEEE Intelligent Systems and their applications, 15(2), pp.14-19, 2000.

[8] S. Waharte, N. Trigoni, and S. Julier, "Coordinated search with a swarm of UAVs," In 6th Annual IEEE Communications Society Conference on Sensor,

Mesh and Ad Hoc Communications and Networks Workshops, pp. 1-3, IEEE, June 2009.

[9] 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, 23(3-4), pp.165-184, 2006.

[10] I.A. Wagner, Y. Altshuler, V. Yanovski, and A.M. Bruckstein, "Cooperative cleaners: A study in ant robotics," The International Journal of Robotics Research, 27(1), pp.127-151, 2008.

[11] N. Noguchi, J. Will, J. Reid, and Q. Zhang, "Development of a master–slave robot system for farm operations," Computers and Electronics in agriculture, 44(1), pp.1-19, 2004.

[12] C. Yinka-Banjo, A. Bagula, and I.O Osunmakinde, "Autonomous Multi-robot Behaviours for Safety Inspection under the Constraints of Underground Mine Terrains," Ubiquitous Computing and Communication Journal, 7(5), p.1316, 2012.

[13] A. Stroupe, T. Huntsberger, A. Okon, H. Aghazarian, and M. Robinson, "Behavior-based multi-robot collaboration for autonomous construction tasks," In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005), pp. 1495-1500, IEEE, 2005.

[14] I. Maza, F. Caballero, J. Capitán, J.R. Martínez-de-Dios, and A. Ollero, "Experimental results in multi-UAV coordination for disaster management and civil security applications," Journal of intelligent & robotic systems, 61(1), pp.563-585, 2011.

[15] Y. Guo, L.E. Parker, and R. Madhavan, "Towards collaborative robots for infrastructure security applications," In Proceedings of The 2004 International Symposium on Collaborative Technologies and Systems, pp. 235-240, 2004.

[16] N. Correll, K.E. Bekris, D. Berenson, O. Brock, A. Causo, K. Hauser, K. Okada, A. Rodriguez, J.M. Romano, and P.R. Wurman, "Analysis and observations from the first amazon picking challenge," IEEE Transactions on Automation Science and Engineering, 2016.

[17] R. Alami, S. Fleury, M. Herrb, F. Ingrand, and F. Robert, "Multi-robot cooperation in the MARTHA project," IEEE Robotics & Automation Magazine, 5(1), pp.36-47, 1998.

[18] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, and E. Osawa., "Robocup: The robot world cup initiative," In Proceedings of the first international conference on Autonomous agents, pp. 340-347, ACM, 1997.

[19] B.P. Gerkey and M.J. Matarić, "A formal analysis and taxonomy of task allocation in multi-robot systems," The International Journal of Robotics Research, 23(9), pp.939-954, 2004.

[20] L.E. Parker, "Distributed intelligence: Overview of the field and its application in multi-robot systems," Journal of Physical Agents, 2(1), pp.5-14, 2008.

[21] I. Stojmenovic and S. Wen, "The fog computing paradigm: Scenarios and security issues," In 2014 Federated Conference on Computer Science and Information Systems (FedCSIS), IEEE, 2014.

[22] S. Yi, C. Li, and Q. Li, "A survey of fog computing: concepts, applications and issues," In proc. of the 2015 Workshop on Mobile Big Data, pp. 37-42, ACM, 2015.

[23] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for vm-based cloudlets in mobile computing," Pervasive Computing, 2009.

[24] M. Satyanarayanan, Z. Chen, K. Ha, W. Hu, W. Richter, and P. Pillai, "Cloudlets: at the leading edge of mobile-cloud convergence," in IEEE International Conference on Mobile Computing, Applications and Services (MobiCASE), 2014.

[25] K. Hong, D. Lillethun, U. Ramachandran, B. Ottenwälder, and B. Koldehofe, "Mobile fog: A programming model for large-scale applications on the internet of things," In proc. of the 2nd ACM SIGCOMM Workshop on Mobile Cloud Computing, pp. 15-20), ACM, 2013.

[26] D. F. Willis, A. Dasgupta, and S. Banerjee, "Paradrop: a multi-tenant platform for dynamically installed third party services on home gateways," in ACM SIGCOMM workshop on Distributed cloud computing, 2014.

[27] K. Ha, Z. Chen, W. Hu, W. Richter, P. Pillai, and M. Satyanarayanan, "Towards wearable cognitive assistance," in Mobisys. ACM, 2014.

[28] J. Zhu, D.S. Chan, M.S. Prabhu, P. Natarajan, H. Hu, and F. Bonomi, "Improving web sites performance using edge servers in fog computing architecture," In IEEE 7th International Symposium on Service Oriented System Engineering (SOSE), pp. 320-323, IEEE, 2013.

[29] Y. Cao, P. Hou, D. Brown, J. Wang, and S. Chen, "Distributed analytics and edge intelligence: Pervasive health monitoring at the era of fog computing," in proc. of the Workshop on Mobile Big Data. ACM, 2015.

[30] M. A. Hassan, M. Xiao, Q. Wei, and S. Chen, "Help your mobile applications with fog computing," in Fog Networking for 5G and IoT Workshop, 2015.

[31] I. Stojmenovic, S. Wen, X. Huang, and H. Luan, H., "An overview of Fog computing and its security issues," Concurrency and Computation: Practice and Experience, 2015.

[32] C. Dsouza, G.J. Ahn, and M. Taguinod, M., "Policy-driven security management for fog computing: Preliminary framework and a case study," In IEEE 15th International Conference on Information Reuse and Integration (IRI), pp. 16-23), IEEE, 2014.

[33] E. Tunstel, M.A. De Oliveira, and S. Berman, "Fuzzy behavior hierarchies for multi-robot control," International Journal of Intelligent Systems, 17(5), pp.449-470, 2002.

[34] B. Tang, Z. Chen, G. Hefferman, T. Wei, H. He, and Q. Yang, "A hierarchical distributed fog computing architecture for big data analysis in smart cities. In Proceedings of the ASE BigData & Social Informatics, ACM, 2015.

[35] E. Gul, B. Atakan, and O.B. Akan, "NanoNS: A nanoscale network simulator framework for molecular communications," Nano Communication Networks, 1(2), pp.138-156, 2010.

[36] A.O. Tarakanov, L.B. Goncharova, and Y.A. Tarakanov, "Carbon nanotubes towards medicinal biochips," Wiley Interdisciplinary Reviews: Nanomedicine and Nanobiotechnology, 2(1), pp.1-10, 2010.

[37] G.M. Patel, G.C. Patel, GR.B. Patel, J.K. Patel, and M. Patel, "Nanorobot: a versatile tool in nanomedicine," Journal of Drug Targeting, 14(2), pp.63-67, 2006.

[38] R.A. Freitas, "Pharmacytes: An ideal vehicle for targeted drug delivery," Journal of Nanoscience and Nanotechnology, 6(9-10), pp.2769-2775, 2006.

[39] A. Cavalcanti, B. Shirinzadeh, and L.C. Kretly, "Medical nanorobotics for diabetes control. Nanomedicine: Nanotechnology, Biology and Medicine," 4(2), pp.127-138, 2008.

[40] I. Jawhar, N. Mohamed, and L. Zhang, "Inter-Vehicular Communication Systems, Protocols and Middleware," in proc. 5th IEEE Int'l Conference on Networking, Architecture, and Storage (NAS 2010), IEEE Computer Society Press, pp. 282-287, July 15-17, 2010.

[41] Open Connectivity Foundation Website, https://openconnectivity.org/, viewed March 24, 2017.

[42] Z. Chen, F. Xia, T. Huang, F. Bu, and H. Wang, "A localization method for the Internet of Things. The Journal of Supercomputing, pp.1-18, 2013.

[43] J. Al-Jaroodi, N. Mohamed, H. Jiang, and D. Swanson, "Middleware Infrastructure for Parallel and Distributed Programming Models on Heterogeneous Systems," in IEEE Transactions on Parallel and Distributed Systems, Special Issue on Middleware Infrastructures, Volume 14, No. 11, pp. 1100-1111, Nov. 2003.

[44] J. Al-Jaroodi, N. Mohamed, H. Jiang, and D. Swanson, "An Agent-Based Infrastructure for Parallel Java on Heterogeneous Clusters," in proc. 4th IEEE Int'l Conference on Cluster Computing (CLUSTER 2002), IEEE, pp. 19-27, Sep. 2002.

[45] Arduino website, https://www. arduino .cc/, viewed Dec. 20, 2017.

[46] DHT Sensor Library Website, https://github.com/adafruit/DHT-sensor-library, viewed December 20, 2017.

[47] CC3000 Wi-Fi board Website, https://www. adafruit.com/products/1469, viewed December 20, 2017.

[48] Arduino IDE Website, http://arduino.cc/en/main/software, viewed December 20, 2017.

[49] Adafruit CC3000 library Website, https://github.com/adafruit/Adafruit_CC3000_Library, viewed December 24, 2017.