IoT

Article

A Pervasive Collaborative Architectural Model at the Network's Periphery

Ghassan Fadlallah, Hamid Mcheick * and Djamal Rebaine

Department of Computer Science and Mathematics, University of Quebec in Chicoutimi, Saguenay, QC G7H 2B1, Canada; Ghassan-Mustapha.Fadlallah1@uqac.ca (G.F.); Djamal_Rebaine@uqac.ca (D.R.)
* Correspondence: Hamid_Mcheick@uqac.ca

Abstract: Pervasive collaborative computing within the Internet of Things (IoT) has progressed rapidly over the last decade. Nevertheless, emerging architectural models and their applications still suffer from limited capacity in areas like power, efficient computing, memory, connectivity, latency and bandwidth. Technological development is still in progress in the fields of hardware, software and wireless communications. Their communication is usually done via the Internet and wireless via base stations. However, these models are sometimes subject to connectivity failures and limited coverage. The models that incorporate devices with peer-to-peer (P2P) communication technologies are of great importance, especially in harsh environments. Nevertheless, their power-limited devices are randomly distributed on the periphery where their availability can be limited and arbitrary. Despite these limitations, their capabilities and efficiency are constantly increasing. Accelerating development in these areas can be achieved by improving architectures and technologies of pervasive collaborative computing, which refers to the collaboration of mobile and embedded computing devices. To enhance mobile collaborative computing, especially in the models acting at the network's periphery, we are interested in modernizing and strengthening connectivity using wireless technologies and P2P communication. Therefore, the main goal of this paper is to enhance and maintain connectivity and improve the performance of these pervasive systems while performing the required and expected services in a challenging environment. This is especially important in catastrophic situations and harsh environments, where connectivity is used to facilitate and enhance rescue operations. Thus, we have established a resilient mobile collaborative architectural model comprising a peripheral autonomous network of pervasive devices that considers the constraints of these resources. By maintaining the connectivity of its devices, this model can operate independently of wireless base stations by taking advantage of emerging P2P connection technologies such as Wi-Fi Direct and those enabled by LoPy4 from Pycom such as LoRa, BLE, Sigfox, Wi-Fi, Radio Wi-Fi and Bluetooth. Likewise, we have designed four algorithms to construct a group of devices, calculate their scores, select a group manager, and exchange inter-group and intra-group messages. The experimental study we conducted shows that this model continues to perform efficiently, even in circumstances like the breakdown of wireless connectivity due to an extreme event or congestion from connecting a huge number of devices.

Keywords: pervasive architectural models; collaborative computing in IoT; connectivity of IoT objects; harsh environments; construct group of nodes; resilient and rescue systems; pervasive computing

Citation: Fadlallah, G.; Mcheick, H.; Rebaine, D. A Pervasive Collaborative Architectural Model at the Network's Periphery. *IoT* **2021**, *2*, 524–548. https://doi.org/10.3390/iot2030027

Academic Editors: Benoît Parrein and Bastien Confais

Received: 29 July 2021
Accepted: 1 September 2021
Published: 6 September 2021

1. Introduction

Continuous and rapid technological innovation, especially in the areas of telecommunications and information technology, has led to a radical expansion of the Internet throughout the world. This has given rise to the Internet of Things (IoT). Moreover, portable smart devices that use the Internet and various wireless technologies have become popular. The facilities in cities, including buildings, offices, factories, markets, home appliances, machines, security systems and cars are connected to each other. This is what Ashton [1] first called the IoT in 1999. IoT supports a wide variety of networks interconnected through a variety of devices, including servers, databases, computers, laptop and wearable devices like smartphones and personal digital assistants.

The improvement of the Internet has accelerated the development of practical solutions to the various problems of distributed systems, such as heterogeneity, openness, connectivity, security, scalability, failures, transparency and quality of service [2]. Advances in technology like device miniaturization and both P2P and wireless communication have enabled the integration of pervasive devices into distributed systems. These enhance and expand collaborative mobile computing by using small low-cost devices (wearable/portable) present in the physical environments of users, including homes, offices, and even natural environments.

This model of computing allows the interaction of remote and mobile users to perform different activities using a variety of pervasive devices. This interaction has become an effective technique for dealing with complex problems by breaking them down into smaller sub-problems processed adequately on large networks. The pervasive devices develop very quickly and their functions are enriched and enhanced while their capabilities of communication are increased. In this paradigm, all users, hardware and software are dynamic, and they change these capabilities in unpredictable ways [2,3].

The development of collaborative systems has led to innovative designs for mobile networks and their devices' technologies. In addition, they enhance mobile collaborative computing and the techniques of interaction between these devices.

We distinguish many architectures that have been established recently in pervasive collaborative computing on distributed mobile networks, such as cloud, fog and edge computing [4].

Cloud computing, the main architecture supporting the IoT, has successfully helped to solve the problems of collaborative computing in areas such as communication, computing capabilities and storage and transfer of data. Similarly, fog, edge and mobile edge computing (MEC) [5] solved the problems faced by cloud computing when offloading to central servers and vice versa. Fog and edge computing architectures [6] (and recently MEC), with several other architectural models, have gained primordial importance in mobile networks.

However, the effectiveness of devices in these structures became disproportionate to the intense demand for communications. This effectiveness is needed even more in cases like natural disasters, industrial accidents and terrorist attacks, which can isolate or destroy their communication service centers, base stations and access points. Therefore, the solution for situations of disconnected communication is most likely realized using networks that comprise mobile devices and have ubiquitous resources that can be connected through their capabilities of P2P connection. These devices can be regrouped instantly in networks in real time and in a variety of positions. Besides extreme events, these networks could be used in normal situations due to their very low cost and their permanent presence in our pockets. These networks provide users with facilities and benefits to manage their work anytime and anywhere. For these reasons, and with the emergence of new technologies expected in the future, we claim that these devices and networks will be the de facto tools to support the paradigm of smart mobile collaborative computing.

The pace of technological development constantly imposes real challenges in a variety of areas, particularly in mobile networks. This is due to the rapid advances in the architecture and techniques of IT models, mobile devices and communication tools, such as wireless and P2P, through self-diffusion. P2P communication allows access to devices without the need for wireless infrastructure such as access points or evolved base stations (eNBs). P2P can use various short-range wireless technologies [7] like Bluetooth, Wi-Fi Direct, LTE Direct (defined by the Third Generation Partnership Project (3GPP) [8]), near field communication (NFC) and proximity services as needed. The mobile networks can be composed of devices that connect spontaneously using these types of mobile communication, and this offers performance services in a variety of areas [9]. In addition, emerging Radio Wi-Fi technology enabled by Pycom's LoPy4 can extend the range of P2P and wireless communication. As a result, networking has become a resource of pervasive devices that can be connected anytime and anywhere. With key technologies such as spatial modulation, millimeter wave, visible light communication and 5G, cellular networks will soon enable people to use P2P technology globally. The objective is to increase the number of connected devices with longer battery life and greatly reduced latency [10].

The IoT paradigm is defined as a dynamic global self-configuration and interoperable network [11,12]. It fascinates and even seduces its users. Because dynamic situations give these users more freedom in terms of place of work and time to continue their different jobs, the demand for the use of these devices or objects is steadily increasing. Likewise, the development of pervasive devices is leading to an intensification of their use in increasingly complex collaborative applications, as well as to an intensive use of resources. [2,13,14]. Unfortunately, technological progress does not follow this demand proportionately. For example, the number of extant mobile devices, including phones and tablets, has increased from more than 7.7 billion in 2014 to more than 12.1 billion in 2018 [15]. Moreover, some expect that, by 2025, there will be up to 74 billion connected IoT devices [16] or perhaps as many as 100 billion [17]. Therefore, since the technological advancement of these devices lags behind the increasing demand for their use, it can be compensated by improving the performance of the applications that these devices will achieve in terms of the speed of response and the completion of tasks. This can be done using job scheduling and load balancing techniques and by maintaining connectivity in networks, especially in harsh environments, to ensure the continuity of activities for the required time.

Therefore, this paper seizes the opportunity of emerging architectures such as edge, fog, cloud, mobile ad hoc cloud computing and multi-group networking to build an autonomous architecture of pervasive, collaborative computing in the IoT paradigm. The main idea of the approach is to establish a peripheral autonomous network of mobile devices using their own P2P radio communication and the radio wireless enabled by Pycom's LoPy4. Therefore, the present work aims to:

1. Reduce the risk of total or partial interruption of network connectivity caused by breakdowns or overwhelming (a huge number of) connections. This involves the improvement and continuity of maintaining the connectivity between mobile devices, regardless of the state of the network.
2. Increase the resilience and reliability of sharing data, information and computations between these devices (objects).

To achieve these goals, we have designed a pervasive mobile network based on a collaborative layered architectural model [18] that allows users to communicate in various situations. For this purpose, we have taken into account the capacity limitations of their devices in terms of data storage, computing, and expected battery life. This model has been designed to enhance computing capacity, improve response time and increase the use of ubiquitous surrounding resources. The development of this architectural model includes the following steps:

(a) Design an architecture sample to maintain connectivity, even during extreme events and harsh environments.
(b) Establish a multi-group network connected by device-to-device communication, such as Wi-Fi Direct and other communication technologies that have a larger range connection, such as those of Pycom's LoPy4.
(c) Propose four new algorithms and validate them through the required applications needed to manage the proposed model, such as dividing tasks and balancing their load. Among these applications is one to program the Lopys as transmitter, bridge and receiver.

Pervasive computing is the natural successor of mobile computing systems. It is an emerging field of research that offers revolutionary models for the most recent computing paradigms. Pervasive computing has led to tremendous advances in technologies such as computing and mobile devices, as well as technologies of communication, wireless networking, middleware programing, smart spaces and distributed and embedded systems.

The present study fosters collaboration between groups of objects through intra-group and inter-group (of devices) communication in the IoT model, where most of these objects are no longer managed by a system administrator. Note that collaborative computing can be defined as "a fertile blend of technologies and techniques which facilitate people working together via computer-assisted means" [19]. Pervasive computing is an emerging field of distributed systems, where objects are small, mobile and connected by wireless and radio P2P technologies. It can be divided into three sub-classes: systems of ubiquitous computing, mobile systems and sensor networks [20]. Our research combines collaborative and pervasive computing to (i) embrace a pervasive collaborative model for IoT and (ii) facilitate and assist cooperation between people, especially in harsh environments, to exploit the capabilities of each object (device). Indeed, we think the interesting applications come from the collaboration of objects with minimal human collective intervention rather than from individual ones.

Other technologies are used in this paper. Let us introduce them briefly. LoRa is a wireless technology that enables communications over long distances, but at a low data rate. It uses sensors and actuators for P2P and IoT applications. It can use the radio spectrum to enable low power broadband communication between remote devices and network gateways. LoRa is the de facto platform for low power wide area network (LPWAN) technology for IoT. The LoRaWAN® specification, designed by Pycom (Guildford, UK), is a networking protocol designed to enable rapid establishment of public or private IoT networks anywhere using bidirectional communication, mobility and localization services with end-to-end security [21]. Wi-Fi-direct is a Wi-Fi ad hoc standard for peer-to-peer wireless connections that allows two devices to communicate without an access point, router, or Internet connection.

The remainder of this paper is organized as follows. Section 2 describes related work, and discusses the problem of pervasive architectural models at the periphery. Section 3 describes the proposed pervasive collaborative architectural model. Section 4 presents the experimental study we conducted. Finally, concluding remarks are presented in Section 5.

2. Related Works on Pervasive Architectural Models and Their Limits

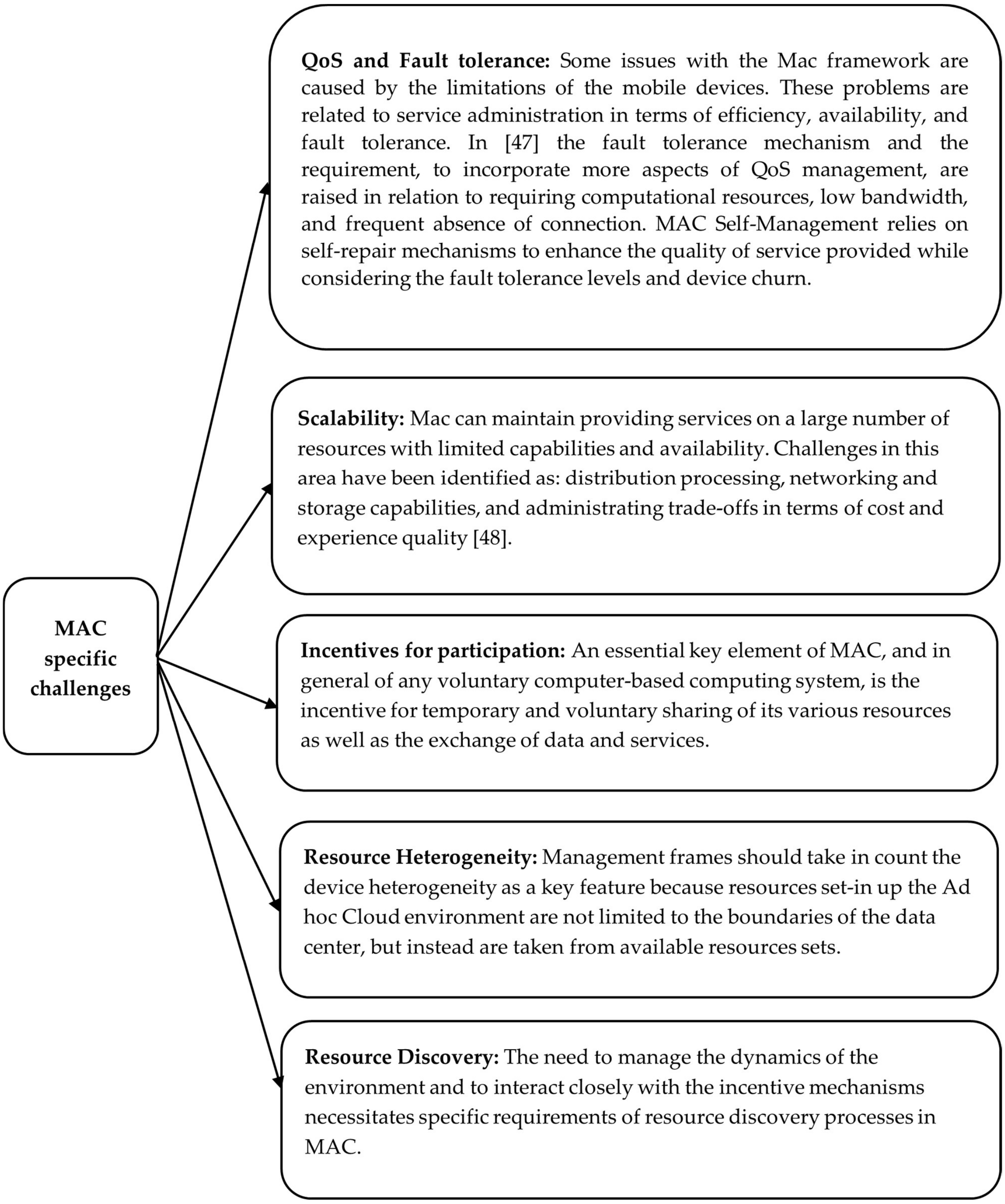
This section surveys pervasive architectural models and identifies their limits.

2.1. Pervasive Architectural Models

Maintaining connectivity between devices in pervasive computing, and particularly in the IoT paradigm, is important for serving people in various situations, especially in rural and harsh environments. For example, connectivity must be maintained with a smart car while it is driving as well as during a live and continuous transfer of data if it is necessary to make quick decisions. In addition, maximizing coverage areas, especially in these environments, is needed to ensure communications between pairs of devices over at least one connection. Many approaches and algorithms are designed to increase connectivity and maximize coverage in networks. Most require a central station [22,23] with advanced resources to control the devices in the network, or they require homogenous devices [24,25]. Recently, Mcheick et al. [26] designed an algorithm CMHWN to provide connectivity between at least two devices and to add flexibility in controlling the overall displacement in the network. Ghaddar et al. [27] designed and validated a connectivity approach to minimize the movement of devices to save energy and support the self-organization of each object (device). Wang et al. proposed a communication system using only Wi-Fi Direct technology [28]. However, these algorithms and systems do not handle heterogeneous communication technologies and require a central station to maintain connectivity. Therefore, many existing technologies can be considered in contemporary architectures to maximize coverage and maintain connectivity in pervasive computing.

In addition, major research efforts have been conducted in the fields of cloud, fog, edge and mobile edge computing and their branching at the peripheral level. The fundamental structure in this area is cloud computing. The main problem of the cloud is that its services are centralized, which sometimes slows down processes and can cause total or partial shutdowns due to a huge number of connections. This problem is the basis for other approaches, which transform these centralized services into decentralized ones through intermediate and peripheral service centers. In this way, several architectural models were developed: fog, edge and mobile edge computing, cloudlet, and collaborative fog computing [29,30].

Table 1 reviews the most recent and widely used of these architectural models.



QoS and Fault tolerance: Some issues with the Mac framework are caused by the limitations of the mobile devices. These problems are related to service administration in terms of efficiency, availability, and fault tolerance. In [47] the fault tolerance mechanism and the requirement, to incorporate more aspects of QoS management, are raised in relation to requiring computational resources, low bandwidth, and frequent absence of connection. MAC Self-Management relies on self-repair mechanisms to enhance the quality of service provided while considering the fault tolerance levels and device churn.

Scalability: Mac can maintain providing services on a large number of resources with limited capabilities and availability. Challenges in this area have been identified as: distribution processing, networking and storage capabilities, and administrating trade-offs in terms of cost and experience quality [48].

Incentives for participation: An essential key element of MAC, and in general of any voluntary computer-based computing system, is the incentive for temporary and voluntary sharing of its various resources as well as the exchange of data and services.

Resource Heterogeneity: Management frames should take in count the device heterogeneity as a key feature because resources set-in up the Ad hoc Cloud environment are not limited to the boundaries of the data center, but instead are taken from available resources sets.

Resource Discovery: The need to manage the dynamics of the environment and to interact closely with the incentive mechanisms necessitates specific requirements of resource discovery processes in MAC.

Figure 1. MAC specific challenges [47,48].

Table 1. Classification of contemporary architectural models.

Architectural Model	Description and Analysis of These Models
Mobile edge computing (MEC)	The main advantage of MEC is to deploy mini cloud servers in each communication base station, especially at the edge of mobile networks. MEC is applied to mobile devices in these edge networks to reduce delays by allowing computational and storage capabilities at the edge of mobile networks instead of being at central servers [5,31]. MEC has better discharge techniques to edge servers, giving the network low latency and high bandwidth [6]. In addition, it provides efficient storage and energy consumption through many architectures such as the software-defined system for mobile edge computing (SDMEC) for storage [32] and ME-VoLTE for computing, which integrates MEC with long-term evolution (LTE) [33].
Cloudlet	A cloudlet is a proximal cloud that brings computing capacities closer to the mobile devices it serves to avoid latency. A cloudlet is formed by a cluster of multi-core computers, a very powerful multi-core server and a high bandwidth wireless LAN with high connectivity [34,35]. "Cloudlets are decentralized and widely dispersed Internet infrastructure whose compute cycles and storage resources can be leveraged by nearby mobile computers [34]."
Collaborative fog computing (CoFog)	CoFog provides dynamic services based on predefined templates using formal mechanisms called operations. An operation represents a relationship between a collaboration request and the services that can fulfill that request. There are two types of operations: conservative, which retains the same type of data, and non-conservative, which creates a new type. CoFog relies on the ability to discover, recover and provide security services using role-based and attribute-access models [19].
Mobile cellular cloud architecture (MoCCA)	MoCCA [29,36] uses small scale micro-servers loaded on mobile phones to build its own mobile cloud architecture. Its resources are smartphones, base stations, base station controllers and mobile switching centers. Its main concerns are (i) connectivity, (ii) computational limitations, (iii) churn, because of user mobility and device volatility, (iv) lack of power; and (v) user incentives.
mClouds	With mClouds [29,37], mobile devices become essential components of cloud computing architectures. An mCloud uses specific resource-discovery procedures, incentive management and distributed processing to execute locally whenever possible.
MobiCloud	MobiCloud [29,38] is a trustworthy collaborative mobile cloud administration system that enables the efficient and collaborative management of mobile smart phone resources. Its objective is to use mobile devices even when there is no internet connectivity. MobiCloud architecture has participant nodes and a field control node (a cloud agent) that provides centralized cloud controller functionalities by locating the required resources.
Mobile cloudlet (M.Cloudlet)	M.Cloudlet uses its own servers without base station services to provide cloud services in order to dispense the connection to remote cloud central servers or central ones of fixed cloudlets. To build a network of mobile cloudlet, M.Cloudlet uses Wi-Fi Direct to connect nearby mobile devices [39]. The M.Cloudlet architecture has two main phases: (i) build M.Cloudlet by discovering and negotiating neighboring devices to form their groups and choose their owners, then exchange information, (ii) control and manage devices, including their tasks and spontaneous lists of participating devices [39].
Multi-group networking	This supports Wi-Fi Direct connections between multiple devices in groups, then interconnects different groups to create ad hoc multi-hop networks. However, the Wi-Fi Direct standard [40] defines only intra-group communications, where the group owner (GO) is central to all communications in its group. A Wi-Fi Direct device called a gateway node can operate as a member of more than one group at a time. In a proposed time-sharing mechanism, the gateway node can switch between two or more groups to implement multi-group communication scenarios using only Wi-Fi Direct [41].
Building smart multi-group networks by Wi-Fi Direct	Independent group formation aims to achieve the efficient use of devices while maintaining connectivity. The main obstacle to the widespread adoption of P2P communication is the lack of coordination of services. Efforts are focused on the potential of Wi-Fi Direct as a P2P communication technology in medium- and large-level frameworks using unrooted Android devices. It functions by (i) designing multi-groups interconnected logical topology by tunnelling the application layer and (ii) proposing a fully distributed intelligent group formation mechanism to construct networks, considering the group's durability, the devices' power consumption, the bandwidth average and network extent [42].
Mobile ad hoc cloud computing (MAC)	MAC computing "is still in its infancy" [29]. Several definitions of MAC have been proposed [43–45], such as "MAC enables the use of many nearby resource-rich mobile devices to provide computational services in the vicinity" [45]. MAC roots are in mobile cloud computing (MCC) but also in opportunistic computing [43] or distributed computing with the caveats of intermittent connectivity and delay tolerance. The ad hoc cloud features are developed by Kirby et al. [46] as self-administration in terms of flexibility, performance and balancing potentially contradictory policy goals. MAC challenges include (i) limited computing and energy resources of mobile devices and their ability to perceive context and location, (ii) network connectivity, (iii) security and (iv) partitioning and application loading. In addition, there are specific challenges, as shown in Figure 1, introduced in the MAC context as mobile devices are the only resources [29].

2.2. *Issues of Pervasive Architectural Models at the Periphery of a Network*

The main problem of our research occurs from the emerging architecture field of distributed systems, especially pervasive collaborative computing at the periphery of a network. These architectures can be classified into three categories: (i) main centralized as cloud, (ii) peripherally centralized including fog, edge computing and mobile edge computing, and (iii) other peripheral architectures such as emerging technologies introduced in recent years, especially mobile ad hoc networks at the periphery (MANET, VANET, multi-groups networks, mobile cloudlet, etc.) [49]. The peripheral architectures introduced most recently make smart mobile systems more efficient in terms of communication, power consumption, data storage and computation capability while they reduce peripheral device discharge to cloud service centers. Note that cloud computing was created as a solution for mega data exchange and distributed collaborative and ubiquitous computing. Fog and edge computing were also designed as system architectures to address the problems of cloud computing near peripheral devices. Still other architectures have been designed as solutions for more distant peripheral devices including cloudlet, mobile cloud computing, mobile cloudlet, mobile ad hoc cloud computing, smart P2P and multi-group networks.

These peripheral architectures provide access to a large number of devices. The main issue is to maintain and enhance connectivity using the offered technologies, which include (i) P2P technology such as WI-FI-Direct, and LTE-Direct and (ii) wireless technology using base stations such as Wi-Fi and Lora.

The emergence of the Internet of Things (IoT) and its widespread deployment as a new model that processes a huge amount of data has led to problems with collaborative structures. These include many applications, such as management, communicating and exchanging data between remote users at the periphery of networks away from their hubs or centers.

Centralized cloud computing and other computing architectures at the levels closest to the periphery were, in their turn, effective solutions to these problems because of the proliferation of data centers and smart micro-devices that reach the network edges. However, the efficiency of these architectures has declined in proportion to the increase in requests from a huge number of users and the colossal amount of data to be processed. Therefore, it has caused a serious bottleneck in terms of connectivity and communication, especially for device users at the periphery. The networks in these paradigms generally comprise small mobile devices that suffer from the problem of limited capacity, as mentioned earlier.

Following a thorough study and analysis of these architectural models, we highlight the following facts about this area: (i) there is a strong trend towards the development and evolution of small, smart mobile devices; (ii) there is excessive demand to use them, and (iii) researchers have made diligent efforts to improve their performance and functionality and overcome their disadvantages. The main research questions to address on the limits of these structures are:

- How to maintain the connectivity of network devices in various situations?
- How to reduce the tasks of loading and unloading these small peripheral devices with central or intermediate service centers (cloud, fog, edge) to avoid the communication bottleneck?
- How to reduce the energy consumption of these devices by minimizing their time of use and improving task scheduling and load balancing algorithms?

We identified five emerging issues in this area of intelligent and collaborative mobile computing at the periphery of networks:

(1) Limited connection bandwidth because of congestion from loading and unloading messages (tasks and data) to fixed or mobile center servers [29].
(2) The limitations of mobile devices in terms of power, computing ability and data storage [39].
(3) The reduced range of mobile network devices in terms of device-to-device connection when wireless connectivity with base stations is lost.
(4) The complete loss of operability or effectiveness of these devices during severe events such as natural disasters and terrorist attacks because of malfunctions or failures in the base network stations.
(5) The gap in performing relatively large jobs using these devices. This is partly due to the inadequate partitioning and scheduling tasks relative to the capabilities and loads of the equipment.

3. Collaborative Architectural Model of Mobile Pervasive Computing

Pervasive computing aims to improve the human experience and quality of life without a clear awareness of the underlying communications and computing technologies [50]. This section is about the architectural model we proposed, the description of the structure of the connectivity and the algorithms to manage the construction and communication of the device groups. We also address briefly the security related to the access of the network.

3.1. Specific Objectives

With the large number of mobile device technologies that contribute significantly to most sectors, there are still unmet needs to ensure the availability, permanence and continuity of wireless connectivity in several environments. The standard wireless connection technologies that use base stations and access points can be interrupted because of failures for a variety of reasons. These include a lack of power, natural disasters and terrorist attacks. Moreover, communication might not be available in certain overcrowded areas (for example, a concert or conference hall), or even to manage smart car communications. As a result, the trend is to address this deficiency by taking advantage of the emergence of device-to-device (P2P) communication technologies such as Bluetooth, Wi-Fi Direct and LTE-Direct. Based on these technologies, an autonomous local communication system can be developed to provide an instant communication environment for mobile users so they can communicate with each other [28]. Consequently, this type of radio communication technology has become the main concern in the recent development of wireless connections, particularly 5G [51].

The objective of the present study is to design an autonomous pervasive collaborative architectural model (APCAM) consisting of a network of ubiquitous smart mobile devices. In this model, data and information are exchanged between devices using device-to-device radio communication (such as Wi-Fi Direct) along with various available wireless communication technologies. This approach helps to strengthen the capabilities and then the performance of mobile peripheral networks by developing and improving their structure at both the hardware and software levels. On the one hand, this model relies on recent work in the field of collaborative mobile computing [36,41,42,52] and on the latest hardware communication technologies such as device-to-device radio communication and Pycom's LoPy4 that can be used as private wireless base stations. On the other hand, it relies on the latest load balancing and scheduling techniques, which must be adapted to its context, especially at the periphery of networks.

The specific objectives of our model are:

(1) Maintain the device's connectivity in harsh environments and at any time.
(2) Make objects (devices) and their networks more autonomous.
(3) Compensate for loading/unloading to/from the service centers of cloud and other edge (middle and peripheral) architectures.
(4) Maintain the capacity of the devices such as energy, computing ability and storage. Therefore, for this paper, we considered these aspects in these two areas:
    (a) Smart mobile network peripheral architectures in terms of technologies, protocols, and applications of networking. There is a variety of devices such as sensors, smartphones, and tablets, in addition to small laptops, that have a capacity to be used as local micro-servers. These devices can be grouped according to their ranges. In this way, a chain of groups is established in several zones, taking advantage of the available connection modes such as wireless, device-to-device and those offered by Pycom's LoPy4 technologies.
    (b) Software in the form of algorithms and applications to carry out tasks such as the creation of groups, the election of a group manager, and exchanging data and messages between nodes, as well as partitioning, scheduling and load balancing of tasks. These algorithms and applications are tested and validated by simulating harsh situations, for example, by sending messages from one device to another far away by combining the P2P connection technologies

of devices and by establishing a proper mode of wireless technology based on communication technologies offered by Pycom's LoPy4.

3.2. Autonomous Pervasive Collaborative Architectural Model (APCAM)

We built this model based on the aforementioned technologies for connecting smart mobile devices. It consists of reassembling these devices in groups connected through the communication technologies available according to their ranges in their zones. Among the most important of these technologies is the Wi-Fi Direct, which uses the gateway node as a means of communicating between these groups (please see details in Figure 2 of the reference [18]). These devices can operate independently of base stations through their own wireless radio technology by integrating with the access points provided by Pycom's LoPy4. Thus, this model can play a crucial role in the case of losing communication system infrastructure as base stations. This is especially important in harsh situations, when an emergency event occurs, where for example smart car communications and natural disasters must be managed. In such cases, the primary concern is maintaining connectivity.

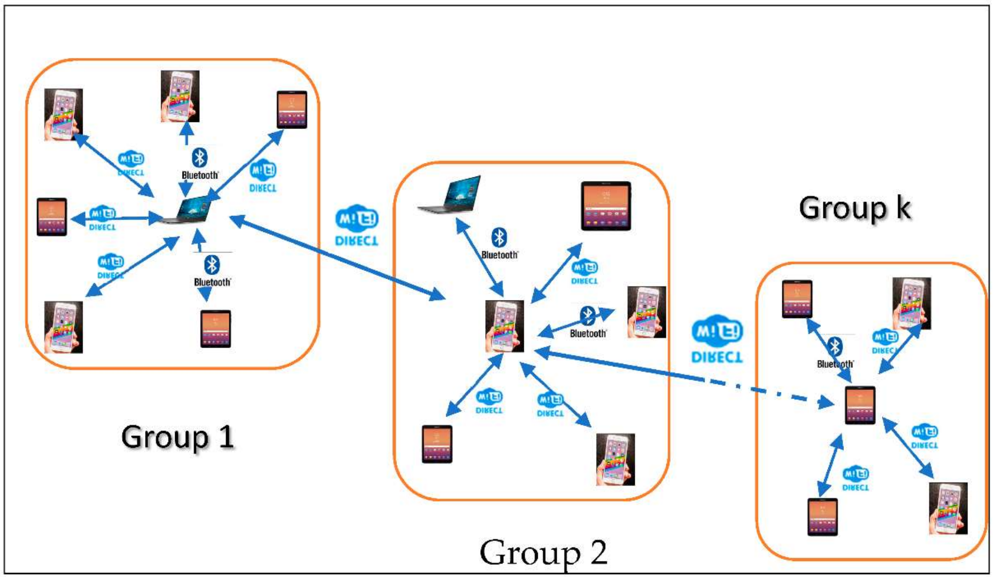


Figure 2. Groups of objects (devices) that communicate by P2P radio connection.

The architectural model we are proposing has the following layers (see details in the Figure 3 of the reference [18]): (i) task application; (ii) service engineering, including two fields: computing modules and operations; (iii) data engineering, consisting of formatting and unification; (iv) adaptation; (v) connectivity, which includes network infrastructure and communication; and (vi) physical [18]. This architecture uses layer pattern architecture to separate the services into different modules. A layer offers a set of cohesive services (modules) exposed into public interfaces.

This APCAM model uses the following integrated wireless technologies that include Wi-Fi in conjunction with a device-to-device connection, such as Wi-Fi Direct, and it is enhanced with wireless connectivity provided by Pycom's LoPy4. LoPy4 is a powerful CPU, Bluetooth low energy (BLE) and state-of-the-art Wi-Fi radio with a range of 1 km [53]. Therefore, these are used by our model as special base stations, especially in extreme situations where Wi-Fi wireless technologies are lost. Our research aims to overcome the difficulties associated with creating an efficient mobile network system by direct radio communication (P2P) between devices that can be clustered into a chain of groups as shown in Figure 2. This network can be created automatically and spontaneously from distributed groups of smart devices by adopting an assembly mechanism based on the extended coverage of connections. This mechanism also allows devices to define their role so they can create an efficient logical topology for intergroup communications [42]. It consists of identifying smart proximity zones according to factors such as the context of the problem, the distance between devices, communication range, network traffic and resources capacities. This makes the model promising as it can be used for effective real-time communication between groups.

The APCAM model combines communication technologies to ensure the continuity of communications in various situations. In this way, by using Pycom's LoPy4, we established our model of communication by integrating several modern wireless and P2P radio technologies, including Wi-Fi, Wi-Fi Direct, Bluetooth and Lora. Our technique manages the communication between devices, especially those deployed in areas outside the coverage of Wi-Fi, in terms of the available alternative communication technologies.

The network of the APCAM model can be classified into two categories according to their ability to communicate with LoPy4. Therefore, our strategy is based on devices that are not in the communication coverage range of LoPy4 and that can be connected through device-to-device radio communication technology (Figure 2). We have also adopted another strategy for devices in the communication coverage range of LoPy4 to integrate device-to-device radio communication and Radio Wi-Fi technology offered by LoPy4 (Figure 3). Both strategies rely on clustering devices into groups, each of which chooses its own manager or group owner to communicate with each other. These strategies include designing appropriate algorithms and applications to operate effectively to manage groups of devices and their intra and interconnections, as well as sharing their information. It is a collaborative distributed environment that enables the proper functioning of the network, considering the availability of the participating devices, their energy and computing capabilities, the average speed of data transfers and the connection coverage area [42].

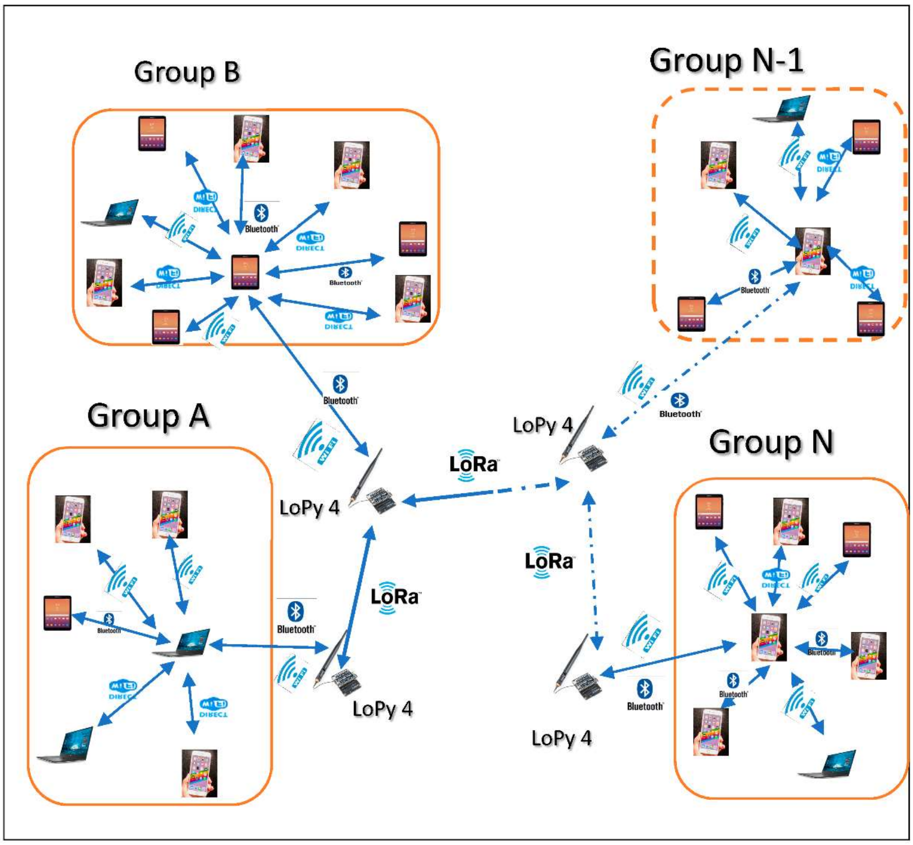


Figure 3. Groups of objects (devices) that communicate using Pycom's LoPy4.

3.3. Structure of the Connectivity Model

This communication model of maintaining connectivity is based on a network of many Pycom's LoPy4 microcontroller boards used as mini station bases for communication. These LoPy4s are programmed in this model in three modes with Python in Visual Studio Code. These modes are: LoPy transmitter, LoPy bridge (receiver/transmitter) and LoPy receiver (see Figure 4).

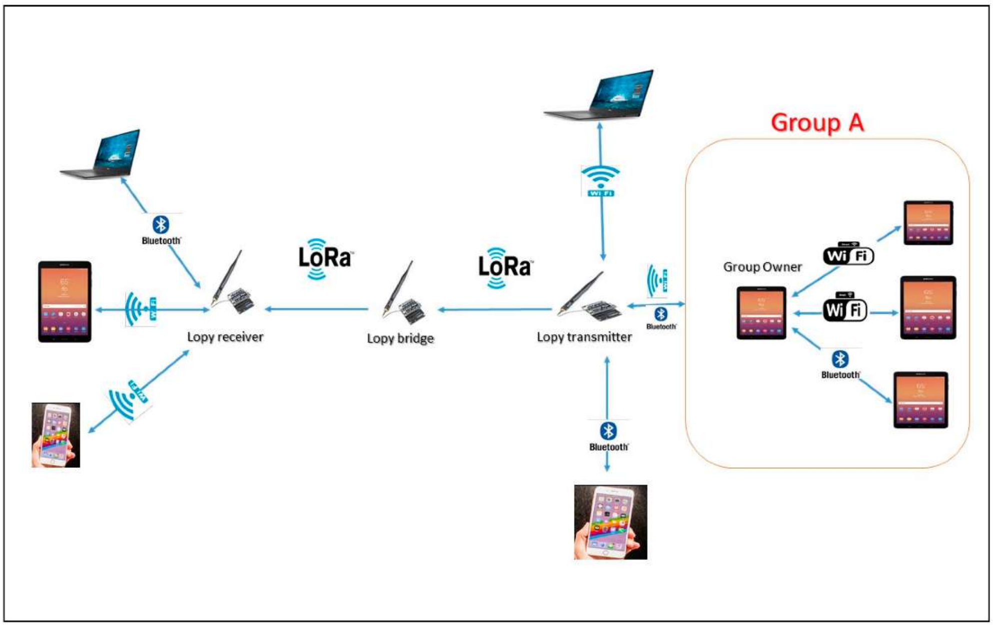


Figure 4. Connectivity Model Structure.

3.4. Communication Behaviours of the APCAM Model

The devices can communicate, through the mobile applications implemented there, with the LoPy4s directly or through their group owners according to the following scenario (see Figure 5). The LoPy transmitter can receive data through Wi-Fi or Bluetooth.

(i) Bluetooth: LoPy4 can receive the information by Bluetooth using any device opening its Bluetooth option.

(ii) Wi-Fi: we created LoPy4 as a Wi-Fi access point such that any device (Android, IOS, Windows or Linux) that connects to this Wi-Fi can communicate with LoPy4 at this address: http://192.168.4.1/test using HTTP post socket.

(iii) Wi-Fi Direct: The devices are split into groups using mobile applications that we have implemented there based on Wi-Fi Direct technology. They use these applications to indicate their group owners. The group owner stores all information received from other group members in a local database. Then, it sends this information to the LoPy4 using a HTTP post socket.

As soon as the LoPy4 transmitter receives the information by Bluetooth or Wi-Fi, it transmits it to the LoPy4 bridge using the Lora socket.

The LoPy bridge is used to connect two remote LoPys: the LoPy transmitter and the LoPy receiver. It receives the information from the LoPy transmitter and sends it to the LoPy receiver. If it is not functional, the LoPy receiver cannot receive the information.

The LoPy receiver is also used as a Wi-Fi access point. All the devices (Android, IOS, Windows and Linux) that are connected to it can have the information that it receives from the LoPy4 transmitter via the LoPy4 bridge in the form of a table.

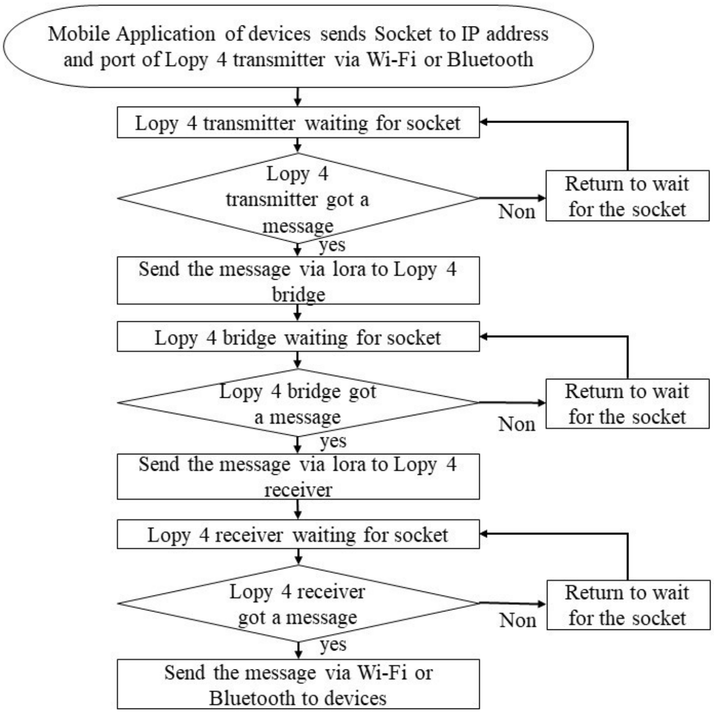

Mobile Application of devices sends Socket to IP address and port of Lopy 4 transmitter via Wi-Fi or Bluetooth
Lopy 4 transmitter waiting for socket
Lopy 4 transmitter got a message
Non
Return to wait for the socket
yes
Send the message via lora to Lopy 4 bridge
Lopy 4 bridge waiting for socket
Lopy 4 bridge got a message
Non
Return to wait for the socket
yes
Send the message via lora to Lopy 4 receiver
Lopy 4 receiver waiting for socket
Lopy 4 receiver got a message
Non
Return to wait for the socket
yes
Send the message via Wi-Fi or Bluetooth to devices

Figure 5. A visualization of the APCAM Model.

3.5. Constructing Connectivity Algorithms to Manage the APCAM Model

The processes of synthesizing, managing and connecting materials in our network model are based on the following algorithms, which design the network as a series of mobile device groups. All the variables used in these algorithms are summarized in Table 2.

Table 2. The variables of APCAM models and their description.

Variable	Description
ValBattery	Battery energy
ValStock	Storage capacity
ValBandwidth	Bandwidth
ValRam	RAM capacity
ValCPU	CPU capacity
ValScore	Score value
PrcBattery	Percentage of battery influence
PrcStock	Percentage of storage influence
PrcBandwidth	Percentage of bandwidth influence
PrcRam	Percentage of RAM influence
PrcCPU	Percentage of CPU influence
Ip_d, ip_q, or ip_go	The IP address of device d, q or group owner
$list_d(2, n)$	An array of two rows and n columns for device d
$n = list_d.length$	Number of devices in $list_d$
tmp_dvc	Temporary device
tmp_scr	Temporary score

(1) Algorithm 1 to calculate the score of device d: CalculatingScores(d): This algorithm, as presented in [54], calculates the score of each device based on five criteria. This score is required to select a group manager (or a group owner) and in the process of routing messages between devices. For the sake of clarity and completeness, it is described below.

Algorithm 1. CalculatingScores(d)

```
Begin
   get factor capacity values of d;
   ValScore = ValBattery ∗ PrcBattery + ValStock ∗ PrcStock + ValBandwidth ∗ PrcBandwidth +
ValRam ∗ PrcRam + ValCPU ∗ PrcCPU;
   return ValScore;
End
```

(2) Algorithm 2 that allows a device to create a list of its connected devices and their scores: This algorithm allows a device such as "d" to create a list of devices connected to it with their scores. When device "q" is connected to device "d", it sends its resource information to device "d." Then, device "d" calculates the score of device "q" by applying the algorithm CalculatingScores(q), then it adds device q to its list identified by two parameters: IP address and score.

Algorithm 2. Creating_List_Devices_Scores(d)

```
int i = 0, j = 0, score = 0;
while connection is active and $i \leq n$ // In case of Wifi and P2P, a group does not exceed n = 12
devices
{
   If device q is connecting to device d {
       $list_d$[i][j] $= ip\_q;$
       score = CalculatingScores(q);
       j = j + 1
       $list_d$[i][j] $= score;$
       i = $i + 1$;
       j = $j - 1$; // value of j is 0 or 1, ex: $list_d[0][0] = ip\_q,$
              // $list_d[0][1] = score, list_d[1][0] =, list_d[1][1] =$, etc.
   }//End if
} //End while
Return $list_d$
End
```

(3) Algorithm 3 to sort the list device scores (Sorting_Devices_by_Scores): Bubble sorting is an algorithm that progressively moves up the largest elements of an array [55].

Algorithm 3. Sorting_Devices_By_Scores (ip_d, $list_d$)

```
// sorting based on the scores
int i, j, n, k;
Begin
  $n = list_d.length;$
   For (i = n; i> = 1; i –)
   {
         For (j = 2; j <= i; j ++)
         {
               k = 1; // rank of the line
               If $list_d$[k][j−1] > $list_d$[k][j] Then
               {
                    tmp_scr = $list_d$[k][j−1];
                    tmp_dvc = $list_d$[k+1][j−1];
                    $list_d$[k][j−1] = [list]_d [k][j];
                    $list_d$[k][j] = tmp_scr;
                    $list_d$[k+1][j] = tmp_dvc;
               } End If
         } End For
   } End For
   return $list_d$;
End
```

(4) Algorithm 4 to select the group manager: Selecting_Group_Manager: Let us randomly choose one of the n-connected devices d_i, $i = 1, \ldots, n$, to be the group owner (GO). It periodically receives data, including scores, from other members at various moments in time, such as every 5 min. At each of these moments, the GO begins to compare iteratively the scores of the members of its group. It starts by comparing itself with resources and scores such as d_i with d_{i+1}. Consequently, the new GO becomes the device with the highest score. Again, by increasing index i by 1, the group owner GO will be compared with d_{i+1} to determine the new group owner until $i = n - 1$.

Algorithm 4. Selecting_Group_Manager

```
int n;
Begin
    $list_d$ = Sorting_Devices_By_Scores($list_d$);
    $n = list_d.length$;
    ip_go = $list_d$[1][n]; // Max($list_d$)
    return ip_go; // Go: group owner
End
```

(5) Algorithm 5 Sending_Message (IP, IP) uses the previous algorithms to reach the target user (object). This algorithm routes a message from device A to target device B in areas covered by device-to-device radio connections such as Lora and Wi-Fi Direct. This algorithm is an iterative algorithm in which the device that has the message (at the beginning it is A) creates its list of devices connected to it with their scores. Then, it looks for device B to check if B can be reached using this device's list to send it the message and accomplish its task. Otherwise, it searches through its list to find the device with the next highest score to B. This algorithm takes advantage of the device's mobility to allow the device carrying the message to move to find in its list of connected devices the appropriate one to send this message. It keeps doing this until it meets the intended device B. We have a dynamic graph because the nodes can be entered in and out of it continuously. These devices are all in constant motion. Therefore, the device carrying the message must move to communicate with the others. Depending on the latitude and longitude of the nodes between B, the device carrying the message and the candidate to carry it, we determine their positions relative to each other. Then, the distances are compared between B and each of the two nodes. The carrier of the message sends it to the other if it has the shortest distance to B. For example, let M be the device carrying the message and N be the candidate to carry it. We calculate the distances MB and NB, and if NB < MB then M sends the message to N.

Before closing this section, let us say a word about the running times of these algorithms, which are, as we will see shortly, very efficient. Indeed, Algorithm 1 is obviously of $O(1)$-time. The time complexity of Algorithm 2 is dominated by the *while* loop iterated at most n times. Since the core of this loop takes $O(1)$-time, it follows that Algorithm 2 is $O(n)$. The time complexity of Algorithm 3 is dominated by the two loops nested one inside the other. Again, as the corps of the two loops is $O(1)$-time, it then follows that the global running time of Algorithm 3 is $O\left(n^2\right)$. Algorithm 4 is obviously of $O(1)$-time. Now, for Algorithm 5, there are two things to distinguish. First, the inner loop *for* runs in $O(n)$-time, whereas the running time of the outer loop while depends on the input. Indeed, as long as condition (ip ! = ip_B) is met, the loop while does not terminate. The device carrying the message moves to find an appropriate place in the list of devices connected to it to receive this message. This continues until it meets the intended device B. Therefore, in the worst case, the test will be done in $O(n)$ time, as we have n devices. Therefore, the overall time complexity of Algorithm 5 is $O\left(n^2\right)$.

Algorithm 5. for Sending_Message (ip$_A$, ip$_B$)

```
Local variables:
int n;
double n = 0; $distance_{TR} = 0$; $x_1 = 0$; $x_2 = 0$; $y_1 = 0$; $y_2 = 0$;
$distance_{ipB} = 0$; $distance_{ip_{max}B} = 0$;
String $ip_{act}$;// IP of the device being processed
String [][] $list_{act}$; // Array of the devices connected to the device being processed
String [][] $list_{act\_srt}$; // Array of devices sorted according to scores and connected to the device
being processed
String min_scr; // threshold score Boolean flag = false
Begin
    ip = ip$_A$;$x_1 = x_{ip}$; $y_1 = y_{ip;}$
    $x_2 = x_B$; $y_2 = y_B$;
    $distance_{TR} = (x_2 - x_1)^2 - (y_2 - y_1)^2$; // distance between transmitter and receiver (Here, A
and B)
    While ip != ip$_B${
    $list_{act}$ = Creating_List_Devices_Scores(ip); // creating the list connected to the device
    $list_{act\_srt}$ = Sorting_Devices_By_Scores(ip, $list_{act}$);
    n = $list_{act\_srt}[0]$.length; // number elements in this array
    // check if B belongs to the list of devices connected to the device being processed
      For (i = n−1; i ≥ n; i–) {
          $ip_{max}$ = $list_{act\_srt}[1][i]$; // starting by the device having the maximum score
If $ip_{max}$ = ip$_{B;}$ //the device being processed can
              //communicate the message directly to B
          Exit //as the message has arrived at its
              // destination B
          Else{
          $x_1 = x_{ip}$; $y_1 = y_{ip}$;
          $x_3 = x_{ip_{max}}$; $y_3 = y_{ip_{max}}$;
          $distance_{ipB} = (x_B - x_1)^2 - (y_B - y_1)^2$; // distance between ip and B
                $distance_{ip_{max}B} = (x_B - x_3)^2 - (y_B - y_3)^2$; // distance between $ip_{max}$ and B
$If\ distance_{ip_{max}B}$ < $distance_{ipB}$ and min_scr ≤ $list_{act\_srt}[0]$ [i] {
//$ip_{max}$ is closer to B and its score > threshold score
              // the message is deposited at $ip_{max}$
              $ip = ip_{max}$;
                  }//End If
          }//End If
      } // End For
    } // End While
End Algorithm
```

3.6. Operating Mechanism and Security Controls

We implemented applications of the previous algorithms with others that are needed to manage the model at the levels of devices, their groups, and message circulation through and between these groups. These applications must be installed on the participating devices. Among them, one is to form groups and elect their owners. Another is the client/server application where the server is supposed to be the owner of the group which sends its IP address and its communication gateway number to clients who are members of that group and to other group owners. Then, through these coordinates they exchange messages and information. The third one of these applications is to program the Lopys as transmitter, bridge and receiver.

The model is generally designed to be open for public use in harsh situations to allow participation and rescue for everyone. Thus, the architecture is proposed with exposed and unencrypted IP addresses such as http://192.168.4.1/test. In practice, this makes it non-secure and therefore dangerous to put into service. To remedy this situation of a public Lopy network, we have used a password for participants that is currently "Wi-Fi". Likewise, we can change the public IP address of the page to a private one. To make the system more secure, we can also add an identification page to identify each user. Thus, each user must have a login and a password. For more security, we can further create a database via a registration form to the network of Lopys to verify the connection of the participants. However, it must be taken into account that the more important the security measures, the lower the chances are of the general public participating in and benefitting from this model. Therefore, the lower the chances are of rescuing people in harsh and catastrophic situations.

4. Experimental Study

Relying on the type of the available communication technology, we classified the experimental study we conducted into two categories. Category A uses device-to-device radio communication technology only, such as Wi-Fi Direct, and Category B combines this technology with others that are available via Pycom's LoPy4, such as Radio Wi-Fi, Lora and Bluetooth. In both cases, we used a combination of mobile phones, tablets and laptops. In case A, we used mobiles and tablets, whereas in case B, we used laptops. The characteristics of several of the devices we used in this experimental study are:

Mobiles: (i) Smartphone Samsung Galaxy S21: CPU Octa-core (1 × 2.9 GHz Cortex-X1 & 3 × 2.80 GHz Cortex-A78 & 4 × 2.2 GHz Cortex-A55)—International, OS Android 11, One UI 3.1, 256 GB RAM, 512 GB storage. (ii) Smartphone Samsung Galaxy S6: Processor Octa-core, 2100 MHz, ARM Cortex-A57 and ARM Cortex-A53, 64-bit—14 nm, RAM 3 GB LPDDR4, OS Android 7.0 Nougat, 64 GB storage. (iii) Smartphone LG K40: OS Android 8.1 (Oreo), LG UX 7, CPU Octa-core 2.0 GHz Cortex-A53, 2 GB RAM, 32 GB storage.
Tablets: (i) Samsung Galaxy Tab A 10.1: Samsung Exynos 7904 CPU, 2 GB RAM, Operating System Android 9 with One UI, 32 GB storage. (ii) Samsung Galaxy Tab A 8.0: Processor Qualcomm Snapdragon 429 4 × 2 GHz, Cortex-A53, 2 GB RAM, Operating System Android 9.0 Pie, 32 GB storage. (iii) Samsung Galaxy Tab 7.0 Plus: processor 1.5 GHz quad-core, 1.5 GB of RAM, Operating System Android, 8 GB storage.
Laptops: (i) Laptop 1 with the following characteristics: Intel® Core™ i7 processor, x64-based processor, Windows 10 Pro 64-bit OS, 2.3 GHz CPU, and 16 GB of RAM. (ii) Laptop 2 with the following characteristics: Intel® Core™ i7 6820HP CPU, 2.7 GHz, x64-based processor, 64-bit OS, and 16 GB of RAM. (iii) MacBook Pro with the following characteristics: Intel Core i7 quad core 2.8 GHz, macOS High Sierra and 16 Go of integrated memory LPDDR3 2 133 MHz.

Let us point out the data we used are generated live from the devices between which they were exchanged. The exchanged messages contain the information of the devices and the users.

4.1. Category A

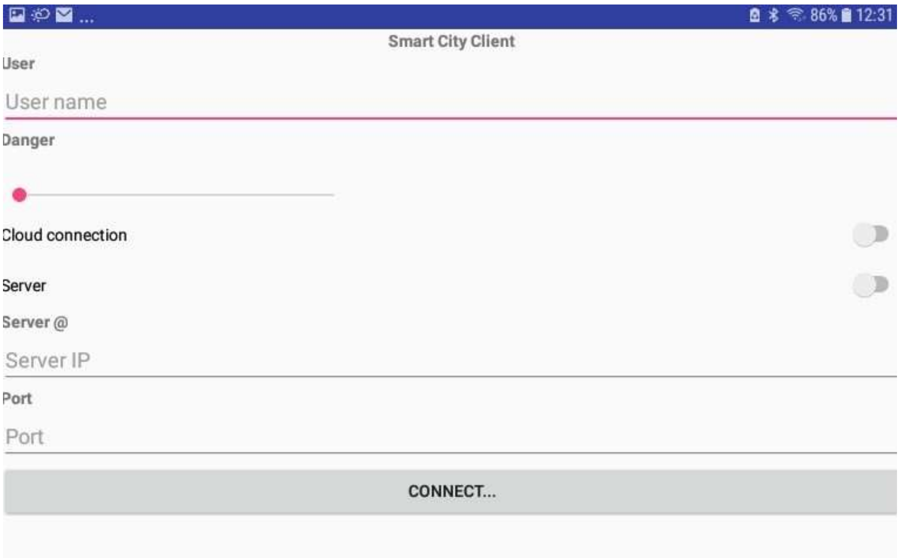
This category includes two types of techniques to achieve and enhance speed and communication in transmitting information:

(1) Finding the shortest path of maintained communication between two devices in the network through which information can be transferred as messages from one device to another. Ensuring the connectivity of the shortest path nodes are presented in the following three case studies [56]:

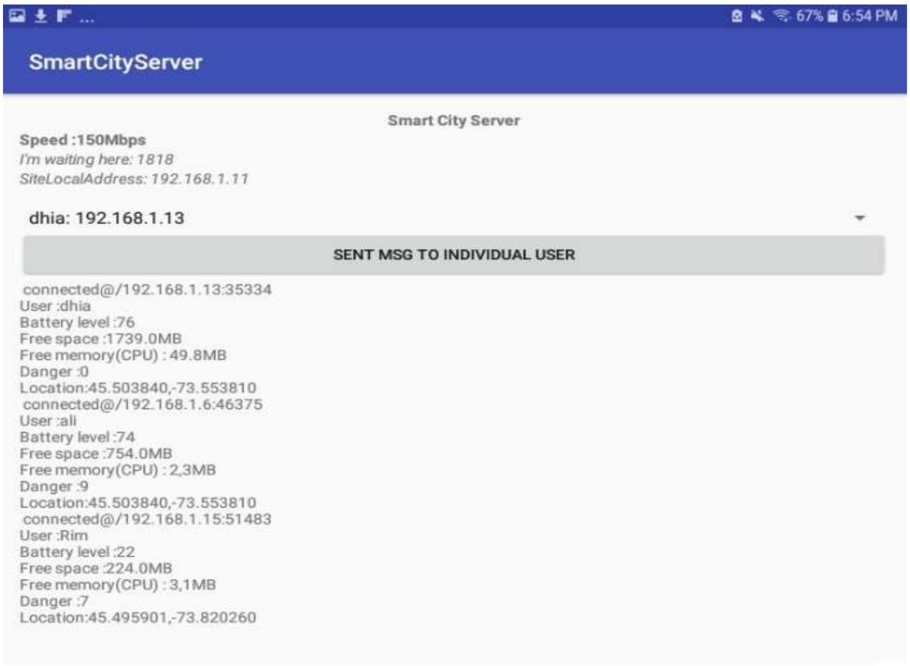
   (i) The nodes are obviously connected.
   (ii) Nodes must be moved.
   (iii) Using and moving a drone with the capability of P2P wireless technology.

(2) Establishing a network by forming a chain of adjacent groups of interfering devices within the available communication range. A device belonging to two groups can act as a gateway between the owners of these groups. The message must be sent in an intelligent way, taking into consideration the distance to the destination device and the continuity of communication through nearby devices. In the following, we present the application interface of forming groups and selecting their owner. An analysis

of the data available from the device's interface in Figure 6 shows that the information about devices in its connection range is periodically communicated to it. In addition, the GO is also changed when the score varies. Different parts are used to show the most important components of the graphical interfaces of our system.



Part 1. Client menu of the device's connection applications



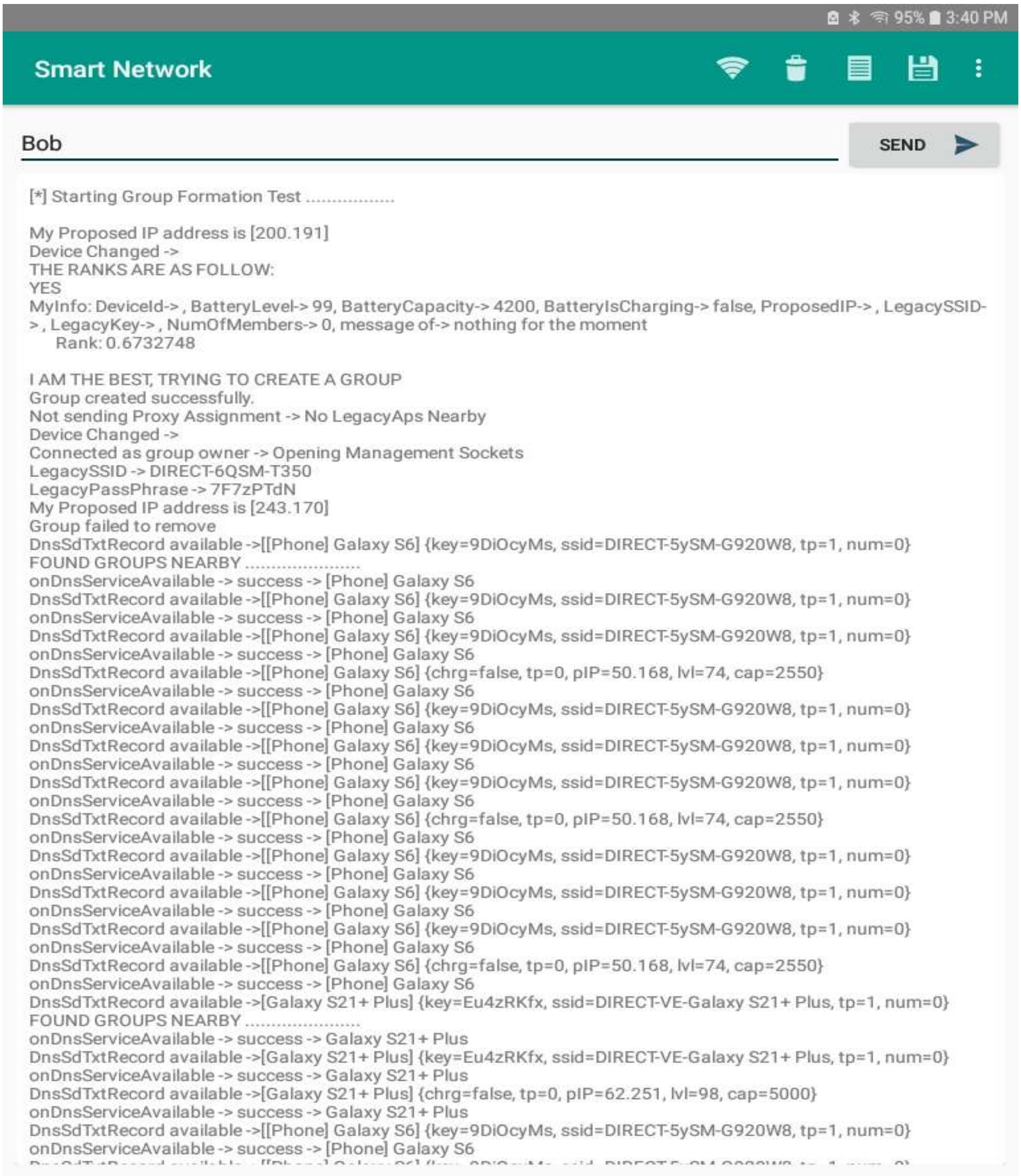
```
Speed :150Mbps
I'm waiting here: 1818
SiteLocalAddress: 192.168.1.11

dhia: 192.168.1.13

 connected@/192.168.1.13:35334
User :dhia
Battery level :76
Free space :1739.0MB
Free memory(CPU) : 49.8MB
Danger :0
Location:45.503840,-73.553810
 connected@/192.168.1.6:46375
User :ali
Battery level :74
Free space :754.0MB
Free memory(CPU) : 2,3MB
Danger :9
Location:45.503840,-73.553810
 connected@/192.168.1.15:51483
User :Rim
Battery level :22
Free space :224.0MB
Free memory(CPU) : 3,1MB
Danger :7
Location:45.495901,-73.820260
```

Part 1. Server menu of the device's connection applications

Figure 6. *Cont.*



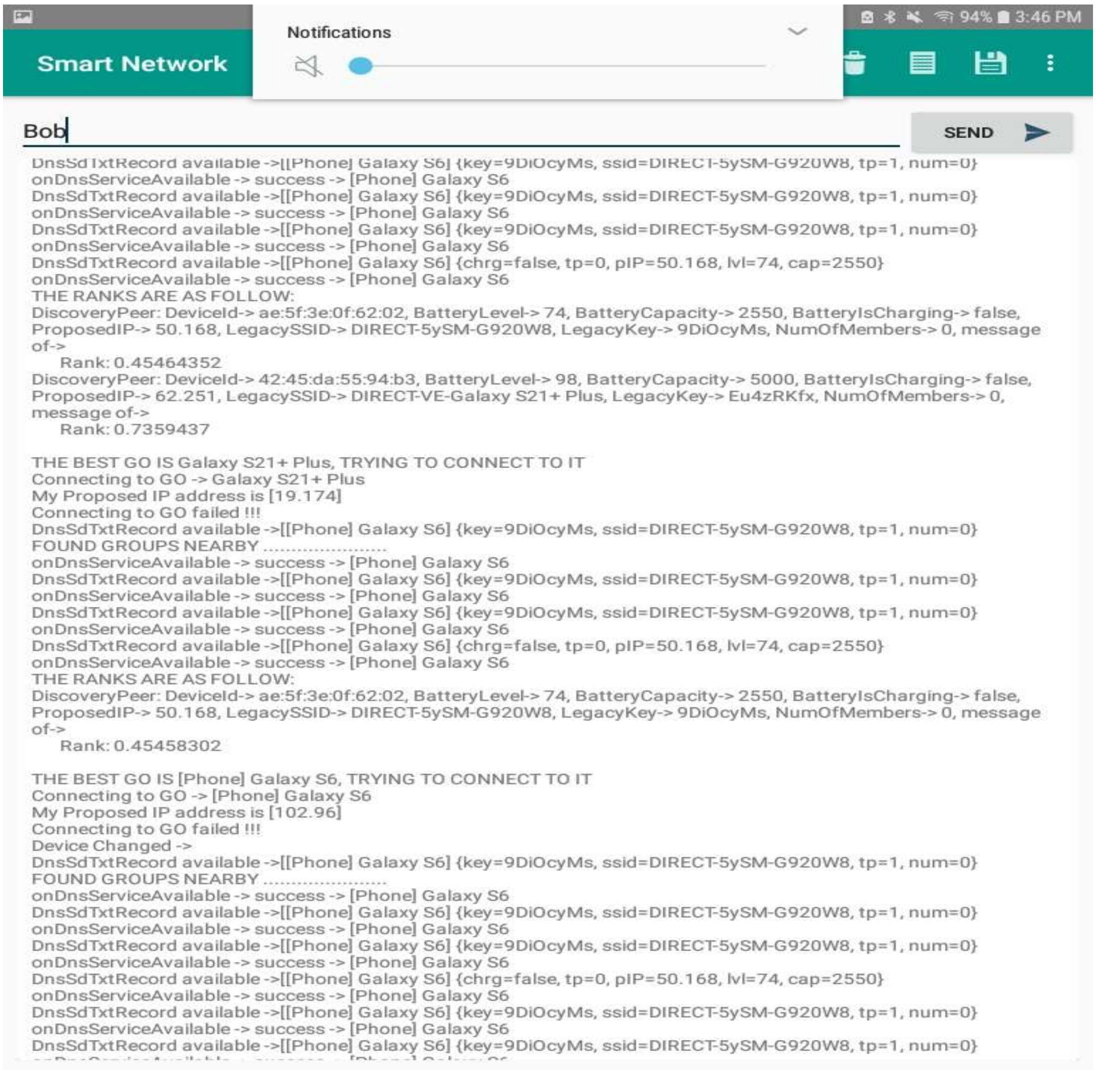
```
[*] Starting Group Formation Test .................

My Proposed IP address is [200.191]
Device Changed ->
THE RANKS ARE AS FOLLOW:
YES
MyInfo: DeviceId-> , BatteryLevel-> 99, BatteryCapacity-> 4200, BatteryIsCharging-> false, ProposedIP-> , LegacySSID-
> , LegacyKey-> , NumOfMembers-> 0, message of-> nothing for the moment
    Rank: 0.6732748

I AM THE BEST, TRYING TO CREATE A GROUP
Group created successfully.
Not sending Proxy Assignment -> No LegacyAps Nearby
Device Changed ->
Connected as group owner -> Opening Management Sockets
LegacySSID -> DIRECT-6QSM-T350
LegacyPassPhrase -> 7F7zPTdN
My Proposed IP address is [243.170]
Group failed to remove
DnsSdTxtRecord available ->[[Phone] Galaxy S6] {key=9DiOcyMs, ssid=DIRECT-5ySM-G920W8, tp=1, num=0}
FOUND GROUPS NEARBY .......................
onDnsServiceAvailable -> success -> [Phone] Galaxy S6
DnsSdTxtRecord available ->[[Phone] Galaxy S6] {key=9DiOcyMs, ssid=DIRECT-5ySM-G920W8, tp=1, num=0}
onDnsServiceAvailable -> success -> [Phone] Galaxy S6
DnsSdTxtRecord available ->[[Phone] Galaxy S6] {key=9DiOcyMs, ssid=DIRECT-5ySM-G920W8, tp=1, num=0}
onDnsServiceAvailable -> success -> [Phone] Galaxy S6
DnsSdTxtRecord available ->[[Phone] Galaxy S6] {chrg=false, tp=0, pIP=50.168, lvl=74, cap=2550}
onDnsServiceAvailable -> success -> [Phone] Galaxy S6
DnsSdTxtRecord available ->[[Phone] Galaxy S6] {key=9DiOcyMs, ssid=DIRECT-5ySM-G920W8, tp=1, num=0}
onDnsServiceAvailable -> success -> [Phone] Galaxy S6
DnsSdTxtRecord available ->[[Phone] Galaxy S6] {key=9DiOcyMs, ssid=DIRECT-5ySM-G920W8, tp=1, num=0}
onDnsServiceAvailable -> success -> [Phone] Galaxy S6
DnsSdTxtRecord available ->[[Phone] Galaxy S6] {key=9DiOcyMs, ssid=DIRECT-5ySM-G920W8, tp=1, num=0}
onDnsServiceAvailable -> success -> [Phone] Galaxy S6
DnsSdTxtRecord available ->[[Phone] Galaxy S6] {chrg=false, tp=0, pIP=50.168, lvl=74, cap=2550}
onDnsServiceAvailable -> success -> [Phone] Galaxy S6
DnsSdTxtRecord available ->[[Phone] Galaxy S6] {key=9DiOcyMs, ssid=DIRECT-5ySM-G920W8, tp=1, num=0}
onDnsServiceAvailable -> success -> [Phone] Galaxy S6
DnsSdTxtRecord available ->[[Phone] Galaxy S6] {key=9DiOcyMs, ssid=DIRECT-5ySM-G920W8, tp=1, num=0}
onDnsServiceAvailable -> success -> [Phone] Galaxy S6
DnsSdTxtRecord available ->[[Phone] Galaxy S6] {key=9DiOcyMs, ssid=DIRECT-5ySM-G920W8, tp=1, num=0}
onDnsServiceAvailable -> success -> [Phone] Galaxy S6
DnsSdTxtRecord available ->[[Phone] Galaxy S6] {chrg=false, tp=0, pIP=50.168, lvl=74, cap=2550}
onDnsServiceAvailable -> success -> [Phone] Galaxy S6
DnsSdTxtRecord available ->[Galaxy S21+ Plus] {key=Eu4zRKfx, ssid=DIRECT-VE-Galaxy S21+ Plus, tp=1, num=0}
FOUND GROUPS NEARBY .......................
onDnsServiceAvailable -> success -> Galaxy S21+ Plus
DnsSdTxtRecord available ->[Galaxy S21+ Plus] {key=Eu4zRKfx, ssid=DIRECT-VE-Galaxy S21+ Plus, tp=1, num=0}
onDnsServiceAvailable -> success -> Galaxy S21+ Plus
DnsSdTxtRecord available ->[Galaxy S21+ Plus] {chrg=false, tp=0, pIP=62.251, lvl=98, cap=5000}
onDnsServiceAvailable -> success -> Galaxy S21+ Plus
DnsSdTxtRecord available ->[[Phone] Galaxy S6] {key=9DiOcyMs, ssid=DIRECT-5ySM-G920W8, tp=1, num=0}
onDnsServiceAvailable -> success -> [Phone] Galaxy S6
```

Part 2: Forming a group of devices and selecting the group owner

Figure 6. *Cont.*



Bob

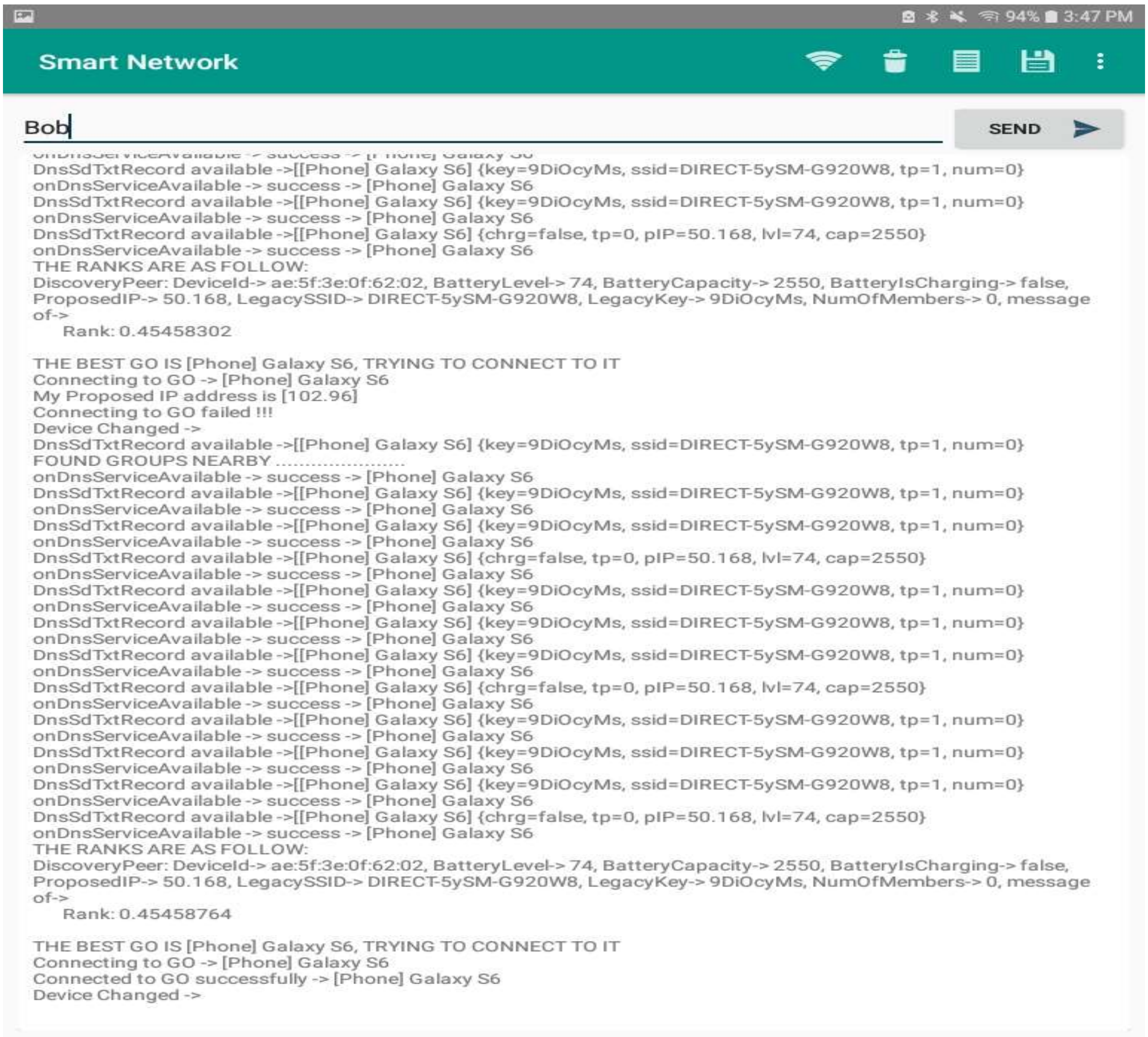
```
DnsSdTxtRecord available ->[[Phone] Galaxy S6] {key=9DiOcyMs, ssid=DIRECT-5ySM-G920W8, tp=1, num=0}
onDnsServiceAvailable -> success -> [Phone] Galaxy S6
DnsSdTxtRecord available ->[[Phone] Galaxy S6] {key=9DiOcyMs, ssid=DIRECT-5ySM-G920W8, tp=1, num=0}
onDnsServiceAvailable -> success -> [Phone] Galaxy S6
DnsSdTxtRecord available ->[[Phone] Galaxy S6] {key=9DiOcyMs, ssid=DIRECT-5ySM-G920W8, tp=1, num=0}
onDnsServiceAvailable -> success -> [Phone] Galaxy S6
DnsSdTxtRecord available ->[[Phone] Galaxy S6] {chrg=false, tp=0, pIP=50.168, lvl=74, cap=2550}
onDnsServiceAvailable -> success -> [Phone] Galaxy S6
THE RANKS ARE AS FOLLOW:
DiscoveryPeer: DeviceId-> ae:5f:3e:0f:62:02, BatteryLevel-> 74, BatteryCapacity-> 2550, BatteryIsCharging-> false,
ProposedIP-> 50.168, LegacySSID-> DIRECT-5ySM-G920W8, LegacyKey-> 9DiOcyMs, NumOfMembers-> 0, message
of->
    Rank: 0.45464352
DiscoveryPeer: DeviceId-> 42:45:da:55:94:b3, BatteryLevel-> 98, BatteryCapacity-> 5000, BatteryIsCharging-> false,
ProposedIP-> 62.251, LegacySSID-> DIRECT-VE-Galaxy S21+ Plus, LegacyKey-> Eu4zRKfx, NumOfMembers-> 0,
message of->
    Rank: 0.7359437

THE BEST GO IS Galaxy S21+ Plus, TRYING TO CONNECT TO IT
Connecting to GO -> Galaxy S21+ Plus
My Proposed IP address is [19.174]
Connecting to GO failed !!!
DnsSdTxtRecord available ->[[Phone] Galaxy S6] {key=9DiOcyMs, ssid=DIRECT-5ySM-G920W8, tp=1, num=0}
FOUND GROUPS NEARBY ......................
onDnsServiceAvailable -> success -> [Phone] Galaxy S6
DnsSdTxtRecord available ->[[Phone] Galaxy S6] {key=9DiOcyMs, ssid=DIRECT-5ySM-G920W8, tp=1, num=0}
onDnsServiceAvailable -> success -> [Phone] Galaxy S6
DnsSdTxtRecord available ->[[Phone] Galaxy S6] {key=9DiOcyMs, ssid=DIRECT-5ySM-G920W8, tp=1, num=0}
onDnsServiceAvailable -> success -> [Phone] Galaxy S6
DnsSdTxtRecord available ->[[Phone] Galaxy S6] {chrg=false, tp=0, pIP=50.168, lvl=74, cap=2550}
onDnsServiceAvailable -> success -> [Phone] Galaxy S6
THE RANKS ARE AS FOLLOW:
DiscoveryPeer: DeviceId-> ae:5f:3e:0f:62:02, BatteryLevel-> 74, BatteryCapacity-> 2550, BatteryIsCharging-> false,
ProposedIP-> 50.168, LegacySSID-> DIRECT-5ySM-G920W8, LegacyKey-> 9DiOcyMs, NumOfMembers-> 0, message
of->
    Rank: 0.45458302

THE BEST GO IS [Phone] Galaxy S6, TRYING TO CONNECT TO IT
Connecting to GO -> [Phone] Galaxy S6
My Proposed IP address is [102.96]
Connecting to GO failed !!!
Device Changed ->
DnsSdTxtRecord available ->[[Phone] Galaxy S6] {key=9DiOcyMs, ssid=DIRECT-5ySM-G920W8, tp=1, num=0}
FOUND GROUPS NEARBY ......................
onDnsServiceAvailable -> success -> [Phone] Galaxy S6
DnsSdTxtRecord available ->[[Phone] Galaxy S6] {key=9DiOcyMs, ssid=DIRECT-5ySM-G920W8, tp=1, num=0}
onDnsServiceAvailable -> success -> [Phone] Galaxy S6
DnsSdTxtRecord available ->[[Phone] Galaxy S6] {key=9DiOcyMs, ssid=DIRECT-5ySM-G920W8, tp=1, num=0}
onDnsServiceAvailable -> success -> [Phone] Galaxy S6
DnsSdTxtRecord available ->[[Phone] Galaxy S6] {chrg=false, tp=0, pIP=50.168, lvl=74, cap=2550}
onDnsServiceAvailable -> success -> [Phone] Galaxy S6
DnsSdTxtRecord available ->[[Phone] Galaxy S6] {key=9DiOcyMs, ssid=DIRECT-5ySM-G920W8, tp=1, num=0}
onDnsServiceAvailable -> success -> [Phone] Galaxy S6
DnsSdTxtRecord available ->[[Phone] Galaxy S6] {key=9DiOcyMs, ssid=DIRECT-5ySM-G920W8, tp=1, num=0}
```

Part 3: Sequel of Part 2, menu for forming a group of devices and selecting the group owner

Figure 6. *Cont.*



```
onDnsServiceAvailable -> success -> [Phone] Galaxy S6
DnsSdTxtRecord available ->[[Phone] Galaxy S6] {key=9DiOcyMs, ssid=DIRECT-5ySM-G920W8, tp=1, num=0}
onDnsServiceAvailable -> success -> [Phone] Galaxy S6
DnsSdTxtRecord available ->[[Phone] Galaxy S6] {key=9DiOcyMs, ssid=DIRECT-5ySM-G920W8, tp=1, num=0}
onDnsServiceAvailable -> success -> [Phone] Galaxy S6
DnsSdTxtRecord available ->[[Phone] Galaxy S6] {chrg=false, tp=0, pIP=50.168, lvl=74, cap=2550}
onDnsServiceAvailable -> success -> [Phone] Galaxy S6
THE RANKS ARE AS FOLLOW:
DiscoveryPeer: DeviceId-> ae:5f:3e:0f:62:02, BatteryLevel-> 74, BatteryCapacity-> 2550, BatteryIsCharging-> false,
ProposedIP-> 50.168, LegacySSID-> DIRECT-5ySM-G920W8, LegacyKey-> 9DiOcyMs, NumOfMembers-> 0, message
of->
    Rank: 0.45458302

THE BEST GO IS [Phone] Galaxy S6, TRYING TO CONNECT TO IT
Connecting to GO -> [Phone] Galaxy S6
My Proposed IP address is [102.96]
Connecting to GO failed !!!
Device Changed ->
DnsSdTxtRecord available ->[[Phone] Galaxy S6] {key=9DiOcyMs, ssid=DIRECT-5ySM-G920W8, tp=1, num=0}
FOUND GROUPS NEARBY .......................
onDnsServiceAvailable -> success -> [Phone] Galaxy S6
DnsSdTxtRecord available ->[[Phone] Galaxy S6] {key=9DiOcyMs, ssid=DIRECT-5ySM-G920W8, tp=1, num=0}
onDnsServiceAvailable -> success -> [Phone] Galaxy S6
DnsSdTxtRecord available ->[[Phone] Galaxy S6] {key=9DiOcyMs, ssid=DIRECT-5ySM-G920W8, tp=1, num=0}
onDnsServiceAvailable -> success -> [Phone] Galaxy S6
DnsSdTxtRecord available ->[[Phone] Galaxy S6] {chrg=false, tp=0, pIP=50.168, lvl=74, cap=2550}
onDnsServiceAvailable -> success -> [Phone] Galaxy S6
DnsSdTxtRecord available ->[[Phone] Galaxy S6] {key=9DiOcyMs, ssid=DIRECT-5ySM-G920W8, tp=1, num=0}
onDnsServiceAvailable -> success -> [Phone] Galaxy S6
DnsSdTxtRecord available ->[[Phone] Galaxy S6] {key=9DiOcyMs, ssid=DIRECT-5ySM-G920W8, tp=1, num=0}
onDnsServiceAvailable -> success -> [Phone] Galaxy S6
DnsSdTxtRecord available ->[[Phone] Galaxy S6] {key=9DiOcyMs, ssid=DIRECT-5ySM-G920W8, tp=1, num=0}
onDnsServiceAvailable -> success -> [Phone] Galaxy S6
DnsSdTxtRecord available ->[[Phone] Galaxy S6] {chrg=false, tp=0, pIP=50.168, lvl=74, cap=2550}
onDnsServiceAvailable -> success -> [Phone] Galaxy S6
DnsSdTxtRecord available ->[[Phone] Galaxy S6] {key=9DiOcyMs, ssid=DIRECT-5ySM-G920W8, tp=1, num=0}
onDnsServiceAvailable -> success -> [Phone] Galaxy S6
DnsSdTxtRecord available ->[[Phone] Galaxy S6] {key=9DiOcyMs, ssid=DIRECT-5ySM-G920W8, tp=1, num=0}
onDnsServiceAvailable -> success -> [Phone] Galaxy S6
DnsSdTxtRecord available ->[[Phone] Galaxy S6] {key=9DiOcyMs, ssid=DIRECT-5ySM-G920W8, tp=1, num=0}
onDnsServiceAvailable -> success -> [Phone] Galaxy S6
DnsSdTxtRecord available ->[[Phone] Galaxy S6] {chrg=false, tp=0, pIP=50.168, lvl=74, cap=2550}
onDnsServiceAvailable -> success -> [Phone] Galaxy S6
THE RANKS ARE AS FOLLOW:
DiscoveryPeer: DeviceId-> ae:5f:3e:0f:62:02, BatteryLevel-> 74, BatteryCapacity-> 2550, BatteryIsCharging-> false,
ProposedIP-> 50.168, LegacySSID-> DIRECT-5ySM-G920W8, LegacyKey-> 9DiOcyMs, NumOfMembers-> 0, message
of->
    Rank: 0.45458764

THE BEST GO IS [Phone] Galaxy S6, TRYING TO CONNECT TO IT
Connecting to GO -> [Phone] Galaxy S6
Connected to GO successfully -> [Phone] Galaxy S6
Device Changed ->
```

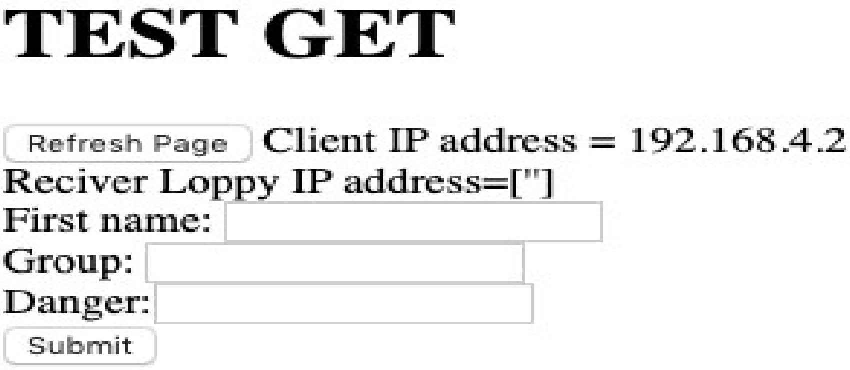
Part 4: Sequel of Part 3, menu for forming a group of devices and selecting the group owner

Figure 6. Application interfaces used on devices.

4.2. Category B

This category integrates connection technology provided by Pycom's LoPy4 with existing Wi-Fi and device-to-device radio communication.

We designed and implemented a mobile application with a graphic interface (Figure 7) to exchange information through our model of connectivity (Figure 5). In the experimental study, we used a network like the one illustrated in Figure 5. We installed the LoPy bridge in a given position. Then, we placed both the LoPy transmitter and the LoPy receiver in cars that drove away from the LoPy bridge in opposite directions. We connected the devices to the LoPy transmitter whenever we travelled 150 m away from the LoPy bridge in both directions. The LoPy bridge received the device's information from the LoPy transmitter and transferred it to the LoPy receiver, which transmitted it to its nearby devices as shown in Tables 3 and 4.



TEST GET

Client IP address = 192.168.4.2

Reciver Loppy IP address=['']

First name:

Group:

Danger:

Figure 7. Transmission of information from LoPy transmitter to LoPy bridge (Receiver).

Table 3. First experimentation study, information required from devices near the LoPy transmitter sent via the LoPy bridge to the LoPy receiver with 192.168.4.1/test. This communication model can be applied between transmitter–bridge as well as between bridge–receiver.

Full Name	Group	Danger	IP
Eric Bouchard	G1	5	192.168.4.2
DIRECT-H8SM-G920W8	G1	1	192.168.4.3
Jad Tremblay	G2	3	192.168.4.6
Jean Richard	G2	4	192.168.4.2
Hugo Michel	G3	7	192.168.4.2
Hanna Hugo	G3	6	192.168.4.2
Luc Besson	G3	6	192.168.4\xae2

Table 4. Second experimentation study (part 2), the same description of Table 3. This communication model can be applied between objects (transmitter–bridge and bridge–receiver).

Full Name	Group	Danger	IP
Ali Jabril	G7	7	192.168.4.3
Mohamad Mickeal	G4	2	192.168.4.3
Alain Legof	N/A	4	Bluetooth
Sam	G5	3	192.168.4.2
Alain April	G3	7	192.168.4.3
DIRECT-ueSM-G920W8	G1	1	192.168.4.5
Dihya Messaoudi	N/A	4	Bluetooth
DIRECT-ueSM-G920W8	G1	1	192.168.4.5

Because the team of volunteers could be scattered and distant in the experimental study, we used the same device in different places with the names of different users.

Throughout this experimental study, our autonomous model is shown to be efficient in providing communication and exchanging information over a range of more than two kilometres. As long as this can be done using a LoPy bridge, this range can be extended proportionally by using additional LoPy bridges.

The experimental study we conducted shows the efficiency of our communication model, which has many advantages over existing models, even those using 5G technology in terms of connection coverage. In fact, the 5G coverage does not exceed 500 m, while it exceeds 1 km through one Pycom's LoPy4. In fact, the 5G coverage range tests on the millimeter wave band produced results in the order of 500 m around the pylon. This requires an important installation of MIMO (multiple input, multiple output) antennas for a single deployment of 5G standalone [57]. Furthermore, our model creates a topology to enhance inter-and intra-group communication. In addition, it selects the objects (devices) with the greatest resources to increase the duration of the connectivity and the functionality of the model. These objects are replaced dynamically by others if their resources decline. Moreover, even if there is no standard wireless or Internet connection, this model can continue to help people communicate using a combination of technologies such as Wi-Fi-Direct, Bluetooth and that of Pycom's LoPy4 from Lora.

5. Conclusions

Ubiquitous smart and pervasive devices are the core elements of the Internet of Things, which is expanding, enhancing and integrating with distributed systems and mobile collaborative computing. The essential factor on which these paradigms are built and enhance their effectiveness is the communication technologies. Connectivity in networks is being improved and modernized by introducing and enhancing more modern wireless technologies as well as P2P connectivity. The main issue of various environments (harsh, urban or rural environment) is to maintain connectivity between objects (users), through both inter- and intra-group connectivity.

In this paper, we presented an autonomous collaborative pervasive architectural model of communication between ubiquitous smart mobile devices, especially those at the ends of networks. We designed four algorithms to increase the availability of communication between objects, taking into account many constraints such as the distributed environment and the mobility and the heterogeneity of these objects.

This model is based on the use of modern wireless communication technologies, in particular device-to-device radio communication. To overcome the short range of radio waves, we integrated Pycom's LoPy4 technology into this model as base stations and access points. The contribution of this model is that it maintains the permanence of the connectivity between its devices in various situations, especially harsh ones. In fact, the results of the experimental study proved the effectiveness of this model in maintaining connectivity using LoPy4 technology across a range of several kilometres in the form of LoPy transmitter, LoPy bridge and LoPy receiver. The proposed distributed collaborative and architectural model mainly improves the performance and efficiency in terms of overcoming difficulties of connectivity and management of pervasive networks, which are automatically and dynamically established in peripheral areas, especially in harsh environments.

For future work, we intend to include other emerging modern technologies of radio communication that could be compatible with LoPy4 so they can be integrated into this model for more effectiveness.

Author Contributions: G.F. designed the pervasive architectural model and its managing algorithms, then he implemented them; G.F., H.M. and D.R. analyzed and validated the computational algorithms; G.F. compared these algorithms and wrote the manuscript in consultation with H.M. and D.R. All authors have read and agreed the published version of the manuscript.

Funding: This work was partially funded by the Natural Sciences and Engineering Research council of Canada (NSERC), Le Fond de l'Université du Québec à Chicoutimi (FUQAC) and Le Décanat de la Recherche et de la Créaction (UQAC).

Institutional Review Board Statement: Not applicable but the University of Quebec at Chicoutimi requires that all research involving the participation of human beings, whether funded or not, conducted or supervised by its professors, employees or students be ethically reviewed. Links: http://recherche.uqac.ca/cer/ (accessed on 3 September 2021).

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable. However, we generated live data from information of the used devices and from messages sent through intermediate devices.

Conflicts of Interest: The authors declare no conflict of interest associated with this publication.

References

1. Kramp, T.; van Kranenburg, R.; Lange, S. Introduction to the Internet of Things. In *Enabling Things to Talk*; Bassi, A., Bauer, M., Fiedler, M., Kramp, T., van Kranenburg, R., Lange, S., Meissner, S., Eds.; Springer: Berlin/Heidelberg, Germany, 2013. [CrossRef]
2. Coulouris, G.; Dollimore, J.; Kindberg, T.; Blair, G. *Distributed Systems: Concepts and Design*, 5th ed.; Addison Wesley: Boston, MA, USA, 2012.
3. L'Informatique de Poche dans l'Entreprise; Les Cahiers d'e-Caria. Available online: https://dochero.tips/linformatique-de-poche-dans-lentreprise.html (accessed on 3 September 2021).
4. Fadlallah, G.; Rebaine, D.; Mcheick, H. Scheduling problems from workshop to collaborative mobile computing: A state of the art. *Int. J. Comput. Sci. Inf. Secur.* **2018**, *16*, 47–69.
5. Abbas, N.; Zhang, Y.; Taherkordi, A.; Skeie, T. Mobile edge computing: A survey. *IEEE Internet Things J.* **2018**, *5*, 450–465. [CrossRef]
6. Kai, K.; Cong, W.; Tao, L. Fog computing for vehicular ad hoc networks: Paradigms, scenarios, and issues. *J. China Univ. Posts Telecommun.* **2016**, *23*, 56–96. [CrossRef]
7. Kar, U.N.; Sanyal, D.K. An overview of device-to-device communication in cellular networks. *ICT Express* **2018**, *4*, 203–208. [CrossRef]
8. 3GPP, 3rd Generation Partnership Project (3GPP). Available online: http://www.3gpp.org/ (accessed on 11 April 2021).
9. Saranya, A.; Rajasekaran, K.; Chandra, S. An Overview of Mobile Ad Hoc Networks. *Int. J. Comput. Sci. Inf. Technol. Res.* **2016**, *4*, 245–250.
10. Rakshith, G.; Mahesh, R. New Technology for Machine to Machine Communication in Softnet Towards 5G. *Int. J. Wirel. Mobile Netw.* **2017**, *9*, 1–23.
11. Jones, R. *The International Telecommunication Union*; IEEE: New York, NY, USA, 2019; p. 1248.
12. Hennebert, C.; Baldini, G.; Peirce, T.; Botterman, M.; Talacchini, M.; Guimaraes Pereira, A.; Handte, M.; Rotondi, D.; Pohls, H. IoT Governance, Privacy and Security Issues. Available online: https://www.researchgate.net/publication/275540220_IoT_Governance_Privacy_and_Security_Issues (accessed on 3 September 2021).
13. Weiwen, Z.; Yonggang, W.; Dapeng, W. Collaborative Task Execution in Mobile Cloud Computing Under a Stochastic Wireless Channel. *IEEE Trans. Wirel. Commun.* **2015**, *14*, 81–93. [CrossRef]
14. Wang, C.M.; Hong, S.F.; Wang, S.T.; Chen, H.C. A dual-mode exerciser for a collaborative computing environment. In Proceedings of the 11th Asia-Pacific, Software Conference, IEEE Computer Society, Busan, Korea, 30 November–3 December 2004; pp. 240–248. [CrossRef]
15. Radicati, S. Mobile Statistics Report: 2014–2018, The Radicati Group, INC. Available online: https://www.radicati.com/?page_id=54 (accessed on 3 February 2014).
16. Available online: https://www.statista.com/statistics/274774/forecast-of-mobile-phone-users-worldwide/ (accessed on 3 September 2021).
17. Rose, C.; Eldridge, S.; Chapin, L. The Internet of Things: An Overview-Understanding the Issues and Challenges of a More Connected World. *Internet Soc.* **2015**. Available online: https://www.internetsociety.org/wp-content/uploads/2017/08/ISOC-IoT-Overview-20151221-en.pdf (accessed on 3 September 2021).
18. Fadlallah, G.; Mcheick, H.; Rebaine, D. Layered Architectural Model for Collaborative Computing in Peripheral Autonomous Networks of Mobile Devices. In Proceedings of the 16th International Conference on Mobile Systems and Pervasive Computing (MobiSPC), Halifax, NS, Canada, 19–21 August 2019.
19. Abdelaziz, J. Un Cadre Architectural Pour la Collaboration dans L'internet Des Objets. Uneapproche Basée sur L'informatique en Brouillard. Ph.D. Thesis, Université du Québec à Chicoutimi, Chicoutimi, QC, Canada, 2018.
20. Steen, M.V.; Tanenbaum, A.S. *Distributed Systems*, 3rd ed.; Pearson Education: London, UK, 2017.
21. Lora Technology. Available online: https://canada.newark.com/wireless-lora-technology (accessed on 3 September 2021).
22. Elhoseny, M.; Hassanien, A.E. Expand Mobile WSN Coverage in Harsh Environments. In *Dynamic Wireless Sensor Networks*; Springer: Cham, Switzerland, 2019; pp. 29–52.
23. Gupta, H.P.; Tyagi, P.K.; Singh, M.P. Regular Node Deployment for k -Coverage in m -Connected Wireless Networks. *IEEE Sens. J.* **2015**, *15*, 7126–7134. [CrossRef]
24. Sharma, D.; Gupta, V. Improving coverage and connectivity using harmony search algorithm in wireless sensor network. In Proceedings of the 2017 International Conference on Emerging Trends in Computing and Communication Technologies (ICETCCT), Dehradun, India, 17–18 November 2017; pp. 1–7.
25. Hasson, S.T.; Finjan, A.A.R. A suggested angles-based sensors deployment algorithm to develop the coverages in WSN. In Proceedings of the 2018 2nd International Conference on Inventive Systems and Control (ICISC), Coimbatore, India, 19–20 January 2018; pp. 547–552.
26. Mcheick, H.; Hatoum, M.S.B.; Ghaddar, A. CMHWN: Coverage Maximization of Heterogeneous Wireless Network. In Proceedings of the 16th ACS/IEEE International Conference on Computer Systems and Applications AICCSA'19, Abu Dhabi, United Arab Emirates, 3–7 November 2019.
27. Ghaddar, A.; Hattoum, M.A.; Fadlallah, G.; Mcheick, H. MCCM: An Approach for Connectivity and Coverage Maximization. *Future Internet* **2020**, *12*, 19. [CrossRef]

28. Wang, Z.; Li, F.; Wang, X.; Li, T.; Hong, T. A WiFi-Direct Based Local Communication System. In Proceedings of the 2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS), Banff, AB, Canada, 4–6 June 2018; pp. 1–6. [CrossRef]
29. Ferrer, A.J.; Marques, J.M.; Jorba, J. Towards the decentralized cloud: Survey on approaches and challenges for mobile, ad hoc, and edge computing. *ACM Comput. Survey* **2019**, *51*, 1–36. [CrossRef]
30. Fadlallah, G.; Mcheick, H.; Rebaine, D.; Adda, M. Towards Mobile Collaborative Autonomous Networks Using Peer-to-Peer Communication. In Proceedings of the 7th International Conference on Software Engineering and New Technologies (ICSENT 2018), Hammamet, Tunisia, 26–28 December 2018; ACM: New York, NY, USA, 2018. ISBN 978-1-4503-6101-9. [CrossRef]
31. Ahmed, A.; Ahmed, E. A Survey on Mobile Edge Computing. In Proceedings of the 2016 10th International Conference on Intelligent Systems and Control (ISCO), Coimbatore, India, 7–8 January 2016; pp. 1–8. [CrossRef]
32. Jararweh, Y.; Doulat, A.; Darabseh, A.; Alsmirat, M.; Al-Ayyoub, M.; Benkhelifa, E. Sdmec: Software defined system for mobile edge computing. In Proceedings of the 2016 IEEE International Conference on Cloud Engineering Workshop (IC2EW), 4–8 April 2016; pp. 88–93.
33. Beck, M.T.; Feld, S.; Fichtner, A.; Linnhoff-Popien, C.; Schimper, T. Me-volte: Network functions for energy-efficient video transcoding at the mobile edge. In Proceedings of the 2015 18th International Conference on Intelligence in Next Generation Networks, Paris, France, 17–19 February 2015; pp. 38–44. [CrossRef]
34. Satyanarayanan, M.; Bahl, P.; Caceres, R.; Davies, N. The case for vm-based cloudlets in mobile computing. *Pervasive Comput.* **2009**, *8*, 14–23. [CrossRef]
35. Fesehaye, D.; Gao, Y.; Nahrstedt, K.; Wang, G. Impact of cloudlets on interactive mobile cloud applications. In Proceedings of the IEEE 16th International Enterprise Distributed Object Computing Conference (EDOC), Beijing, China, 10–14 September 2012; pp. 123–132. [CrossRef]
36. Masson, A.M.G. MoCCA: A Mobile Cellular Cloud Architecture. *J. Cyber Secur. Mobil.* **2013**, *2*, 105–125. [CrossRef]
37. Miluzzo, E.; Cáceres, R.; Chen, Y.F. Vision: mClouds Computing on Clouds of Mobile Devices. In Proceedings of the Third ACM Workshop on Mobile Cloud Computing and Services MCS, Lake District, UK, 25 June 2012. [CrossRef]
38. Hammam, A.; Senbel, S. A reputation trust management system for ad hoc mobile clouds. *Intell. Syst. Ref. Libr.* **2014**, *70*, 519–539.
39. Mehanna, A.A.; Abdel-Fattah, M.A.; Abdel-Gaber, S.M. Cloudlet: A Mobile Cloudlet Model Using Wi-Fi Direct. *Int. J. Comput. Sci. Inf. Secur.* **2016**, *14*, 972.
40. Wi-Fi Alliance, P2P Task Group. Wi-Fi Peer-to-Peer (P2P) Technical Specification; Version 1.5. Available online: https://cse.iitkgp.ac.in/~{}bivasm/sp_notes/wifi_direct_2.pdf (accessed on 3 September 2021).
41. Funai, C.; Tapparello, C.; Heinzelman, W. Enabling Multi-hop Ad Hoc Networks Through Wi-Fi Direct Multi-group Networking. In Proceedings of the International Conference on Computing, Networking and Communications, Santa Clara, CA, USA, 26–29 January 2017; pp. 491–497.
42. Casetti, C.; Chiasserini, C.F.; Duan, Y.; Giaccone, P.; Manriquez, A.P. Data Connectivity and Smart Group Formation in Wi-Fi Direct Multi-group Networks. *IEEE Trans. Netw. Serv. Manag.* **2018**, *15*, 245–259. [CrossRef]
43. Yaqoob, I.; Ahmed, E.; Gani, A.; Mokhtar, S.; Imran, M. Mobile ad hoc cloud: A survey. *Wirel. Commun. Mob. Comput.* **2016**, *16*, 2572–2589. [CrossRef]
44. Balasubramanian, V.; Karmouch, A. An Infrastructure as a Service for Mobile Ad hoc Cloud. In Proceedings of the IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 9–11 January 2017.
45. Yaqoob, I.; Ahmed, E.; Gani, A.; Mokhtar, S.; Imran, M. Heterogeneity Aware Task Allocation in Mobile Ad Hoc Cloud. *IEEE Access* **2017**, *5*, 1779–1795. [CrossRef]
46. Kirby, G.; Dearle, A.; Macdonald, A.; Fernandes, A. An Approach to Ad hoc Cloud Computing. *arXiv* **2010**, arXiv:1002.4738.
47. Shiraz, M.; Gani, A.; Khokhar, R.H.; Buyya, R. A Review on Distributed Application Processing Frameworks in Smart Mobile Devices for Mobile Cloud Computing. *IEEE Commun. Surv. Tutor.* **2013**, *15*, 1294–1313. [CrossRef]
48. Hoang, D.T.; Lee, C.; Niyato, D.; Wang, P. A survey of mobile cloud computing: Architecture, applications, and approaches. *Wirel. Commun. Mob. Comput.* **2013**, *13*, 1587–1611.
49. Sherin, B.C.; Anita, E.A.M. A Survey of Scheduling Algorithms for Wireless Ad-hoc Networks. *Int. J. Adv. Sci. Eng.* **2018**, *4*, 776–787. Available online: www.mahendrapublications.com (accessed on 3 September 2021). [CrossRef]
50. Pervasive and Mobile Computing, Science Direct Journal Publisher: Elsevier. Available online: https://www.researchgate.net/journal/Pervasive-and-Mobile-Computing-1574-1192 (accessed on 3 September 2021).
51. Rappaport, T.S.; Xing, Y.; MacCartney, G.R., Jr.; Molisch, A.F.; Mellios, E.; Zhang, J. Overview of Millimeter Wave Communications for Fifth-Generation (5G) Wireless Networks-with a focus on Propagation Models. *IEEE Trans. Antennas Propag.* **2017**, *65*, 6213–6230. [CrossRef]
52. Oide, T.; Abe, T.; Suganuma, T. Infrastructure-Less Communication Platform for Off-The-Shelf Android Smartphones. *Sensors* **2018**, *18*, 776. [CrossRef] [PubMed]
53. Pycom Enterprise, Description and Technical Details of Pycom's Lopy4. Available online: https://static6.arrow.com/aropdfconversion/c500cfa0a971d49ac7d694a083bb5f20650e211a/pgurl_6269436244227100.pdf (accessed on 12 April 2021).
54. Fadlallah, G.; Rebaine, D.; Mcheick, H. A Greedy Scheduling Approach for Peripheral Mobile Intelligent Systems. *IoT* **2021**, *2*, 14. [CrossRef]
55. Le Tri a Bulle. Available online: https://rmdiscala.developpez.com/cours/LesChapitres.html/Cours4/TBchap4.6.htm (accessed on 3 September 2021).

56. Fadlallah, G.; Mcheick, H.; Rebaine, D. Internet-of-Things (IoT) Shortest Path Algorithms and Communication Case Studies for Maintaining Connectivity in Harsh Environments. In Proceedings of the 2020 International Symposium on Networks, Computers and Communications (ISNCC2020), Montreal, QC, Canada, 20–22 October 2020; pp. 1–8.
57. Viavi Enterprise, Technologie 5G. Available online: https://www.viavisolutions.com/fr-fr/technologie-5g (accessed on 3 September 2021).