Wireless Communications and Mobile Computing

IoT Approaches for Distributed Computing

Lead Guest Editor: Javier Prieto
Guest Editors: Abbes Amira, Javier Bajo, Santiago Mazuelas, and Fernando de la Prieta

IoT Approaches for Distributed Computing

Wireless Communications and Mobile Computing

IoT Approaches for Distributed Computing

Special Issue Editor in Chief: Javier Prieto
Guest Editors: Abbes Amira, Javier Bajo, Santiago Mazuelas, and Fernando de la Prieta

Editorial Board

Contents

Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 9741053, 2 pages
https://doi.org/10.1155/2018/9741053

Editorial

IoT Approaches for Distributed Computing

Javier Prieto,[1,2] **Abbes Amira**,[3] **Javier Bajo**,[4] **Santiago Mazuelas**,[5] **and Fernando De la Prieta**[1]

[1] *BISITE Research Group, University of Salamanca, Edificio I+D+i, C/Espejo, 37007 Salamanca, Spain*
[2] *StageMotion, R&D Department, C/Orfebres 10, 34005 Palencia, Spain*
[3] *College of Engineering, Building B09, Qatar University, PO. Box 2713, Doha, Qatar*
[4] *Artificial Intelligence Department, Polytechnic University of Madrid, 28660 Madrid, Spain*
[5] *BCAM-Basque Center for Applied Mathematics, 48009 Bilbao, Spain*

Correspondence should be addressed to Javier Prieto; javierp@usal.es

Received 4 February 2018; Accepted 8 February 2018; Published 8 March 2018

21.000 million devices will be connected to the Internet by 2021, and 16.000 of them will be part of the Internet of Things (IoT). The usage of manifold connected sensors (temperature, humidity, pressure, vibration, air quality, etc.) in different fields (plants, animals, geological phenomena, cities, homes, etc.) will enable the collection of a vast amount of data subsequently transformed into information and knowledge. However, such a knowledge creation process cannot be handled in a totally centralized way and must be combined with distributed computing so that information transmitted is reduced by sharing the processing load among devices. In traditional distributed computing, shared processing is enabled by additional hardware architectures that have to satisfy higher processing capabilities while ensuring lower power consumption.

The distinct characteristics of IoT technologies require a more intricate trade-off communication versus computation. In particular, a large number of sensors and QoS strict requirements demand new distributed techniques. As the sensor volume grows, infrastructures for IoT distributed computing must include nodes close to the edge that facilitate data analysis for a cluster of sensors. They must also perform edge analytics to reduce the data sent to the core from high-frequency readings and decrease the bandwidth needed. Finally, they must guarantee that customer experience is not compromised, which requires new robust techniques with strict QoS and latency requirements. The emerging paradigm of fog computing enables us to meet these requirements by moving storage and compute services to the network edge or even to the end devices (e.g., to a data hub or to a smart access point).

This special issue aims to be a compendium of the latest development on IoT related to new abstraction or multiagent approaches to distribute tasks among edges and Cloud; new techniques and communication standards for sharing information to increase spectrum efficiency while keeping data consistency and availability; and new metadata, policies, and hardware/software capabilities to aid fog-orchestration in distributed databases.

The paper "Distributed Measurement Data Gathering about Moving Objects" presents techniques for the acquisition of data related to moving objects that reduces the resources consumed by communication tasks. The methods proposed use Fog computing and automated prediction and result in improved network traffic. These methods can enable efficient Internet of Things composed of moving vehicles with strict communication requirements.

The paper "*MeReg*: Managing Energy-SLA Tradeoff for Green Mobile Cloud Computing" proposes an adaptive heuristics energy-aware algorithm, which creates an upper CPU utilization threshold using recent CPU utilization history to detect overloaded hosts and dynamic VM selection algorithms to consolidate the VMs from overloaded or underloaded host. The algorithm tries to minimize total energy consumption and maximize Quality of Service, including the reduction of service level agreement (SLA) violations. The proposed solution contributes to reduce electrical energy consumption, which affects businesses using mobile

cloud computing (MCC) as well as the environment through car-bon dioxide (CO2) emissions.

The paper "Distributed Image Compression Architecture over Wireless Multimedia Sensor Networks" describes techniques that improve the energy consumption for networks that obtain image signals. Specifically, the paper proposes techniques for distributed compression of images, optimal camera coverage design, and routing schemes for reduced transmission energy. The techniques proposed are of particular interest for emerging multimedia sensor networks since both the transmission of original multimedia signals and centralized compression require unaffordable energy consumption.

The paper "An Adaptive Joint Sparsity Recovery for Compressive Sensing Based EEG System" proposes a scheme to reduce the energy consumption associated with the transmission of data in IoT devices such as a wearable electroencephalogram (EEG). This scheme is based on Compressive Sensing (CS) EEG signal compression and recovery. The scheme exploits the joint sparsity of multichannel EEG signals and improves the reconstruction quality and efficiency of the system.

The paper "Using Emotions in Intelligent Virtual Environments: The EJaCalIVE Framework" proposes a framework for the creation of emotional virtual environments that incorporate agents, eHealth related devices, human actors, and emotions projecting them virtually and managing the interaction between all the elements. This framework allows the design and programming of intelligent virtual environments, as well as the simulation and detection of human emotions. The framework is also validated in a case study that simulates a residence for the elderly which enable the training of an assistance robot.

Acknowledgments

We would like to thank all the reviewers who have participated in reviewing the articles submitted to this special issue.

Javier Prieto
Abbes Amira
Javier Bajo
Santiago Mazuelas
Fernando De la Prieta

Hindawi
Wireless Communications and Mobile Computing
Volume 2017, Article ID 8780560, 13 pages
https://doi.org/10.1155/2017/8780560

Research Article

Distributed Measurement Data Gathering about Moving Objects

Ivan Kholod, Nikolay Plokhoy, and Andrey Shorov

Faculty of Computer Science and Technology, Saint Petersburg Electrotechnical University "LETI", Professora Popova Str. 5, Saint Petersburg, Russia

Correspondence should be addressed to Ivan Kholod; iiholod@mail.ru

Received 28 July 2017; Revised 3 November 2017; Accepted 23 November 2017; Published 19 December 2017

Academic Editor: Santiago Mazuelas

This paper describes approaches to gathering measurement data about moving objects in networks with low bandwidth. The first approach uses Fog computing conception and suggests moving assessing the quality of the measurement data into measuring points. The second approach uses prediction of telemetry quality by mining models. In addition, the paper presents implementation of these approaches based on actor model. As a result, it became possible not only to load balancing among edge and cloud nodes, but also to significantly reduce the network traffic, which in turn brings the possibility of decreasing the requirements for communication channels bandwidth and of using wireless networks for gathering measurement data about moving objects.

1. Introduction

The task of measurement data gathering from moving objects (MOs) is a very relevant one. The following examples are given to illustrate this:

(i) In the transportation industry, telemetry provides meaningful information about driver's performance by collecting data from a vehicle, leading to higher fuel consumption efficiency through driver's feedback, which includes in-cab coaching.

(ii) Space science measurements are used by manned or unmanned spacecraft for data transmission.

(iii) Rocketry measurement equipment forms an integral part of the rocket range assets used to monitor the position and health of a vehicle launch.

(iv) Today nearly every type of aircraft, missiles, or spacecraft carries a wireless telemetry system as it is tested. Aeronautical mobile telemetry is used for the safety of pilots and persons on the ground during flight tests. Telemetry from an on-board flight test instrumentation system is the primary source of real-time measurement and status information transmitted during the testing of manned and unmanned aircraft.

(v) Motor racing measurement is a key factor in modern motor racing, allowing race engineers to interpret data collected during a test or a race and use it to properly tune a car for optimum performance.

This task has become even more relevant with development of the unmanned vehicles industry. Measurements (speed, fuel availability, temperature, etc.) transferred from the vehicle to the control point using telemetry channels are used for unmanned vehicle management.

Measurement data gathering about MO involves the use of measuring points (MPs) located along the route of the movement of the MO. Each of them measures various parameters of the objects' performance. Among them we can distinguish the following kinematic parameters: coordinates (x, y, z) and the velocity vector (vx, vy, vz); parameters transferred through the telemetry channel: temperature, vibration, fuel level, and so on. Measurements can be obtained by different measuring systems (MSs): radar, lidar, optics, telemetry, navigation, and so on.

All measurement data that have been received from MSs are continuously transferred to the measurements processing center (MPC) (Figure 1). At the MPC, the measurement data are combined into a main stream of measurements. This stream can be used to identify potential hazardous situations with MO and make decisions to manage them. In addition, tuning of MS can be performed based on the results of

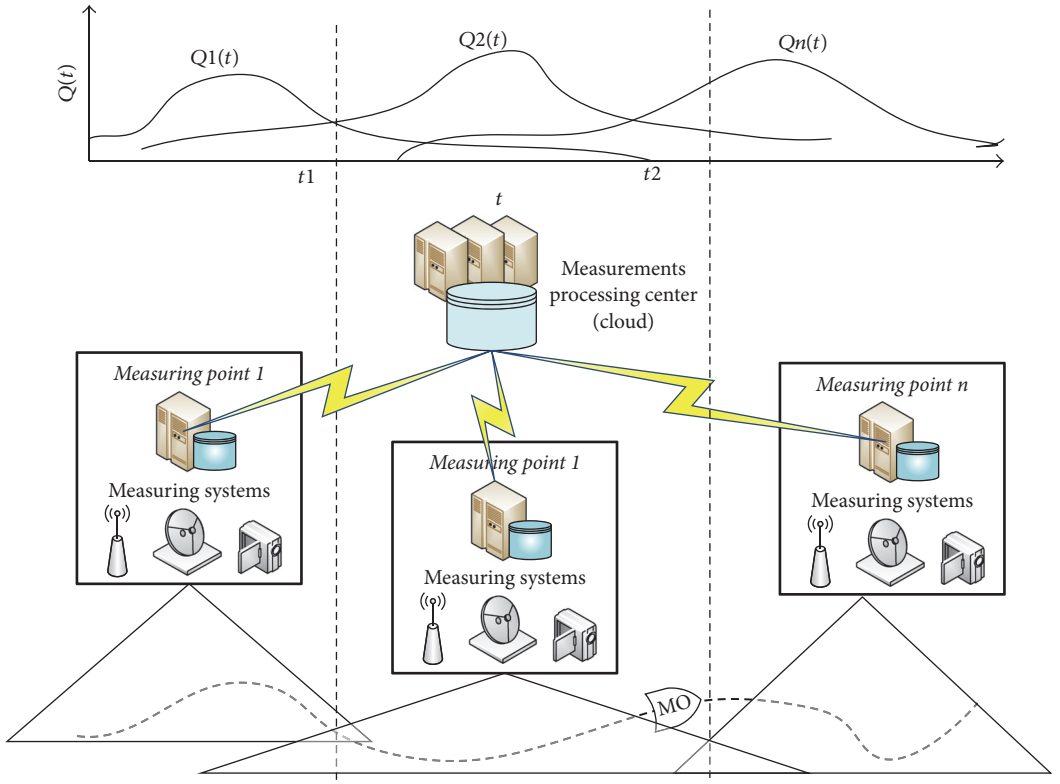


Figure 1: Gathering of measurement data from moving objects.

measurement processing (e.g., aiming of radar antennas and camera lenses) using current and predicted parameters of MO (its location, speed, etc.). A MPC can be created based on cloud computing technology.

Typically, during measurements, all MSs work in the measurement mode, regardless of the presence of the object in their detection zones. It is necessary that measurements of an object are taken as soon as it enters the MS detection zone. However the detection zones are different for each MP and they may have significant intersections (Figure 1). As a consequence, the MPC receives duplicate data of different quality (Q) from all MPs throughout the period of measurement. Thus, the collected data have a significant percent of duplicated measurements and noise.

Thus the following problems of measurement data gathering and processing arise during measurement:

(i) Geographical isolation of MPs from each other and from the MPC

(ii) Duplication of data received from different MPs, as their detection zones intersect

(iii) Need to control both MSs and MO itself, requiring real-time analysis of the collected data

(iv) Using wireless networks with limited bandwidth (satellite, radio relay, cellular, etc.).

More recently, wireless networks, including cell phone networks, wireless local area networks (WLANs), wireless sensor networks, satellite communication networks, and terrestrial microwave networks, have increasingly been used for data gathering, including telemetry [1]. This is explained by convenience of their use: no need for cable installation, ease of connection, and so on. However, one of the principal disadvantages of networks of this type is limited bandwidth.

This paper proposes to use dynamic management of measurement data streams in order to decrease network traffic by transferring only necessary measurements. It allows reducing the volume of transferred data and decreasing bandwidth requirements for communication channels.

2. Related Work

The problem of remote data (telemetry) gathering is not new. First data-transmission circuits go live between the Russian Tsar's Winter Palace and army headquarters in 1845. At the end of the last century and at the beginning of this century, the concept of Machine to Machine communications (M2M) became widespread. M2M technology continues where telemetry left off: wireless telemetry systems today cost substantially less compared to what they did 10 to 15 years ago. Applications that were previously not economically feasible are now cost effective. M2M brings discipline to telemetry through open standards and protocols.

At the present time, we can observe rapid growth in the number of the Internet of Things (IoT) devices. The basic idea of IoT is to connect all things (devices) in the world to the Internet. According to Gartner, Inc. (a technology research and advisory corporation), there will be nearly 26 billion devices on the Internet of Things by 2020 [2].

The basic architecture of IoT which is widely used to explain the approaches of IoT has three layers [3]:

(i) The perception layer is the bottom layer which can be regarded as the hardware or physical layer which does the data collection.

(ii) The network layer (the middle layer) is responsible for connecting the perception layer and the application layer so that data can be passed between them.

(iii) The application layer usually plays the role of providing services or applications that integrate or analyze the data received from the other two layers.

Some researchers [4] propose to extend the layers. They add new layers:

(i) Access gateway layer takes care of message routing, publishing, and subscribing and also performs cross platform communication, if required.

(ii) Middleware layer acts as an interface between the hardware layer at the bottom and the application layer at the top and is responsible for critical functions such as device management and information management and also takes care of issues like data filtering, data aggregation, semantic analysis, access control, and information discovery.

The data processing and analysis services for IoT are deployed on the top layers and divide them into the following sublayers [5, 6]:

(i) Data gathering layer manages collected data: object identification, data abstraction, compression, parsing, and merging; various data are saved in the corresponding database or data warehouse.

(ii) Data processing layer executes data preprocessing and filtering of observed events and allows us to aggregate, organize, and analyze data according to events.

Usually, for implementation of the application and middleware layers, cloud computing technologies are used [6]. A cloud provides scalable storage, computation time, and other tools to build application services. It can be also used to process telemetry gathering from objects.

Similar architecture is also applied for measurement data gathering about moving objects. For example, space situational awareness (SSA) program in European space agency (ESA) and the Egyptian space program use cloud computing for telemetry processing [7]. It enables software and hardware decoupling and makes flexible telemetry data analysis possible. The large amount of available computational resources facilitates a shift in approaches to software development, deployment, and operations. This approach was also used in the Cloud-Based Ground System for space telemetry processing [8].

In this case, the network layer is responsible for connecting the IoT devices (such as sensors, radar, cameras, and other measuring systems) and a cloud. It creates very large network traffic. Solution of this problem can be Fog computing [9]. The fog extends the cloud to be closer to the sources that produce and act on IoT data.

Fog computing either completely solves or decreases the influence of common problems in distributed systems:

(i) Big delays in the network

(ii) Scaling data sources

(iii) Problems related to endpoints mobility

(iv) High cost of the broadband

(v) Wide geographical spread of the systems.

Despite Fog computing becoming popular, there are no ready solutions for its implementation. This can be explained by the fact that such concept is very young and has a high level of abstraction. The paper describes approaches to measurements data gathering based on Fog computing technology. These approaches suggest moving part of computations closer to MPs. It allows redistributing the computational load and reducing network traffic that it is very important to use wireless networks for measurements data gathering.

3. Generic Approach to Telemetry Gathering

3.1. Formal Representation of a Measurement Data Gathering System from Moving Object. The entire measurements data gathering system can be represented formally as a set:

$$\mathrm{GS} = \langle \mathrm{mo}, \mathrm{mpc}, \mathrm{MP} \rangle , \tag{1}$$

where

(i) mo is a controlled moving object;

(ii) mpc is a MPC; MP is a set of MPs.

The state of any MO is characterized by a set of parameters:

$$\begin{aligned} S_{\mathrm{mo}} &= S^{k}{}_{\mathrm{mo}} \cup S^{t}{}_{\mathrm{mo}} \\ &= \{x, y, z, vx, vy, vz\} \cup \{p_1, p_2, \ldots, p_k, \ldots, p_u\} , \end{aligned} \tag{2}$$

where

(i) $S^{k}{}_{\mathrm{mo}}$ are kinematic parameters: coordinates (x, y, z) and the velocity vector (vx, vy, vz);

(ii) $S^{t}{}_{\mathrm{mo}}$ are parameters p_k, $k = 1 \cdots u$, transferred through the telemetry channel: temperature, vibration, fuel level, etc.

The MP is a set of MPs:

$$\mathrm{MP} = \{\mathrm{mp}_1, \mathrm{mp}_2, \ldots, \mathrm{mp}_r, \ldots, \mathrm{mp}_m\} . \tag{3}$$

Each MP mp_r includes MSs that measure kinematic parameters of MO and MSs receiving telemetry data from MO. Therefore, a vector of MO parameters is formed at each measuring point mp_r at each time instant t_j:

$$\begin{aligned} s_r(t) &= s^{k}{}_r(t) \cup s^{t}{}_r(t) \\ &= \{x_r(t), y_r(t), z_r(t), vx_r(t), vy_r(t), vz_r(t)\} \\ &\quad \cup \{p_{r.1}(t), \ldots, p_{r.k}(t), \ldots, p_{r.u}(t)\} , \end{aligned} \tag{4}$$

where

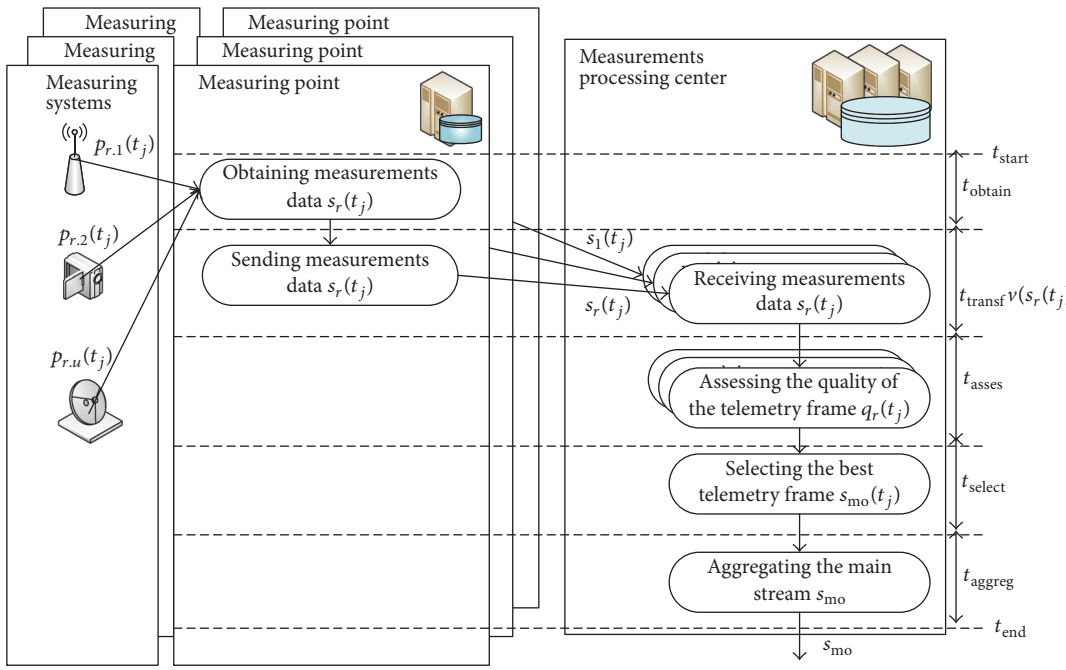


Figure 2: Diagram of the approach with full measurement data gathering (FG).

(i) $s^k{}_r(t_j)$ is a vector of kinematic parameters obtained at time t_j at the point mp_r;

(ii) $s^t{}_r(t_j)$ is a telemetry frame containing a set of parameters of the object MO obtained at time t_j by telemetry at the point mp_r.

For each telemetry frame $s^t{}_r(t_j)$, it is possible to measure its quality $q_r(t_j)$, characterizing the integrity and reliability of the received data. The quality of telemetry frame can be estimated in many ways and is not considered in this paper (verification of built-in markers (e.g., Reed-Solomon codes) for each frame can serve as an example of estimation of telemetry data quality).

The main purpose of a measuring system is to produce a main stream containing the best measurements of all parameters of a MO at each time instant of the measurement:

$$s_{mo} = \left\{ s_{mo}(t_1), s_{mo}(t_2), \ldots, s_{mo}(t_j), \ldots, s_{mo}(t_w) \right\}, \quad (5)$$

where $s_{mo}(t_j)$ is the set of the best values of MO parameters at time t_j.

Thus, the generic approach with full measurement data gathering (FG) involves the following procedure performed at each time instant t_j (Figure 2):

(1) Obtaining measurements data $s_r(t_j)$ on each MP mp_r $r = 1 \cdots m$

(2) Transferring measurements data $s_r(t_j)$ from each MP mp_r $r = 1 \cdots m$ into the MPC

(3) Assessing the quality of the telemetry frame $q_r(t_j)$ received from each MP

(4) Selecting the best telemetry frame $s_{mo}(t_j)$

(5) Aggregating the main stream s_{mo}.

The runtime of a data transfer operation is determined (based on Hockney's model [10]) by the volume of the data v, the latency α, and the bandwidth β of a communication channel:

$$t_{transf}(d) = \alpha + \frac{v(d)}{\beta}, \quad (6)$$

where $v(d)$ is volume of data d.

The total time for the formation of the main stream T_{FG} (from t_{start} to t_{end}) at each instant time t_j can be calculated using the following formula:

$$T_{FG} = \max_{r=1}^{m}(t_{obtain}) + \max_{r=1}^{m}\left(t_{transf}\left(s_r\left(t_j\right)\right)\right) + \sum_{r=1}^{m} t_{asses} + t_{select} + t_{aggregate}. \quad (7)$$

Network traffic for a data-transmission operation is determined by the amount of transferred data. Therefore, network traffic V_{FG} at each instant time t_j is determined by the following formula:

$$V_{FG}\left(t_j\right) = \sum_{r=1}^{m} v\left(s_r\left(t_j\right)\right). \quad (8)$$

The quality of the best telemetry frame $Q(t_j)$ at each instant time t_j is determined by the maximum quality among frames gathering from all MPs:

$$Q_{FG}\left(t_j\right) = \max_{r=1}^{m}\left(q_r\left(t_j\right)\right). \quad (9)$$

The advantage of the FG approach is the highest possible quality of telemetry because it gathers telemetry from all MPs and selects the best frames from them. The disadvantages of this approach are as follows:

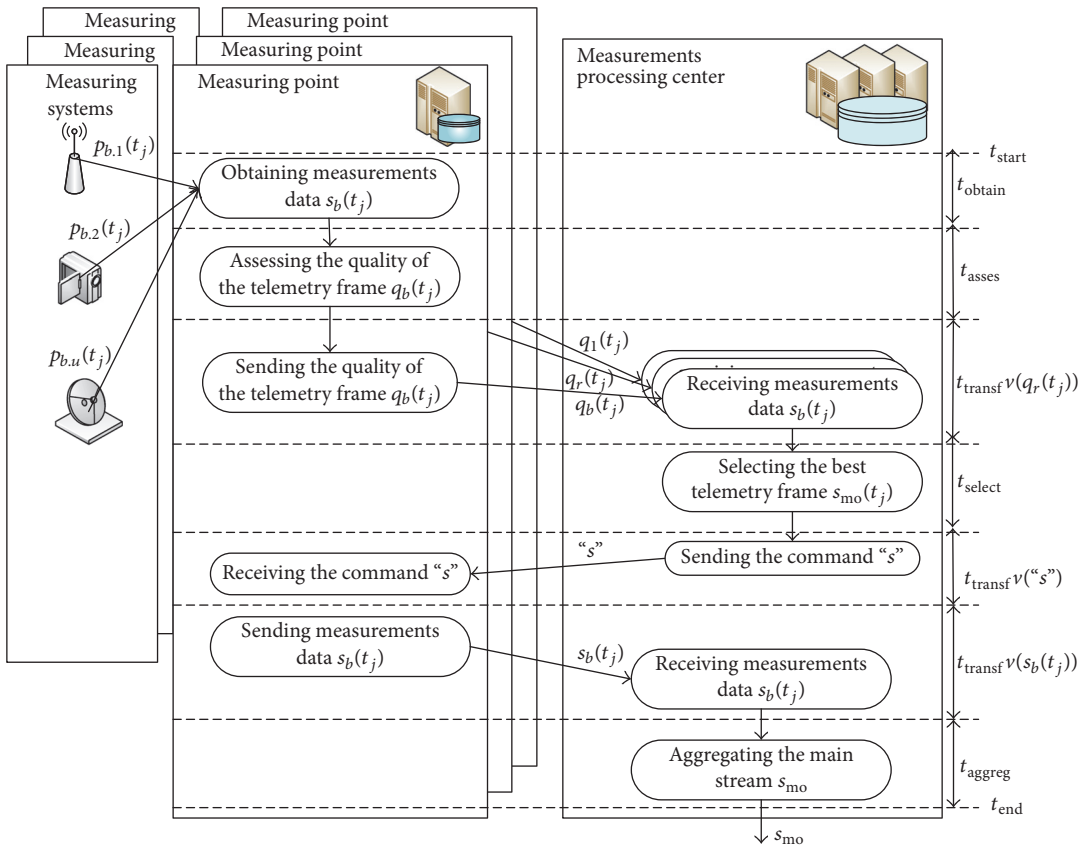


Figure 3: Diagram of the approach with a distributed quality assessment (DQA).

(i) High network traffic V_{FG} between MP and the MPC due to the need to transfer all measurements from MP, even those that will not eventually be included in the main measurement stream

(ii) A large number of calculations in the MPC associated with the quality evaluation of the incoming streams from each $mp_r \in MP$, selection of the best stream, and their aggregation into the main stream.

The situation becomes even more difficult if there are several MOs and measurement streams multiply depending on the number of MOs.

3.2. Gathering with a Distributed Quality Assessment. In order to reduce network traffic during measurements data gathering, it is proposed to transfer only the telemetry frame that will eventually be included into the main stream. To achieve this, the telemetry frame quality must be assessed at the MP and transferred to the MPC. The MPC collects assessments of quality and selects the best one. The best telemetry frame is requested from the corresponding MP and sent to the MPC to be included into the main stream.

Thus, the following procedure is performed at each time instant t_j (Figure 3):

(1) Obtaining measurements data $s_r(t_j)$ on each MP mp_r $r = 1 \cdots m$

(2) Assessing the quality of the telemetry frame $q_r(t_j)$ on each MP mp_r $r = 1 \cdots m$

(3) Transferring the quality of the telemetry frame $q_r(t_j)$ from each MP mp_r $r = 1 \cdots m$ into the MPC

(4) Selecting the best telemetry frame

(5) Transferring the command "s" to the selected MP mp_b to transfer the telemetry frame

(6) Transferring the best telemetry frame $s_b(t_j)$ from MP mp_b to the MPC

(7) Aggregating the main stream s_{mo}.

The total time for the formation of the main stream T_{DQA} at each instant time t_j can be calculated using the following formula:

$$\begin{aligned} T_{DQA} = {} & \max_{r=1}^{m}\left(t_{obtain}\right) + \max_{r=1}^{m}\left(t_{asses}\right) \\ & + \max_{r=1}^{m}\left(t_{transf}\left(\nu\left(q_r\left(t_j\right)\right)\right)\right) + t_{select} \\ & + t_{transf}\left(\nu\left("s"\right)\right) + t_{transf}\left(\nu\left(s_r\left(t_j\right)\right)\right) \\ & + t_{aggregate}. \end{aligned} \tag{10}$$

Network traffic for data-transmission operations is determined by the amount of data transferred. Therefore, network

traffic V_{DQA} at each instant time t_j is determined by the following formula:

$$V_{\mathrm{DQA}}\left(t_j\right) = \sum_{r=1}^{m} v\left(q_r\left(t_j\right)\right) + v\left(\text{“}s\text{”}\right) + v\left(s_r\left(t_j\right)\right). \quad (11)$$

The quality of the best telemetry frame $Q(t_j)$ at each instant time t_j is also determined by the maximum quality of the snapshots collected from all measuring points:

$$Q_{\mathrm{DQA}}\left(t_j\right) = \max_{r=1}^{m}\left(q\left(s^t_r\left(t_j\right)\right)\right). \quad (12)$$

This approach makes it possible to decrease network traffic significantly, because

$$\sum_{r=1}^{m} v\left(q_r\left(t_j\right)\right) + v\left(\text{“}s\text{”}\right) + v\left(s_r\left(t_j\right)\right) \ll \sum_{r=1}^{m} v\left(s_r\left(t_j\right)\right). \quad (13)$$

In addition, calculations required to assess the quality of telemetry frame are performed on each MP, which significantly decreased the load at the MPC.

The disadvantage of approach with a distributed quality assessment (DQA) is a possible increase in the main stream formation time after the moment of measurement, because additional functions are executed:

$$\begin{aligned} &\max_{r=1}^{m}\left(t_{\mathrm{transf}}\left(s_r\left(t_j\right)\right)\right) + \sum_{r=1}^{m} t_{\mathrm{asses}} \\ &\quad < \max_{r=1}^{m}\left(t_{\mathrm{asses}}\right) + \max_{r=1}^{m}\left(t_{\mathrm{transf}}\left(v\left(q^t_r\left(t_j\right)\right)\right)\right) \\ &\quad + t_{\mathrm{transf}}\left(v\left(\text{“}s\text{”}\right)\right) + t_{\mathrm{transf}}\left(v\left(s_b\left(t_j\right)\right)\right). \end{aligned} \quad (14)$$

3.3. Gathering with Quality Assessment Prediction. In order to reduce the main stream formation time, it is suggested to predict quality of telemetry frame at each time instant $q(s^t{}_r(t_j))$. By predicting the change in the quality of the telemetry, it is possible to preemptively determine the time instants when it is necessary to transfer a frame to be included in the main stream.

The main problem is to calculate $q(s^t{}_r(t_{j+n}))$ for a period of time n sufficient for MS switching. Such prediction can be achieved by applying mining models, obtained through the use of data mining algorithms, including those used for time series analysis (logical regression, etc.).

In this case, quality value $q(t_{j+n})$ will be approximated for the following period. However, to improve the accuracy of the prediction, not only the existing values, but also additional attributes that affect the quality of the telemetry can be used. In this case, it is possible to produce the following vector that would characterize the quality of telemetry frame $s^t{}_r(t_j)$ at time instant t_j from MP mp_r:

$$\begin{aligned} x_r\left(t_j\right) = \{&x\left(t_j\right), y\left(t_j\right), z\left(t_j\right), vx\left(t_j\right), vy\left(t_j\right), vz\left(t_j\right), \\ &q\left(s^t{}_r\left(t_j\right)\right), q\left(s^t{}_r\left(t_{j+n}\right)\right)\}. \end{aligned} \quad (15)$$

Thus training set is the time-ordered sequence of all vectors from the start time instant of measurement t_0 till the finish t_w on each MP mp_r:

$$\begin{aligned} X_r = \{&x_r\left(t_0\right), x_r\left(t_1\right), \ldots, x_r\left(t_j\right), \ldots, x_r\left(t_{j+n}\right), \ldots, \\ &x_r\left(t_w\right)\}. \end{aligned} \quad (16)$$

A mining model can be constructed by data mining algorithms using such vector sets obtained from previous measurements of this MP's objects. In this case, the quality assessment $q(s^t{}_r(t_{j+n}))$ is known for each $x_r(t_j)$. In addition, the mining model can be corrected in the course of measurements, using the measurements that have already been performed.

Using approach with a quality assessment prediction (QAP), the formation of the main stream $s_{\mathrm{mo}}(t)$ at each time instant t_j is performed concurrently on each MP mp_r and at the MPC (Figure 4):

(i) Previously, the following actions are performed on each MP mp_r $r = 1 \cdots m$:

  (1) Obtaining measurements data $s_r(t_j)$ on each MP mp_r $r = 1 \cdots m$
  (2) Assessing the quality of the telemetry frame $q_r(t_{j-1})$ on each MP mp_r $r = 1 \cdots m$
  (3) Sending the quality assessment $q_r(t_{j-1})$ and kinematic parameters $s^k{}_r(t_{j-1})$ from each MP mp_r $r = 1 \cdots m$ into the MPC.

(ii) Previously, the following actions are performed at MPC:

  (1) Receiving the quality assessment $q_r(t_{j-1})$ and kinematic parameters $s^k{}_r(t_{j-1})$ from each MP mp_r $r = 1 \cdots m$
  (2) Predicting the quality of the telemetry frame $q'{}_r(t_j)$ for each MP mp_r $r = 1 \cdots m$ at time t_j
  (3) Selecting the best MP mp_b at time t_j
  (4) Sending “s” command for transferring the telemetry frame to the best MP mp_b.

(iii) The following actions are performed on selected MP mp_b after receiving “s” command for transferring the telemetry frame:

  (1) Obtaining measurements data $s_r(t_j)$ on each MP mp_r $r = 1 \cdots m$
  (2) Sending the telemetry frame $s_b(t_j)$ from the best MP mp_b to the MPC (only for the best MP mp_b at time instance t_j).

(iv) The following actions are performed at MPC:

  (1) Receiving the telemetry frame $s_b(t_{j-1})$ from the best MP mp_b
  (2) Aggregating the main stream s_{mo}.

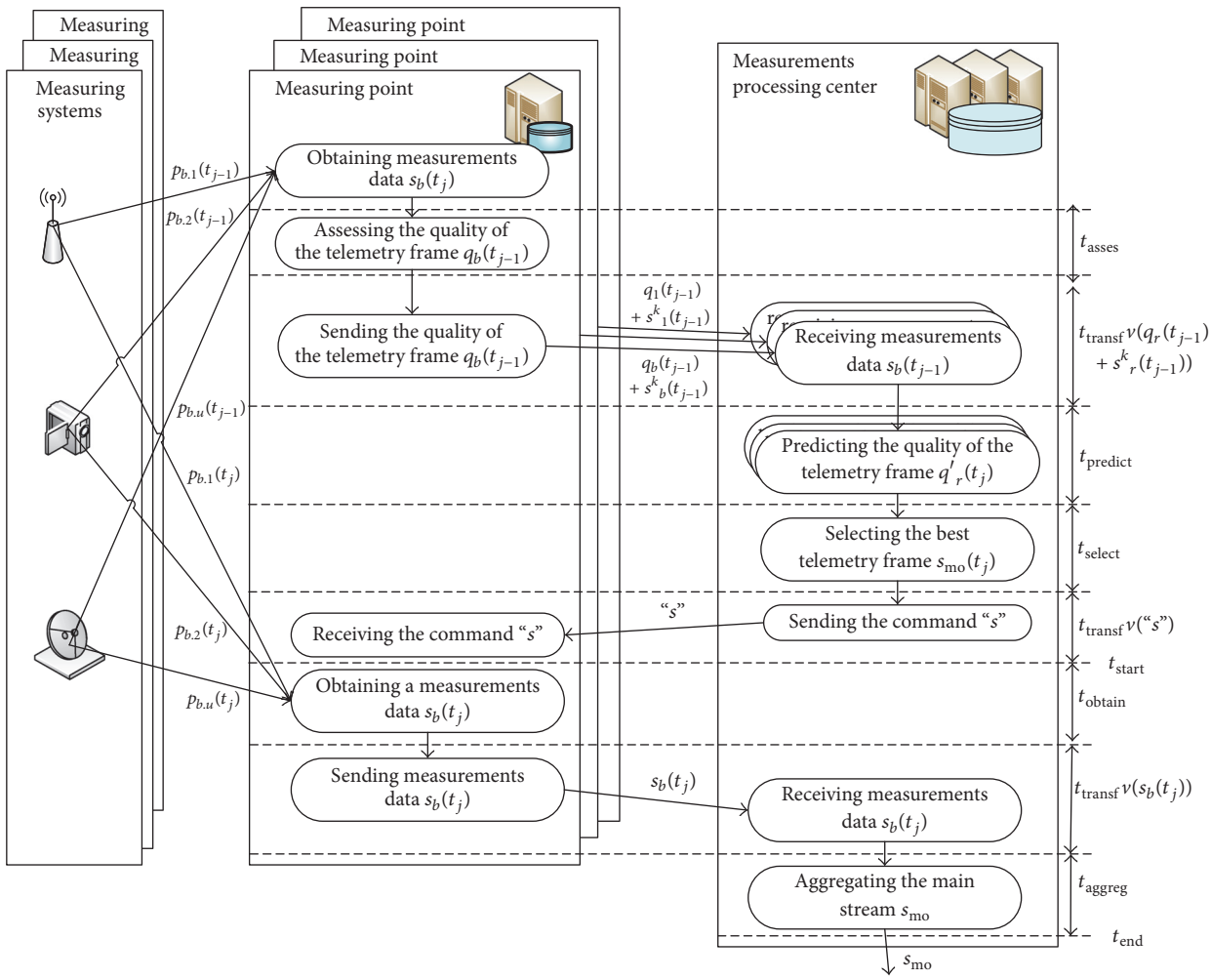


Figure 4: Diagram of the approach with a quality assessment prediction (QAP).

The total time for the formation of the main stream in approach QAP T_{QAP} at each instant time t_j can be calculated using the following formula:

$$T_{\mathrm{DQA}} = \max_{r=1}^{m}\left(t_{\mathrm{obtain}}\right) + t_{\mathrm{transf}}\left(v\left(\text{“}s\text{”}\right)\right) + t_{\mathrm{aggregate}}. \tag{17}$$

Time of full cycle for each measurement is calculated by the following formula:

$$\begin{aligned} T'_{\mathrm{DQA}} = {} & T_{\mathrm{DQA}} + \max_{r=1}^{m}\left(t_{\mathrm{asses}}\right) \\ & + \max_{r=1}^{m}\left(t_{\mathrm{transf}}\left(v\left(q_r^t\left(t_j\right)\right) + v\left(s_r^k\left(t_j\right)\right)\right)\right) \\ & + \sum_{r=1}^{m} t_{\mathrm{predict}} + t_{\mathrm{select}}. \end{aligned} \tag{18}$$

In this case, network traffic V_{DQA} at each instant time t_j is determined by the following formula:

$$\begin{aligned} V_{\mathrm{DQA}}\left(t_j\right) = {} & \sum_{r=1}^{m}\left(v\left(q_r\left(t_j\right)\right) + v\left(s_r^k\left(t_j\right)\right)\right) + v\left(\text{“}s\text{”}\right) \\ & + v\left(s_b\left(t_j\right)\right). \end{aligned} \tag{19}$$

The quality of the telemetry frame $q_r(t_j)$ at each time point t_j is determined by prediction accuracy of the built mining model (k_m):

$$Q_{\mathrm{QAP}}\left(t_j\right) = \max_{r=1}^{m}\left(q\left(s_r^t\left(t_j\right)\right)\right) \times k_m. \tag{20}$$

Due to the fact that the volume of kinematic parameters $s^k{}_r(t_j)$ is generally much smaller than the volume of telemetry frame $s^t{}_r(t_j)$

$$v\left(s^k{}_r\left(t_j\right)\right) \ll v\left(s^k{}_r\left(t_j\right)\right), \tag{21}$$

this approach results in much lower network traffic than the generic approach and slightly higher network traffic than the previous one:

$$V_{\mathrm{DQA}} < V_{\mathrm{QAP}} \ll V_{\mathrm{FG}}. \tag{22}$$

Also, calculations on assessment of the quality of telemetry frame are performed on each MP, which considerably relieves the MPC from the computational load.

Thus the approach QAP has low network traffic and the time of formation of the main stream. The disadvantage of

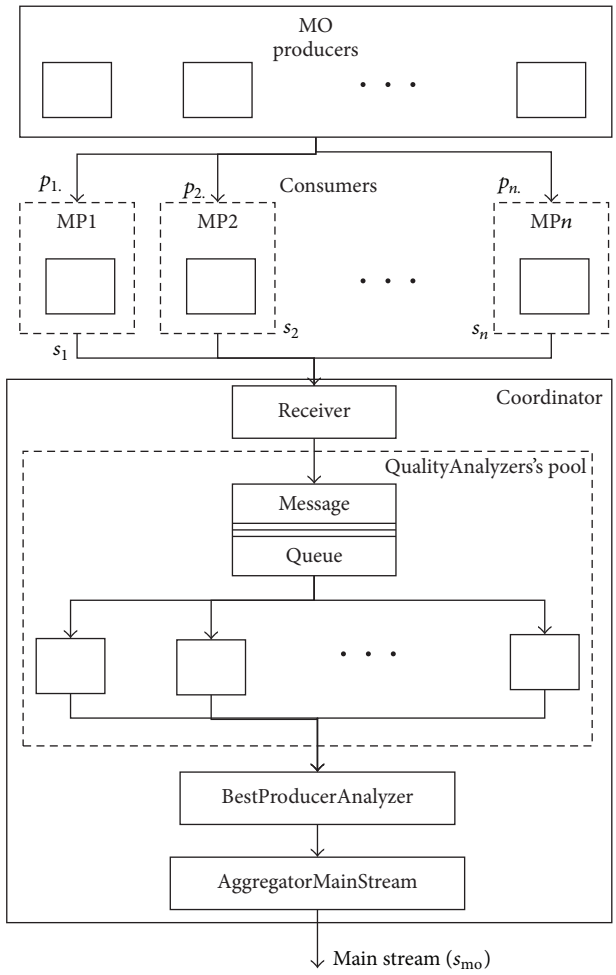


Figure 5: Implementation of generic approach (FG).

this approach is a potential decrease in quality of main stream. The extent of the decrease depends on the quality of the developed predictive model. However, it should be noted that even an approach with full measurement gathering fails to achieve 100% quality for all telemetry frames, as there may be time instants when all MPs receive frames with low quality. In view of this, telemetry analysis systems should provide the option of processing frames of poor quality. Therefore, the telemetry quality criterion can be noncritical.

4. Implementation of the Proposed Approaches

4.1. Elements of the Measurement Data Gathering System. In order to implement the measurement data gathering system from distributed sources, we used actors model [11] and its software implementation, Akka library [12].

The overall principle of the abovementioned mechanism using complete data transfer involving actors model is shown in Figure 5. The main entities are presented in Figure 5:

(i) Producers (data sources) are various measurement means mounted on MO and generating the telemetric stream; each producer generates data packet with 0.25 sec interval and transfers it to the primal consumer.

(ii) Consumers (primary consumers) are MS mounted on MP and gathering measurements coming from the sources. They form the measurement frame and transfer them to MPC for further processing.

(iii) Coordinator (calculation center) is a set of actors hidden behind general façade and representing functionality implemented by MPC and, more specifically, preprocessing, frames quality analysis, and forming the main stream.

(iv) Receiver is an actor presenting secondary consumer, in particular calculation center (CC). The actor is a point of interaction of the CC with all Consumers actors. After receiving another data packet, actor transfers it for processing.

(v) QualityAnalyzers are set of actors united into pool and located in CC. They assess the quality of incoming packets. Each data packet is placed to the processing queue, upon which vacant analyzer addresses this queue and retrieves the packet for processing. Analyzed data packets are further transferred to the best producer analyzer.

(vi) BestProducerAnalyzer is an actor determining the best snapshot among ones received from primary consumers (PCs). Analysis is performed with the packet receiving interval (0.25 sec), but there is an initial delay provided (3 sec delay was used during testing). Upon determining the best snapshot, actor transfers it to the main stream aggregator.

(vii) AggregatorMainStream is an actor which "glues" the best snapshots into main stream. Formed stream is transferred to the end consumers. The way of implementing the main stream translation process depends on particular implementation.

4.2. Implementation of Gathering with a Distributed Quality Assessment. In order to implement the approach DQA, several modifications were made in the system affecting the actors model.

The QualityAnalyzers pool becomes unnecessary since the stream quality assessment is performed in MP, and there should be one actor of this type per each unit. Therefore, the functionality provided by these actors can be integrated to the Consumer-type factors. Thus, each unit shall have its own queue for snapshots received from Producers.

It is unnecessary to implement the BestProducerAnalyzer actor as an individual actor anymore, as its functionality can be combined with Receiver-type actor.

Other types of actors remain unchanged. The actors used in this implementation of the data gathering mechanism are shown in Figure 6.

4.3. Implementation of Gathering with Quality Assessment Prediction. In order to implement the approach with quality assessment prediction of measurements, implementation of

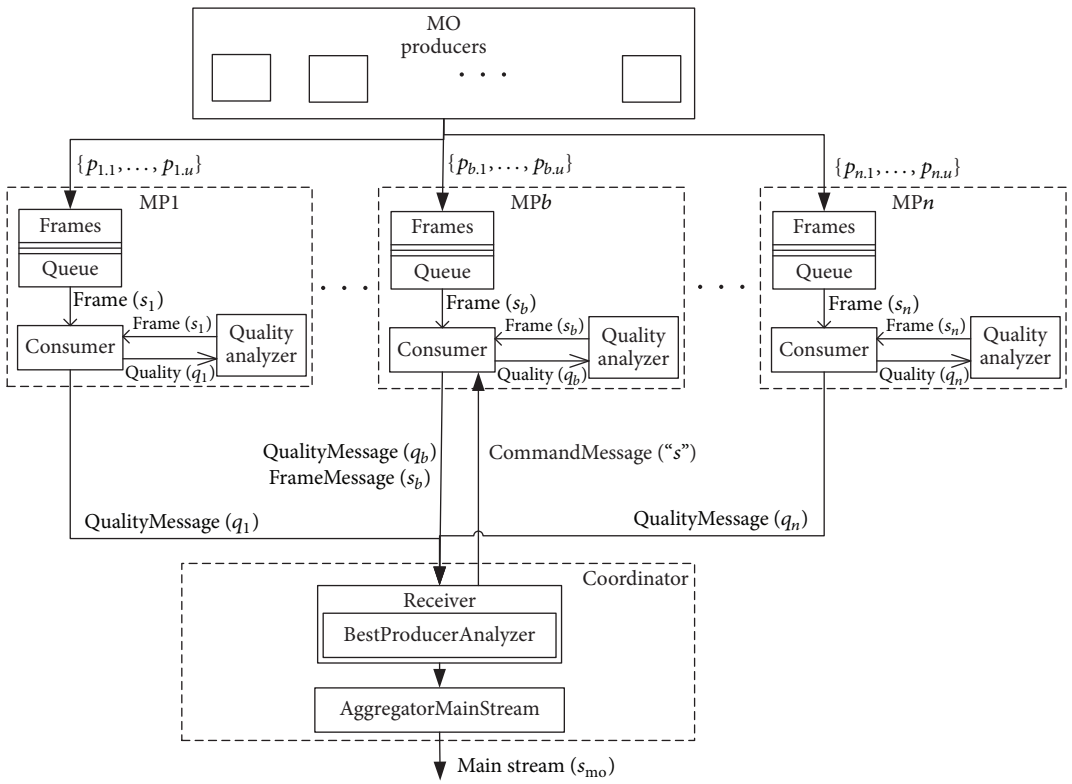


Figure 6: Implementation of gathering with a distributed quality assessment (DQA).

the Receiver actor was modified. It uses mining model to predict the best MP at next time instance. The mining model used for prediction was implemented as PredictionModel class. Corresponding actors are shown in Figure 7.

5. Experiments

Experiments were performed with the distributed system where each of the actors was located on an individual machine. Four physical computers with two virtual machines deployed on each of them were used. Ubuntu Server 16.04 was installed on virtual machines.

For experiments, we used data sets of real telemetry from flight object. The data sets were received from 2 MPs with a measuring duration of 12,5 minutes (2,700 measurements with period 0,25 sec). Each MP contains the following MS:

(i) Radar measuring kinematic parameters: coordinates (x, y, z) and the velocity vector (vx, vy, vz)

(ii) Telemetry.

The first characteristic used to assess different implementations was the frame delay, that is, the time interval from the moment of obtaining measurements data to the moment of aggregating the main stream. Developed system allows setting the initial frame delay. This is a time interval where the start time of all system components and expected time delays in message transfer via network are set. Main stream aggregator located in CC starts the stream formation process after the specified initial delay from the start of the test. During the test, initial delay was set as one iteration (0.25 s).

From the retrieved data, it follows that the average delay value shall be (Figure 8)

(i) for approach with full data gathering (FG), about 0.2629 sec,

(ii) for approach with distributed quality assessment (DQA), 0.2921 sec,

(iii) for approach with quality assessment prediction (QAP), 0.2645 sec.

Additional delay of the distributed implementation option made by the system (except 0.25 sec of initial delay) is 0.0421 sec which is three times larger compared to the similar indication of the first implementation (0,0129 sec). Differences result from the measurements data delivery mechanisms in these two implementations. In the DQA implementation, additional delay is made at the phase of transferring the quality report to the CC and sending request for best-quality snapshot to a certain PC.

Delay in the QAP implementation is close to the result received from the generic approach. The value is increased insignificantly due to necessity to switch between data sources when swapping the best MP.

The next researched characteristic is quality of the main stream. Since the testing was performed with preformed data sets, it was possible to perform preassessment of their quality. For each data frame, quality characteristic was calculated

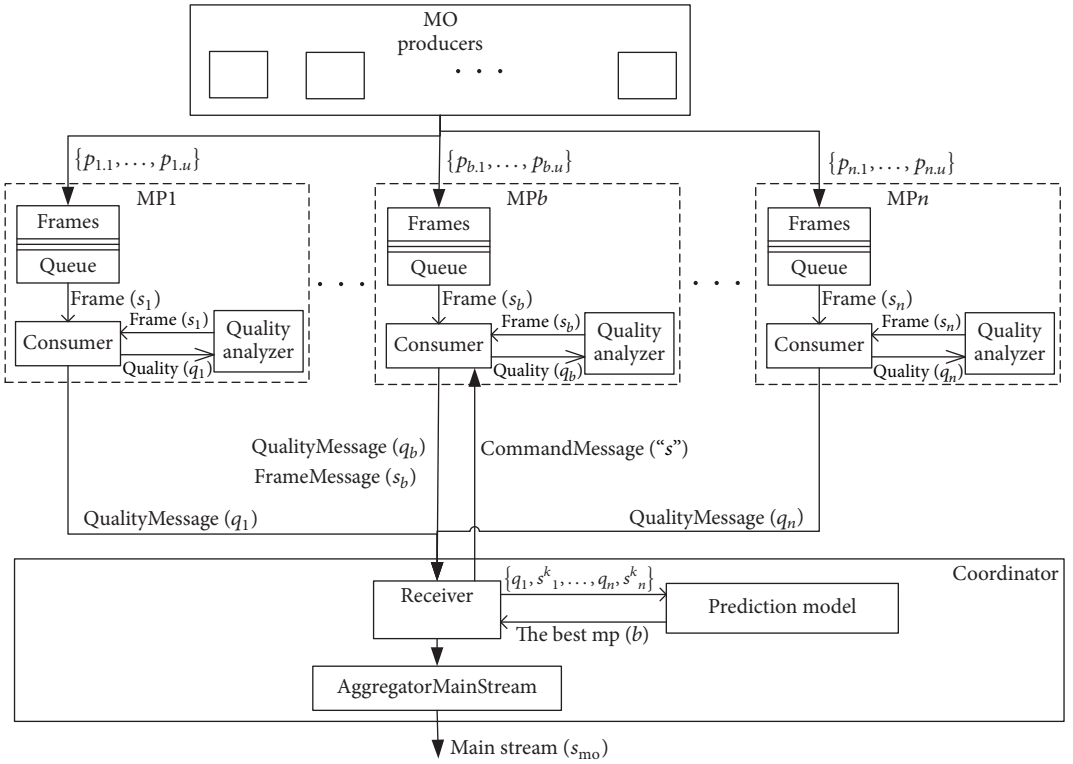


Figure 7: Implementation of gathering with quality assessment prediction (QAP).

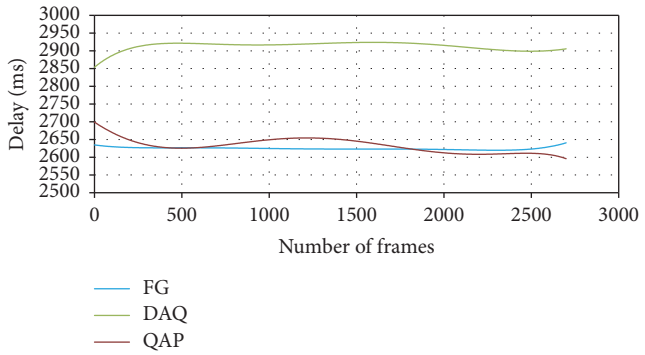


Figure 8: Summary diagram of the frame delay in three different implementations.

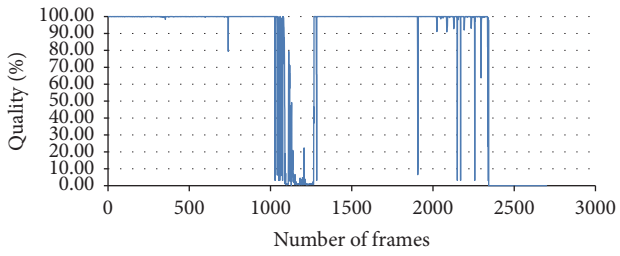


Figure 9: Diagram of the frame quality assessment from the first MP.

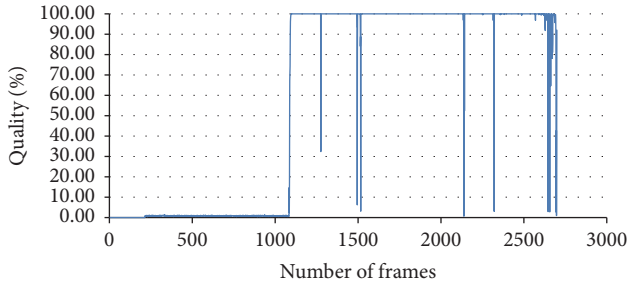


Figure 10: Diagram of the frame quality assessment from the second MP.

(Figure 9 for telemetry from first MP and Figure 10 for telemetry from second MP). Figure 11 shows that the quality of main stream (green line) is produced by the approaches FG and DAG.

Results received from the QAP implementation present greater interest. To predict quality, several mining models were used. They were trained by data mining algorithms from the Weka library [13]:

(i) GaussianProcesses [14]

(ii) MultilayerPerceptron [15]

(iii) RegressionByDiscretization [16]

(iv) Decision tree (M5PBase) [17]

(v) SVM regression (SMOreg) [18].

The following metrics were used to assess the accuracy of the prediction model (where y is a true value, y' is a value

Table 1: Comparison of prediction models.

Prediction model	Correlation coefficient	MAE	RMSE	RAE	RRSE
GaussianProcesses	0,3977	15,9667	25,3496	100,90%	91,71%
MultilayerPerceptron	0,9562	4,034	8,2296	25,49%	29,77%
RegressionByDiscretization	0,9922	0,3714	3,4396	2,35%	12,44%
SMOreg	0,4246	8,8983	28,9597	56,23%	104,77%
M5P	0,989	1,5178	4,4831	9,59%	16,22%

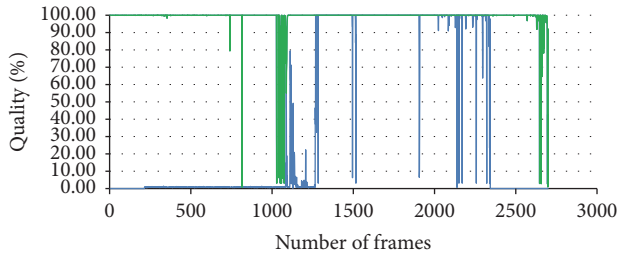


Figure 11: Diagram of the main stream (green color) quality assessment for the FG and DAG implementation.

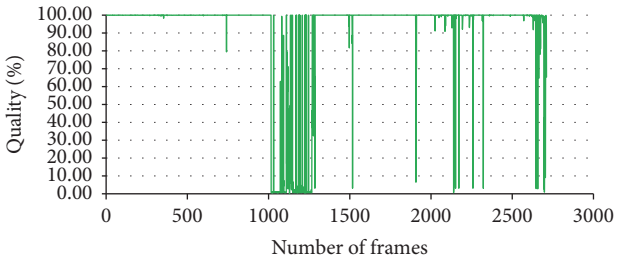


Figure 12: Diagram of the main stream assessment using RegressionPredictionModel.

estimated using some algorithm, and $\overline{y}$ is a mean value of y):

(i) Mean absolute error (MAE):

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n} \left| y'_i - y_i \right|}{n} \tag{23}$$

(ii) Root mean squared error (RMSE):

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{n} \left(y'_i - y_i \right)^2}{n}} \tag{24}$$

(iii) Relative absolute error (RAE):

$$\mathrm{RAE} = \frac{\sum_{i=1}^{n} \left| y'_i - y_i \right|}{\sum_{i=1}^{n} \left| \overline{y}_i - y_i \right|} \tag{25}$$

(iv) Root relative squared error (RRSE):

$$\mathrm{RRSE} = \sqrt{\frac{\sum_{i=1}^{n} \left(y'_i - y_i \right)^2}{\sum_{i=1}^{n} \left(\overline{y}_i - y_i \right)^2}}. \tag{26}$$

The results of the comparison are presented in Table 1.

The prediction models MP5 and RegressionByDiscretization have the best metrics. They were selected for the implemented approach of gathering with quality assessment prediction.

As a result of using the MP5 model, the results were close to the one presented in Figure 11. The correlation coefficient of these two series was 0.9844, which conforms with theoretical assessment presented in Table 1. The data received from RegressionByDiscretization testing are slightly different from

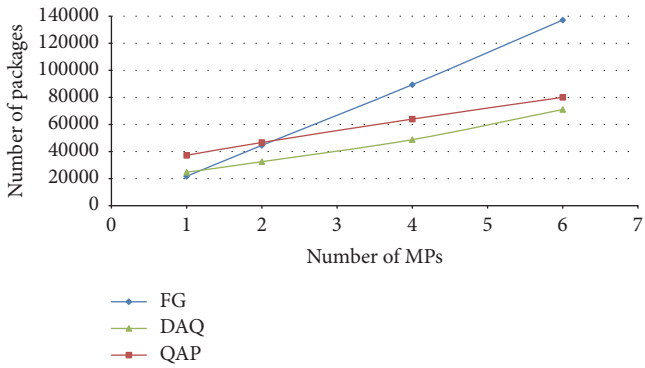


Figure 13: Summary diagram of packet quantity dependence from quantity of PCs.

theoretically predicted ones (Figure 12). In this case, the correlation coefficient was 0,7841.

As it can be noticed, the model fails in the areas where the best sources swap. This may indicate incorrectly calculated dependence between parameters for prediction model.

Another criterion for implementation comparison is network traffic volume analysis. Measurements were made with Wireshark utility [19], a traffic analyzing software for Ethernet network.

Traffic measurement was made with different system configurations, in particular with varying PC quantity used in test emulations. In this kind of test, quality of the formed main stream was not taken into account. Thus, for each implementation variant, 4 system configurations were prepared: with 1, 2, 4, and 6 PCs. Test was performed three times for each configuration, and results were averaged and generated traffic volume was assessed based on those values.

Diagrams (Figures 13 and 14) represent results of the implementation testing. The size of one source data file is

Table 2: Experimental results.

Approach	Delay (second)	Mining model correlation	Number of MPs	Network traffic (Mb)	Quality of main stream
Generic approach	0.2629	-	2	686	93%
			4	1,376	
			6	2,077	
Approach with distributed quality assessment	0.2921		2	374	93%
			4	396	
			6	412	
Approach with quality assessment prediction by regression model	0.2645	0,9807	2	375	83%
			4	401	
			6	420	
Approach with quality assessment prediction by M5PBase model		0,9526	2	372	92%
			4	399	
			6	419	

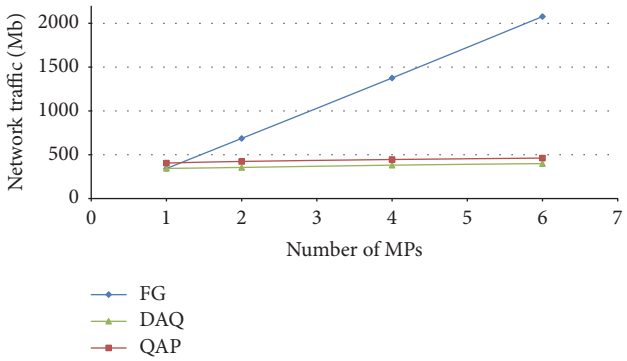


Figure 14: Summary diagram of traffic volume dependence from quantity of PCs.

~339 Mb. It can be seen that the dependence of the traffic volume from quantity of MPs is linear, wherein actual data confirm theoretical assumptions about linear dependence of the traffic volume from source data volume and, at the same time, number of MPs in the system.

Analysis of the above data allows concluding that the implementations of proposed approaches use much greater number of TCP packets for data transfer (37241 compared to 21752 with a single MP). Yet, specific size of these packets is significantly smaller, as the general traffic volume for one used MP is 355.31 Mb compared to 342.715 Mb in the implementation of generic approach. This can be explained by smaller size of the packets formed for transferring. At the same time, traffic volume for implementation of proposed approaches is not directly dependant on quantity of MPs in the system. This significantly reduces total volume with their greater amount (411.98 Mb compared to 2077.23 Mb with 6 active MPs).

Table 2 summarises experimental results for all approaches of measurement data gathering.

The experimental results showed a significant reduction of traffic transmitted over the network by using approaches to distributed measurement data gathering. However, the DAQ approach had a bigger delay to form the main stream. Using prediction model for quality assessment solves this problem. At the same time, the quality of a main stream varies depending on the selected mining model used to predict the stream quality.

6. Conclusion

Use of Fog computing for data gathering from moving objects allows not only reducing the load to data processing center by putting part of calculations on MP, but also significantly reducing the network traffic. Both approaches give significant reduction of the network load. Also, when increasing quantity of MPs in these variants, network traffic does not increase, which allows increasing the number of measurement tools in the system and gaining better control over the objects. This allows using wireless networks with limited capacity for data gathering.

The disadvantage of the distributed quality assessment approach is an increase of time delay in the main stream forming. If this value is crucial, it is possible to use approach with assessment prediction. It gives the same time delay as the implementation with full data gathering.

Quality of the formed main stream with this approach is fully dependent on prediction model implementation. Prediction can be improved also by model learning in the process of measurement values gathering. It is the objective of future research.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This paper has been prepared with the support of the Ministry of Education and Science of the Russian Federation in the framework of the state work "Organization of Scientific Research" of State Task no. 2.6113.2017/ВУ.

References

[1] G. Miao, J. Zander, K. W. Sung, and S. Ben Slimane, *Fundamentals of Mobile Data Networks*, Cambridge University Press, Cambridge, UK, 2016.

[2] Gartner, "Gartner Says the Internet of Things Installed Base Will Grow to 26 Billion Units By 2020," STAMFORD, Conn., 2013, https://www.gartner.com/newsroom/id/2636073.

[3] C.-W. Tsai, C.-F. Lai, and A. V. Vasilakos, "Future internet of things: open issues and challenges," *Wireless Networks*, vol. 20, no. 8, pp. 2201–2217, 2014.

[4] F. Chen, P. Deng, J. Wan, D. Zhang, A. V. Vasilakos, and X. Rong, "Data mining for the internet of things: Literature review and challenges," *International Journal of Distributed Sensor Networks*, vol. 2015, Article ID 431047, 2015.

[5] S. Bin, L. Yuan, and W. Xiaoyi, "Research on data mining models for the internet of things," in *Proceedings of the 2nd International Conference on Image Analysis and Signal Processing (IASP'10)*, pp. 127–132, April 2010.

[6] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswamia, "Rajkumar Buyya, Slaven Marusic, Marimuthu Palaniswamia, Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions," *Future Generation Computer Systems, Elsevier, Volume 29, Issue*, vol. 29, no. 7, pp. 1645–1660, 2013.

[7] A. El-Sharkawi, A. Shouman, and S. Lasheen, "Service Oriented Architecture for Remote Sensing Satellite Telemetry Data Implemented on Cloud Computing," *International Journal of Information Technology and Computer Science*, 2013, Published Online June 2013 in MECS.

[8] C. Mirchandani, "Cloud-based Ground System for Telemetry Processing," in *Proceedings of the Complex Adaptive Systems, 2015*, pp. 183–190, USA, November 2015.

[9] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the 1st ACM Mobile Cloud Computing Workshop, MCC 2012*, pp. 13–15, Finland, August 2012.

[10] R. W. Hockney, "The communication challenge for MPP: Intel Paragon and Meiko CS-2," *Parallel Computing*, vol. 20, no. 3, pp. 389–398, 1994.

[11] C. Hewitt, P. Bishop, and R. Steiger, "A universal modular ACTOR formalism for artificial intelligence," *International Joint Conferences on Artificial Intelligence*, pp. 235–245, 1973.

[12] "Akka Documentation," http://akka.io/docs/.

[13] I. H. Witten, F. Eibe, and A. Mark, *Data Mining: Practical Machine Learning Tools and Techniques*, vol. 629, Morgan Kaufmann Publishers, San Francisco, 3nd edition, 2011.

[14] C. E. Rasmussen, "Gaussian processes for machine learning," in *Advanced Lectures on Machine Learning*, vol. 3176 of *Lecture Notes in Computer Science*, pp. 63–71, The MIT Press, Cambridge, Mass, USA, 2004.

[15] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, Springer series in statistics, Springer, New York, NY, USA, 2008.

[16] D. D. Rucker, B. B. McShane, and K. J. Preacher, "A researcher's guide to regression, discretization, and median splits of continuous variables," *Journal of Consumer Psychology*, vol. 25, no. 4, pp. 666–678, 2015.

[17] J. R. Quinlan, *Learning with continuous classes*, University of Sydney, Australia, 2006.

[18] S. K. Shevade, S. S. Keerthi, C. Bhattacharyya, and K. R. K. Murthy, "Improvements to the SMO algorithm for SVM regression," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 11, no. 5, pp. 1188–1193, 2000.

[19] "Wireshark Developer's Guide," https://www.wireshark.org/docs/wsdg_html_chunked/.

Hindawi
Wireless Communications and Mobile Computing
Volume 2017, Article ID 6741972, 11 pages
https://doi.org/10.1155/2017/6741972

Research Article

MeReg: Managing Energy-SLA Tradeoff for Green Mobile Cloud Computing

Rahul Yadav and Weizhe Zhang

School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China

Correspondence should be addressed to Rahul Yadav; rahul@stu.hit.edu.cn and Weizhe Zhang; wzzhang@hit.edu.cn

Received 1 August 2017; Revised 13 October 2017; Accepted 1 November 2017; Published 17 December 2017

Academic Editor: Javier Bajo

Mobile cloud computing (MCC) provides various cloud computing services to mobile users. The rapid growth of MCC users requires large-scale MCC data centers to provide them with data processing and storage services. The growth of these data centers directly impacts electrical energy consumption, which affects businesses as well as the environment through carbon dioxide (CO_2) emissions. Moreover, large amount of energy is wasted to maintain the servers running during low workload. To reduce the energy consumption of mobile cloud data centers, energy-aware host overload detection algorithm and virtual machines (VMs) selection algorithms for VM consolidation are required during detected host underload and overload. After allocating resources to all VMs, underloaded hosts are required to assume energy-saving mode in order to minimize power consumption. To address this issue, we proposed an adaptive heuristics energy-aware algorithm, which creates an upper CPU utilization threshold using recent CPU utilization history to detect overloaded hosts and dynamic VM selection algorithms to consolidate the VMs from overloaded or underloaded host. The goal is to minimize total energy consumption and maximize Quality of Service, including the reduction of service level agreement (SLA) violations. CloudSim simulator is used to validate the algorithm and simulations are conducted on real workload traces in 10 different days, as provided by PlanetLab.

1. Introduction

Mobile devices, such as smartphones and tablets, are becoming essential to human life as the most effective computational and convenient communication tools are not bounded by time and place. These devices are replacing desktop or laptop computers by using the cloud computing environment or mobile cloud computing (MCC). The MCC is a combined infrastructure of cloud computing and mobile computing in which data processing and storage are performed on the cloud, and mobile devices are mainly used as client to communicate with the application and retrieve processed results from the cloud [1]. The rapid growth of mobile computing usage is evident in the study of Juniper Research, which states that the consumer and enterprise market for cloud-based mobile applications increased to $9.5 billion by 2014 [2], directly impacting cloud infrastructure. Cloud computing is leveraged on existing technologies and ideas, such as data centers and virtualization technology. This new perspective revolutionized traditional information technology (IT) business by helping developers and companies overcome lack of hardware capacity (such as CPU, memory, and storage) by allowing users to access on-demand resources through the Internet [3, 4].

Cloud computing is mainly divided into three types of service models, namely, Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). Moreover, cloud computing has four types of deployment models such as private, public, hybrid, and community clouds [5, 6]. Provision of MCC services to users requires large-scale cloud computing platform, which drains enormous amount of electric power and increases MCC operational costs, CO_2 emissions. Data centers consume approximately 1.3% of the total worldwide electricity supply, which is predicted to increase to 8% by 2020 [7]. Therefore, CO_2 also increase substantially, which directly impacts the environment. Unfortunately, large amounts of electrical energy are wasted by servers during low workload. The server resources

utilization data collected from more than 5000 production servers over a six-month period have shown that most of the time servers operate at 10% to 50% of their full capacity, leading to wasting the energy on low utilization of resources [8].

The Quality of Service (QoS) constraint plays an important role between mobile cloud service providers and users. Meeting QoS requirements is determined via Service Level Agreements (SLAs) that describe the required performance levels, such as minimal throughput and maximal response time or latency of the system. Therefore, the main challenge is to minimize power consumption of mobile cloud data centers while satisfying QoS requirements [9].

Hardware virtualization technology transforms traditional hardware to the new paradigm. This technology consolidates workload, called virtual machine (VM) consolidation, and exploits low-power hardware states. Most current studies have minimized the overall energy consumption through two widely used techniques, such as VM consolidation and dynamic server provisioning [10, 11]. Dynamic server provisioning methods reduce electric power consumption by reducing the computational resources during low workloads [12]. This reduction means turning the unnecessary servers to sleep-mode when the workload demand decreases. Similarly, when data processing and data storage demands increase, these servers are reactivated according to requirements [13, 14]. The server shares its resources among multiple performance-isolated platforms called VMs by using hypervisor technology. Each VM runs more than one task simultaneously. Dynamic VM consolidation also plays an important role in minimizing overall energy consumption in mobile cloud data centers. The VM consolidation occurs when a server (host) detects overload or underload, during which VM migrates one by one from the overloaded host to another appropriate host until the overload returns to its normal state. Similarly, when the host detects underload, all VMs migrate to appropriate hosts and turn this host to sleep-mode [15, 16]. Basically, these approaches have two main objectives: minimizing overall energy consumption and maximizing the QoS. The QoS requirements are formalized via SLA metric and such features are described as minimal throughput and maximal response time or latency delivered by the deployed system [17].

The basic task of efficient energy consumption in mobile cloud data centers is divided into five parts as follows:

(1) Determine when a host is considered overloaded so that some VMs would migrate one by one to other efficient hosts under SLA constraint until the host returns to normal state. To detect overloaded hosts, we used *MeReg* algorithm, which is introduced in this paper.

(2) Determine when a host is considered underloaded so that all VMs would migrate from it to the appropriate hosts and it will turn into sleep-mode. To detect underloaded host, we used constant lower CPU utilization threshold proposed in Beloglazov and Buyya [18].

(3) Select VMs from an overloaded host that should have migrated from it. To select, we used our previous work in Yadav et al. [19].

(4) Select all VMs from an underloaded host that should have migrated from it. To select, we used our previous work in Yadav et al. [19].

(5) Find a new VM allocation where selected VMs from overloaded and underloaded hosts would be placed to activate or reactivate hosts. We used the modified best fit decreasing (MBFD) algorithm proposed in Beloglazov et al. [16] for VM placement.

In this study, we proposed a regression-based adaptive heuristic algorithm for estimating an upper threshold to detect the overloaded hosts of mobile cloud data center. From these hosts, several VMs are migrated to another host to minimize the performance degradation. We used a novel MuMs dynamic VM selection algorithm to balance trade-offs among electric power consumption, number of migrations, performance of host, and total number of hosts that were shut down. These algorithms estimate the upper threshold and selection of VMs based on the statistical analysis of CPU utilization history of hosts. The following are the main contributions of this paper:

(i) An adaptive heuristic *MeReg* algorithm to estimate upper CPU utilization threshold using recent CPU utilization history for detecting overloaded hosts is introduced. This algorithm mainly aims to minimize overall power consumption under the required SLA of mobile cloud data center.

(ii) The performance and effectiveness of the *MeReg* algorithm are evaluated using the CloudSim simulator on real and random workload traces and compared with other proposed approaches in the literature.

The rest of this paper is organized as follows: In Section 2, we discussed some previous literature related to mobile cloud data center resources and energy efficiency management. In Section 3, we presented the mobile cloud platform architecture. Section 4 is a key part of this paper where we discussed host overload detection. In Section 5, we proposed energy efficiency metric for measuring the effectiveness of the proposed algorithms in the cloud environment. In Section 6, the experiment setup for proposed algorithms is discussed. In Section 7 results of the proposed algorithms are analysed and compared, and in Section 8, the study is concluded by a summary with future research direction.

2. Related Work

Researchers have examined the design of mobile cloud models and its associated software architecture [20]. A paradigm shift is evident from traditional to mobile cloud computing which requires large-scale of cloud data center, wherein the cost of computational resources is no longer the major portion of the overall cost. However, the cost of power consumption and cooling infrastructure are still considered primary cost drivers. Power consumption and CPU

utilization in servers or mobile are directly proportional to one another [21, 22]. Therefore, recent techniques for minimizing power consumption and maximizing QoS are discussed in this study. In one of the first works introduced by Zhang et al. [23], dynamic efficient energy techniques for mobile computing that schedule multiple computing tasks are dynamically reconfigured and selectively turned off to minimize overall energy consumption in mobile computing.

Esfandiarpoor et al. proposed a VM consolidation algorithm that efficiently reduces energy in cloud data center by considering structural features, such as racks and network topology. Moreover, they focused on the cooling and network structure of cloud data center hosting the physical machines when consolidating VMs. Few racks and routers are employed without compromising the SLA so that idle routing and cooling equipment could be turned off to reduce energy consumption [24]. Zhu et al. [25] investigated the dynamic VM consolidation problem and applied a static host CPU utilization threshold of 85%, which is determined if the host is overloaded when CPU utilization threshold exceeded 85%. However, static CPU threshold is unsuitable for systems with dynamic workload, as this static model does not adapt to system workload changes. In this study, we introduced a dynamic adapt threshold value according to the statistical analysis of workload history.

Nathuji and Schwan [26] proposed dynamic VM consolidation to minimize the energy consumption of hosts in data centers. They investigated energy management techniques in the large-scale virtualized resources of data center. They proposed a new energy management method for virtualized resources of data center called Soft Resource Scaling. In addition, the authors suggested dividing the resource management problem into two levels: local and global. At the local level, the algorithms handle the energy management of guest VMs. By contrast, global policies coordinate multiple physical machines. They also explored the benefits of efficient energy consumption using live migration and found that total energy consumption can be significantly reduced.

Beloglazov et al. [16] proposed a cloud computing architectural framework and the provision of mobile cloud data center resources in power efficient manner, while meeting SLA requirements. They established two parts of the VM consolidation problem: (1) submission of new requests for VM provisioning and allocation of VMs on hosts; (2) significant use of current VM allocations. To solve the problem of VM placement on hosts, they used the MBFD algorithm. This algorithm first sorts current CPU utilization of all VMs in decreasing order and allocates each VM to a host, which provides efficient energy consumption environment. In another work, Beloglazov and Buyya [18] introduced a heuristic-based energy-aware approach, which focused on the statistical analysis of CPU utilization history to determine an upper threshold for detecting overloaded hosts

Ranganathan et al. [27] described server power management method at the collective systems level instead of the individual server level. This approach permits active servers to borrow power from inactive servers. Similarly, Venkatachalam et al. [28] introduced an efficient energy technique for minimizing the overall energy consumed by the server CPU at a given period. They also focused on GPU electric energy consumption.

The energy consumption of the data centers is broken down in [29, 30]. Most studies have considered energy consumption modeling at the CPU level: however, network devices also consume considerable amount of energy in terms of data center energy consumption. Therefore, load balancing of data center network devices is important to minimize the energy consumption cost. Shang et al. [31, 32] introduced a distributed green-routing algorithm which consider computation, communication, and thermal temperature within the data center. The future decision of the proposed load-balancing algorithm requires a full energy model including networks and servers in the data center. Liu et al. [33] introduced a distributed flow scheduling (DFS) for efficient energy consumption in data center network devices. However, this approach did not consider the nature of communication sources, sinks, and corresponding computation.

3. System Architecture

The general architecture of MCC includes mobile devices, network connection, and cloud computing data center. In Figure 1, mobile devices are directly connected to the base station using the mobile network. The base station establishes and controls the air connection between mobile devices and the network [34] and communicates with the cloud data center via the Internet to complete the task of the mobile users such as data processing and storage. The cloud data center includes numerous virtualized resources to improve performance of the services. These resources consist of n heterogeneous hosts. Wherein each host contains multicore CPU, primary memory, secondary memory, and network I/O. The CPU performance is determined in terms of millions of instruction per second (MIPS). The submission of multiple requests for VM provisioning is allocated to hosts simultaneously. The allocation of VMs to hosts is based on CPU utilization of the host. The energy consumed by the CPU is linearly proportional to its utilization [18]. Therefore, efficient consolidation of VM would reduce the electric energy consumption and the SLA violation rate. When the running VM cannot obtain its resources from the cloud data center such as MIPS and memory, then SLA violation would occur. In this case, a cloud service provider should pay cloud service users penalty, when an overloaded host is confirmed. The next step is selecting VMs for migration from the overloaded host to appropriate host and apply iteratively to the host until it is no longer considered overloaded.

In this MCC model, three main important players handle all workflows within cloud data centers. The key players are global controller, local controller, and virtual machine manager (VMM). A local controller resides in each host as a separate VM and is tasked to monitor the status of the VM, and CPU utilization as well as decide what time VM should be migrated from the host. The global controller resides on a single master host and gathers all information from the local controller to maintain overall resources utilization. Moreover, it decides where VM should be optimally placed. Finally, the VMM resides along the hypervisor and helps in resizing the

Table 1: The electric energy consumed by the considered servers at different level of workload in watts (W).

Server	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Fujitsu M1	13.3	18.3	21.1	23.4	26.5	29.6	34.7	40.7	46.8	57.4	60
Fujitsu M3	12.4	16.7	19.4	21.4	23.4	26.1	29.7	34.8	41	47.1	51.2
Hitachi TS10	37	39.9	43.2	45.5	48.8	52.8	57.8	65.1	73.8	80.8	85.2
Hitachi SS10	36	38.8	41.2	43.7	46.3	49.4	53.1	58.8	64.2	67	69.7

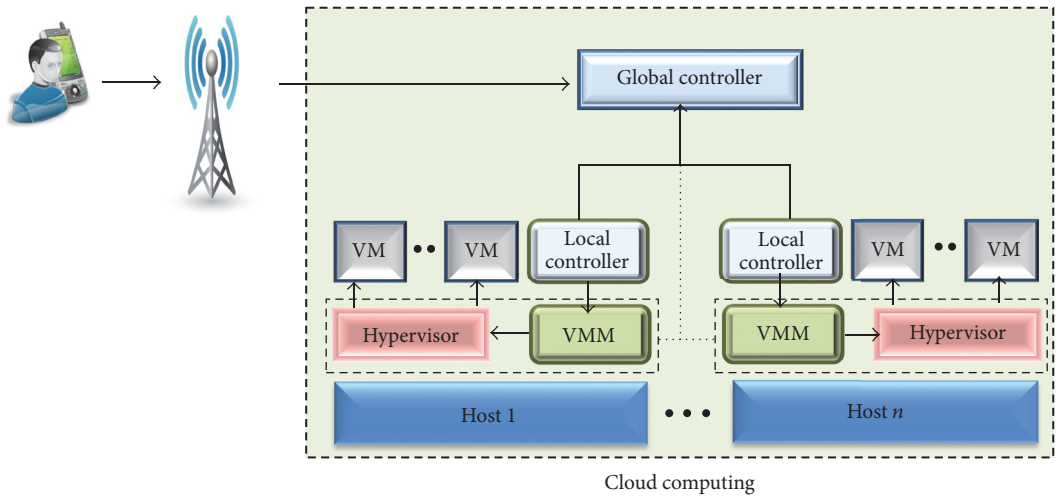


Figure 1: Mobile cloud computing system architecture.

VM and changes the power state of the host, which helps efficiently utilizing energy.

3.1. Energy Model. Relative to other types of equipment, the major energy consumers of mobile cloud data center components are CPU, network, and memory. Recent works show that the electric power consumed by the host's processor is directly proportional to its utilization. Utilization of the processor depends on the workload of the host and changes according to the variability of the workload [35]. Therefore, utilization of the processor is a function of time, and its value changes according to workload variability. The overall electric energy consumption by the host can be defined as an integral function of the power consumed by the host at a given period and is described as follows [16]:

$$E = \int_{t_0}^{t_1} P\left(u\left(t\right)\right) dt, \tag{1}$$

where E is the total electric energy consumed by the server. $P(u(t))$ is the continuous function of workload utilization at time t.

Moreover, we considered four different types of hosts, namely, Fujitsu M1, Fujitsu M3, Hitachi TS10, and Hitachi SS10. The features of these hosts are shown in Table 2. The energy consumption of these servers is obtained from the SPECpower [36]. The electric energy consumption of these hosts at different workloads is shown in Table 1.

Table 2: Characteristics of the hosts.

Server	CPU	Core	Clock speed	Memory
Fujitsu M1	Xeon 1230	4	2.7 GHz	8 GB
Fujitsu M3	Xeon 1230	4	3.5 GHz	8 GB
Hitachi TS10	Xeon 1280	4	3.5 GHz	8 GB
Hitachi SS10	Xeon 1280	4	3.6 GHz	8 GB

4. *MeReg* Host Overloaded Detection

The mobile cloud computing platform has recently become popular worldwide because of its dynamic nature. However, the dynamic characteristics of mobile cloud computing pose a big concern for cloud service provider (CSP). Therefore, the constant CPU utilization threshold is unsuitable for detecting an overloaded host in cloud environments. We proposed a novel algorithm for host overload detection based on a regression model called *M* estimator regression model. This algorithm dynamically estimates the upper CPU utilization threshold based on the historical dataset of CPU utilization, which is automatically adjusted according to the historical CPU workload.

Robust regression techniques provide more efficient optimal solution than traditional approaches. These techniques are not directly influenced by the outlier in the dataset, which makes it more robust and trustworthy for the dynamic environment of the cloud. The "M estimation Regression" *(MeReg)* generates a regression line in which the median

of the squared residuals is minimized [37]. The *MeReg* is a more robust estimator than the median, standard deviation, variance, and ordinary least squares estimators. "Ordinary least squares (OLS) have the following disadvantages: (1) a single corrupt data point can give the resulting regression line an arbitrarily large slope; (2) it can behave badly when the residual distribution is not normal, particularly when the residuals are heavily tailed" [38, 39]. To initialize the *MeReg* algorithm, we first need to generate the OLS model representing the relationship between input data X and the value of the output data Y using line the straight as follows:

$$\begin{aligned} Y_i &= \theta_1 + \theta_2 X_i + \varepsilon_i, \\ \varepsilon_i &= Y_i - (\theta_1 + \theta_2 X_i), \end{aligned} \tag{2}$$

where ε_i is the independent variable called residuals. This model mainly aims to minimize the value of residuals ε_i. If the values of all residuals ε_i converge to the zero, then an optimal model is generated, wherein all given data points lie on this model. $i \in V$, where V is set of all VMs CPU utilization dataset of the data center. The goal is to minimize the sum of distance between the estimated linear parameter and actual CPU utilization data point. The objective function of estimation can be defined as follows:

$$\begin{aligned} \min \mathcal{F}(\widehat{\varepsilon}_i) &= \sum_{i=1}^{m} \frac{(Y_i - (\theta_1 + \theta_2 X_i))}{\sigma}, \\ \sigma &= \frac{\text{median}\left|\varepsilon_i - \text{median}(\varepsilon_i)\right|}{0.6745}, \end{aligned} \tag{3}$$

where σ represents a residuals standard deviation of CPU utilization data point. To make this model more robust, Tukey's bisquare function as an objective function of M estimation is used, where $\widehat{\varepsilon}_i$ is the residual divided by residuals standard deviation, and constant c is called a *tuning constant*. The small value of c produce increases resistance to outliers but at the expense of very low efficiency when the residuals are normally distributed. Therefore, the value of $c = 4.685$ is usually selected to provide 95% efficiency when the residuals are normally distributed [39]. The $\mathcal{U}(\widehat{\varepsilon}_i)$ bisquare objective function is given as follows:

$$\mathcal{U}(\widehat{\varepsilon}_i) = \begin{cases} \dfrac{\widehat{\varepsilon}_i^2}{2} - \dfrac{\widehat{\varepsilon}_i^4}{2c^2} + \dfrac{\widehat{\varepsilon}_i^6}{6c^4}, & \left|\widehat{\varepsilon}_i\right| \le c \\ \dfrac{\widehat{\varepsilon}_i^2}{6} & \left|\widehat{\varepsilon}_i\right| > c. \end{cases} \tag{4}$$

To define the *weight function* of the residuals, we should obtain the partial differentiation of this equation with respect to θ_2. Let ψ be the first derivative function of $\mathcal{F}(\widehat{\varepsilon}_i)$, which define the weight function

$$\begin{aligned} \sum_{i=1}^{m} X_i \psi\left(\frac{(Y_i - (\theta_1 + \theta_2 X_i))}{\sigma}\right) &= 0, \\ w(\widehat{\varepsilon}_i) &= \frac{\psi(\widehat{\varepsilon}_i)}{\widehat{\varepsilon}_i}. \end{aligned} \tag{5}$$

The weight function w of this model also changed according to observations.

$$w_i = \begin{cases} \left(1 - \left(\dfrac{\widehat{\varepsilon}_i}{c}\right)^2\right)^2, & \left|\widehat{\varepsilon}_i\right| \le c \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

To determine the optimal solutions or values of θ_1 and θ_2 by Tukey's bisquare weighted function,

$$\sum_{i=1}^{m} X_i w_i \left(\frac{(Y_i - (\theta_1 + \theta_2 X_i))}{\sigma}\right) = 0. \tag{7}$$

We utilize this approach to fit a trend polynomial model to all observations of the CPU utilization of VMs. In every iteration, weight function is defined according to new residuals that is called iteratively reweighed least squares and is repeated until it converges to the optimal values of θ_1 and θ_2, which determine the minimum value of $\mathcal{U}(\widehat{\varepsilon}_i)$ metric. This minimum value is called *MeReq*, which estimates the upper threshold of CPU utilization.

The detection of the overloaded host is determined by the upper CPU utilization threshold metric used in [18]. We extended this metric through *MeReg* to detect overloaded hosts shown as follows:

$$UpT = 1 - p \times MeReq, \tag{8}$$

where p is the safety parameter of this algorithm, which define how fast the system is in consolidating VMs. Moreover, the small value of safety parameter p implies low energy consumption but high SLA violation and vice versa [18]. The pseudocode of *MeReg* host overloaded detection algorithm, which helps in understanding the full workflow of the algorithm, is discussed in Algorithm 1.

5. Efficiency Metrics

Various matrices are used to evaluate the results and compare the effectiveness of the algorithm. The first metric is called total energy consumed by the data center resources at different workloads. The second type of efficiency metric is the average percentage of the SLA violation, which only occurs when provision VMs are not obtaining the requested resources (or when the average computing power of the shared host is not allocated to the requested VMs). This metric directly influence the QoS, which is not negotiated between cloud provider and its users. If an SLA violation occurs, then the CSP should pay some penalty to users.

5.1. Performance Metric (Pertric). To maximize the overall performance with minimum energy consumption, average SLA violation, and number of the reactivation hosts, we introduced a performance metric. If the host reactivated from energy saving-mode called reactivated host. These hosts directly affect the energy consumption of the data center. To address this concern, a performance metric is described as follows:

$$Pertric = ASLA \times HS \times E, \tag{9}$$

(1) **Input:** Dataset of the CPU utilization
(2) **Output:** Boolean // Host is overloaded or Not
(3) Initiate the $Y[]$ and $X[]$ // $Y[]$ is the CPU utilization dataset.
(4) **for each** $j \in [1, 100]$ **do**
(5) **for each** $i \in [Y.\text{length}]$ **do**
(6) $\varepsilon_i \leftarrow Y_i - (\theta_1 + \theta_2 X_i)$
(7) **end for**
(8) Calculated the σ
(9) $\sigma \longleftarrow \dfrac{\text{median}|\varepsilon_i - \text{median}(\varepsilon_i)|}{0.6745}$
(10) Initialised $\widehat{\varepsilon}_i[]$ array
(11) **for each** $i \in [Y.\text{length}]$ **do**
(12) $\widehat{\varepsilon}_i \longleftarrow \dfrac{(Y_i - (\theta_1 + \theta_2 X_i))}{\sigma}$
(13) **end for**
(14) Calculate Tukey's bisquare function
(15) **if** $\widehat{\varepsilon}_i \le c$ **then**
(16) $\mathcal{U}(\widehat{\varepsilon}_i) \longleftarrow \dfrac{\widehat{\varepsilon}_i^2}{2} - \dfrac{\widehat{\varepsilon}_i^4}{2c^2} + \dfrac{\widehat{\varepsilon}_i^6}{6c^4}$
(17) **else if** $\widehat{\varepsilon}_i > c$ **then**
(18) $\mathcal{U}(\widehat{\varepsilon}_i) \longleftarrow \dfrac{\widehat{\varepsilon}_i^2}{6}$
(19) Calculate the weighted value
(20) **if** $\widehat{\varepsilon}_i \le c$ **then**
(21) $w_i \longleftarrow \left(1 - \left(\dfrac{\widehat{\varepsilon}_i}{c}\right)^2\right)^2$
(22) **else if** $\widehat{\varepsilon}_i > c$ **then**
(23) $w_i \longleftarrow 0$
(24) Finding the value of θ_1 and θ_2 by using as follows
(25) $\sum_{i=1}^{m} X_i w_i \left(\dfrac{(Y_i - (\theta_1 + \theta_2 X_i))}{\sigma}\right) \longleftarrow 0$
(26) **end for**
(27) MeReg $\leftarrow$ minimum value of $\mathcal{U}(\widehat{\varepsilon}_i)$
(28) $upT \leftarrow p \times$ MeReg
(29) **return** HostUtilisation $> upT$

Algorithm 1: *MeReg* host overloaded detection.

where *Pertric* represents the overall performance metric, *HS* represents the total number of the host shutdowns after applying these algorithms, and E is the total electric energy consumption of the data center. The average SLA violation percentage in the data center is represents as *ASLA*.

6. Experiment Setup

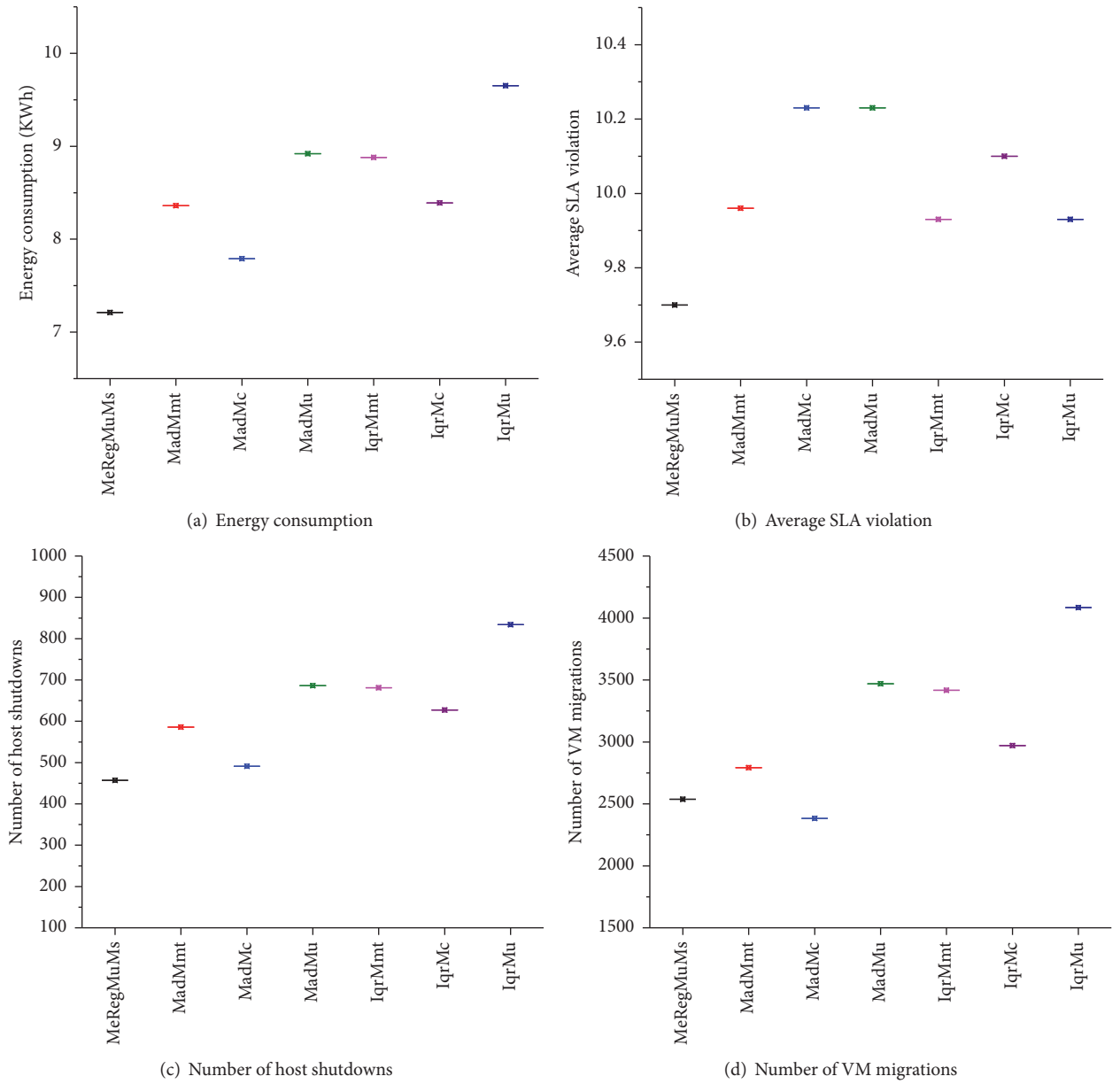
The deployment of real large-scale virtualized infrastructure is very expensive and conducting a repeatable experiment to analyse and compare the result of the proposed algorithm is difficult. Therefore, simulation is a best choice for evaluating VM selection policy to repeat the experiment of the proposed algorithms. We chose the CloudSim toolkit [40] for analysis and compared the performance of the proposed host overloaded detection algorithm. This is a modern open source simulator, which provides an IaaS cloud computing framework that enables us to conduct repeatable experiments for which results can be analysed and compared on large-scale virtualized cloud data centers.

Table 3: Types of Amazon EC2 VM.

VM Types	MIPS	Memory
Hight-CPU instance	2500	850 MB
Extra-large instance	2000	3750 MB
Small instance	1000	1700 MB
Microinstance	500	613 MB

In our cloud computing simulation setup, we installed 800 heterogeneous servers with real configurations. These hosts are Fujitsu M1, Fujitsu M3, Hitachi TS10, and Hitachi SS10. The features of these servers are presented in Table 2. The electric energy consumption of these servers at different workloads is shown in Table 1.

The CPU clock speed of servers is mapped onto MIPS ratings; that is, each core of the servers Fujitsu M1, Fujitsu M3, Hitachi TS10, and Hitachi SS10 is mapped 2700, 3500, 3500, and 3600 MIPS, respectively. The network bandwidth of each server is modeled to possess 1 GB/s. The corresponding VM types are supported by Amazon EC2 VM, as shown in Table 3.



(a) Energy consumption

(b) Average SLA violation

(c) Number of host shutdowns

(d) Number of VM migrations

Figure 2: Evaluation of the proposed host overload detection algorithm using random workload.

Simulation must be conducted using real workload traces of the data center server, which is applicable on real cloud environment. To achieve this objective, we used the data provided by PlanetLab as part of the CoMon project [41]. We utilized more than a thousand heterogeneous VM CPU utilization data from more than 500 heterogeneous servers placed worldwide. The features of the data daily are discussed in Beloglazov and Buyya [18].

7. Simulation and Analysis

Real time CPU utilization data of heterogeneous servers is used to evaluate the performance of *MeReg* host overloaded detection algorithm. We simulated the proposed algorithm with the *MuMs* VM selection scheme and compared it with the overloaded hosts detection algorithms and VM selection policy described in Beloglazov and Buyya [18]. These overloaded host algorithms are median absolute deviation (MAD), and interquartile range (IQR) with maximum correlation (MC), minimum migration time (MMT), and minimum utilization (MU) of VM selection policy. We used the values of safety parameters (p) 1, 2.5, and 1.5 for *MeRegMuMs*, *MAD*, and *IQR*, respectively.

7.1. Random Workload. In the random workload, every VM runs an application with a variable utilization of CPU, which is generated with a uniform distribution. In Figure 2(a), the electric energy consumption by using *MeRegMuMs* host overloaded detection algorithm must be lesser than the other approaches. Figure 2(b) shows significant reduction

in average SLA violation. Moreover, in Figures 2(c) and 2(d) the number of shutdown hosts and VM migrations are also reduced more efficiently than the other host overloaded detection algorithms.

7.2. Real Workload. The real workload dataset is provided by the PlanetLab as part of the CoMon project. In the CoMon project, data of thousands of VMs CPU utilization worldwide are collected every five minutes and stored in different extension files. We selected this real dataset to evaluate the proposed policy. Analysis of the proposed policy using real workload is discussed in the following subsections.

7.2.1. Evaluation of Energy Consumption. The total electric energy consumption of the resources of the hosts in the data center depends on CPU utilization, primary memory, network devices, and disks. However, numerous studies have revealed that the host CPU consumes more electric energy than other resources in the hosts [29]. Therefore, we are more focused on the CPU utilization of hosts. In this section we analysed the simulation of *MeRegMuMs* host overloaded detection with the MAD and IQR. As shown in Figure 3, electric energy consumption by the proposed algorithm is 17.3% lesser than means of other algorithms.

7.2.2. Evaluation of the Average SLA Violation. Maintaining the QoS is an important aspect of cloud computing environment. The required QoS are determined by SLAs [9]. In this section, we analysed and compared the percentage of average SLA violation in overloaded hosts. Cloud users do not want SLA violation and performance degradation. If these situations occur then CSP should pay the penalty to users. Thus, reduced SLA violation is desired among users and CSPs. Figure 4 shows that the percentage of average SLA using the *MeRegMuMs* host overloaded detection is 23.3% lesser than that of traditional algorithms.

7.2.3. Number of Host Shutdowns and VM Migrations. The cost of dynamic live migration of VMs is always high, which includes processing power on the allocated host, and performance degradation [9, 14]. Therefore, minimizing the total number of VMs migrations is one of the objectives of this study. In this section, we analysed and compared the simulation of the number of host shutdowns and VM migrations. If the number of reactivated hosts increase, then energy consumption is maximized. The host is reactivated to allocate new VMs and shutdown when it detect underload.

In the experiment environment, we installed 800 hosts but the number of host shutdowns is greater than 800 due to host reactivation. Figure 5 shows that the proposed algorithm also, minimized 25.9% of host reactivations of hosts relative to traditional MadMmt, MadMc, MadMu, IqrMmt, IqrMc, and IqrMu algorithms.

Meanwhile, the number of migration is directly proportional to performance degradation. If the total number of VM migrations decreases then performance degradation also decreases, which is desired by users and CSPs. The comparison of the proposed policy VM migration with other

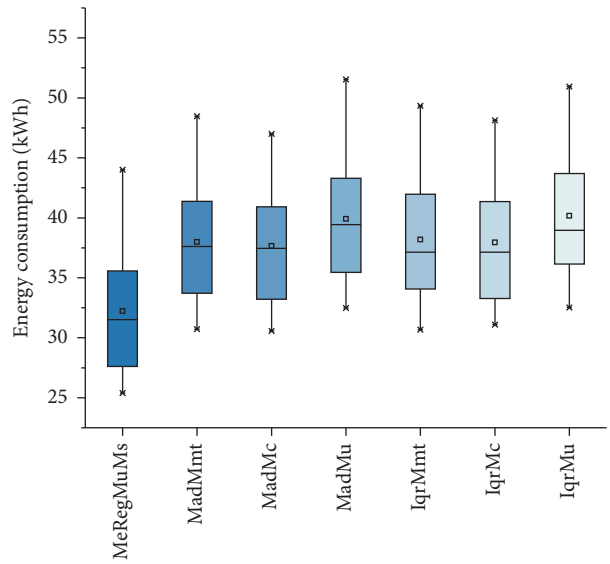


Figure 3: Energy consumption comparison using real workload.

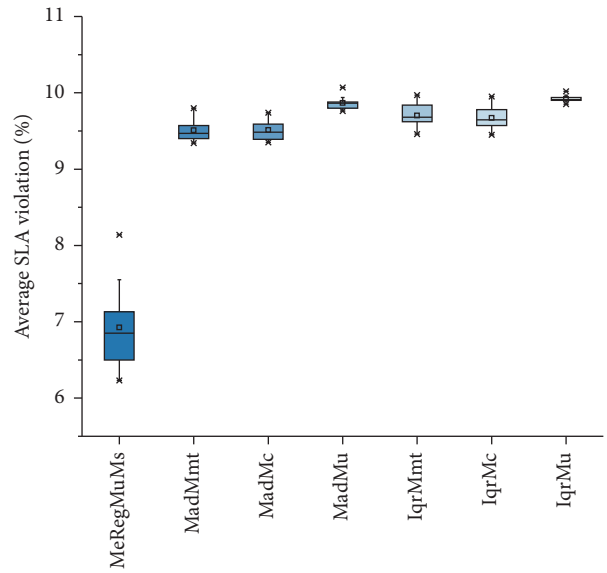


Figure 4: Comparison of the average percentage of SLA violation.

old algorithms proposed in Beloglazov and Buyya [18] is described in Figure 6.

7.2.4. Evaluation of Pertric. In this section, we discussed the overall performance of the cloud data center using proposed *MeReg* host overloaded detection algorithm. The overall performance calculated by the Pertric metric proposed in Section 5.1 is also discussed. The main objective is to propose this metric to analyse the all aspects of energy-awareness in the cloud data center, such as minimization of electric energy consumption, average percentage SLA violation, and number of reactivated hosts for placing new VMs.

Figure 7 shows the effectiveness of the *MeReg* host overload detection algorithms using MuMs VMs selection policy relative to other old host overload detection algorithms

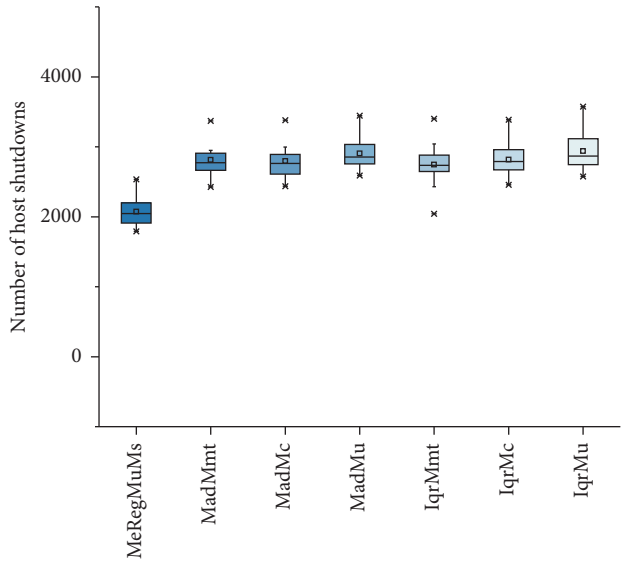


Figure 5: Total number of host shutdowns.

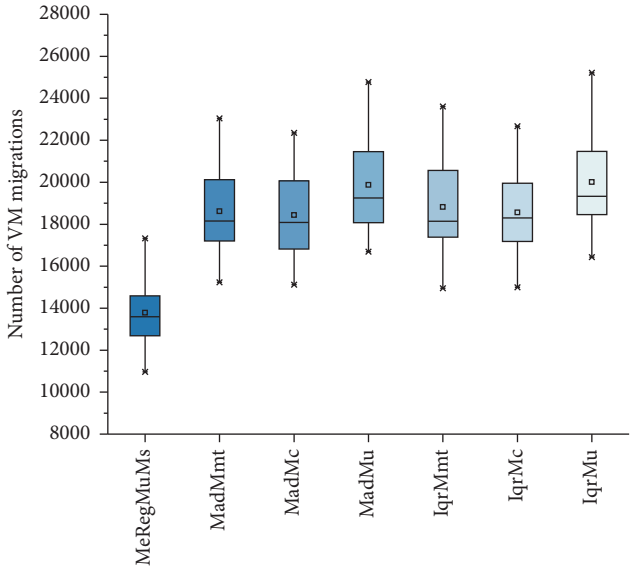


Figure 6: Total number of VM migrations.

using VM selection policies such as MadMmt, MadMc, MadMu, IqrMmt, IqrMc, and IqrMu.

7.2.5. Statistical Analysis. Statistical analysis validated the proposed algorithm, and the results demonstrated the efficiency of the proposed algorithm compared with other approaches. One-way ANOVA on the Pertric Matrices is conducted to analyse the tradeoff between minimizing the overall energy consumption and maximizing the QoS of the data center demonstrated in Table 4. Based on the One-way ANOVA result, *MeRegMuMs* significantly reduced energy consumption and maximized QoS, compared with *MadMc, MadMmt, MadMu, IqrMc, IqrMmt,* and *IqrMu*. Table 4 shows that the F ratio (10.61) is greater than the F critical value (2.24), which indicates that the null hypothesis is rejected

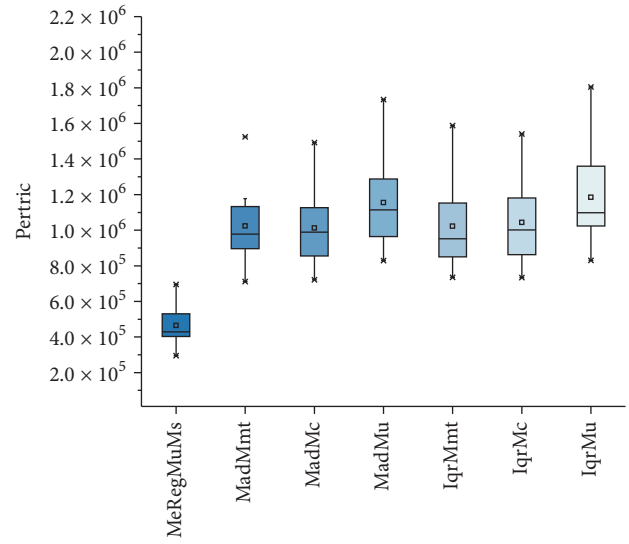


Figure 7: Performance metric (Pertric) comparison.

and the population means are significantly different from one another at the 0.05 level. Therefore, the *MeRegMuMs* algorithm is significantly different from other algorithms, such as *MadMc, MadMmt, MadMu, IqrMc, IqrMmt,* and *IqrMu* with p value of $4.068E - 8$.

One sample t-test of VM migration time duration and host running time is also carried out. The average value of the sample mean times before a VM migration during the host detection underload or overload is 19.67 seconds with a 95% CI: 18.23, 20.12. The average value of the sample means host running time before transition to energy-saving-mode is 21.3 minutes with 95% CI: 20.2, 22.8.

8. Conclusion and Future Work

Mobile cloud computing enables seamless and rich functionality of the cloud computing services to mobile users. Mobile cloud data centers worldwide are growing according to the increasing demand of data processing and storage by mobile users. Therefore, to keep the mobile cloud data centers running, massive amount of electric energy is required, which leads to high operational costs and CO_2 emission. High emission of CO_2 negatively impacts the social environment. In this study, we introduced a novel adaptive heuristic host overload detection algorithm called *MeReg*, which minimizes electric energy consumption and maximize QoS in terms of required SLA of the data center. A host overload problem directly influences VM performance, which is totally against SLA. Therefore, a regression-based technique called M estimation is used to find optimal upper CPU utilization threshold for detecting overloaded hosts. For VM consolidation from overloaded hosts, the approach used in previous study called *MuMs* policy is implemented, which selects VM from overloaded or underloaded hosts and migrates it to appropriate hosts. CloudSim simulator is used in the implementation of the proposed algorithm to obtain the results using 10 different real workload traces.

Table 4: Summary of the one-way ANOVA test.

Source of variation	df	SS ($\times 10^{10}$)	MS ($\times 10^{10}$)	F ratio	p value	F critical
Between groups	6	347	57.8	10.61	$4.07E-8$	2.246
Within groups	63	343	5.45			
Total	69	690				

In the future, we plan to further extend this work by introducing a machine learning based technique called Markov chain for VM consolidation policy, which works better in a dynamic environment such as cloud computing. The implementation of these algorithms in the open source real cloud platform such as OpenStack would also be studied.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The National Key Research and Development Plan under Grant no. 2016YFB0800801 and the National Science Foundation of China (NSFC) under Grants nos. 61672186 and 61472108 support this work.

References

[1] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: Architecture, applications, and approaches," *Wireless Communications and Mobile Computing*, vol. 13, no. 18, pp. 1587–1611, 2013.

[2] N. Fernando, S. W. Loke, and W. Rahayu, "Mobile cloud computing: a survey," *Future Generation Computer Systems*, vol. 29, no. 1, pp. 84–106, 2013.

[3] S. Vakilinia, B. Heidarpour, and M. Cheriet, "Energy efficient resource allocation in cloud computing environments," *IEEE Access*, vol. 4, pp. 8544–8557, 2016.

[4] I. Petri, H. Li, Y. Rezgui, Y. Chunfeng, B. Yuce, and B. Jayan, "A HPC based cloud model for real-time energy optimisation," *Enterprise Information Systems*, vol. 10, no. 1, pp. 108–128, 2016.

[5] B. P. Rimal, E. Choi, and I. Lumb, "A taxonomy and survey of cloud computing systems," in *Proceedings of the 5th International Joint Conference on INC, IMS and IDC (NCM '09)*, pp. 44–51, Seoul, Republic of Korea, August 2009.

[6] G. Motta, N. Sfondrini, and D. Sacco, "Cloud computing: An architectural and technological overview," in *Proceedings of the International Joint Conference on Service Sciences, Service Innovation in Emerging Economy: Cross-Disciplinary and Cross-Cultural Perspective (IJCSS '12)*, pp. 23–27, May 2012.

[7] S. Lambert, W. Van Heddeghem, W. Vereecken, B. Lannoo, D. Colle, and M. Pickavet, "Worldwide electricity consumption of communication networks," *Optics Express*, vol. 20, no. 26, pp. B513–B524, 2012.

[8] L. A. Barroso and U. Hölzle, "The case for energy-proportional computing," *The Computer Journal*, vol. 40, no. 12, pp. 33–37, 2007.

[9] F. Farahnakian, A. Ashraf, T. Pahikkala et al., "Using Ant Colony System to Consolidate VMs for Green Cloud Computing," *IEEE Transactions on Services Computing*, vol. 8, no. 2, pp. 187–198, 2015.

[10] A. Corradi, M. Fanelli, and L. Foschini, "VM consolidation: A real case based on OpenStack Cloud," *Future Generation Computer Systems*, vol. 32, no. 1, pp. 118–127, 2014.

[11] W.-Z. Zhang, H.-C. Xie, and C.-H. Hsu, "Automatic memory control of multiple virtual machines on a consolidated server," *IEEE Transactions on Cloud Computing*, vol. 5, no. 1, pp. 2–14, 2017.

[12] S. E. Dashti and A. M. Rahmani, "Dynamic VMs placement for energy efficiency by PSO in cloud computing," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 28, no. 1-2, pp. 97–112, 2016.

[13] Y.-J. Hong, J. Xue, and M. Thottethodi, "Dynamic server provisioning to minimize cost in an IaaS cloud," in *Proceedings of the ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems, SIGMETRICS'11*, pp. 147-148, ACM, June 2011.

[14] A. Beloglazov, R. Buyya, Y. C. Lee, and A. Zomaya, "A taxonomy and survey of energy-efficient data centers and cloud computing systems," *Advances in Computers*, vol. 82, pp. 47–111, 2011.

[15] Z. Cao and S. Dong, "Dynamic VM consolidation for energy-aware and SLA violation reduction in cloud computing," in *Proceedings of the 13th International Conference on Parallel and Distributed Computing, Applications, and Technologies, PDCAT '12*, pp. 363–369, IEEE, December 2012.

[16] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing," *Future Generation Computer Systems*, vol. 28, no. 5, pp. 755–768, 2012.

[17] R. N. Calheiros, R. Ranjany, and R. Buyya, "Virtual machine provisioning based on analytical performance and QoS in cloud computing environments," in *Proceedings of the 40th International Conference on Parallel Processing, ICPP '11*, pp. 295–304, September 2011.

[18] A. Beloglazov and R. Buyya, "Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in Cloud data centers," *Concurrency and Computation: Practice and Experience*, vol. 24, no. 13, pp. 1397–1420, 2012.

[19] R. Yadav, W. Zhang, H. Chen, and T. Guo, "MuMs: Energy-Aware VM Selection Scheme for Cloud Data Center," in *Proceedings of the 28th International Workshop on Database and Expert Systems Applications (DEXA)*, pp. 132–136, Lyon, France, August 2017.

[20] S. Kosta, A. Aucinas, P. Hui, R. Mortier, and X. Zhang, "Thinkair: dynamic resource allocation and parallel execution in the cloud for mobile code offloading," in *Proceedings of the IEEE INFOCOM*, pp. 945–953, IEEE, March 2012.

[21] S. U. R. Malik, K. Bilal, S. U. Khan, B. Veeravalli, K. Li, and A. Y. Zomaya, "Modeling and analysis of the thermal properties

exhibited by cyberphysical data centers," *IEEE Systems Journal*, vol. 11, no. 1, pp. 163–172, 2017.

[22] W. Zhang, S. Han, H. He, and H. Chen, "Network-aware virtual machine migration in an overcommitted cloud," *Future Generation Computer Systems*, vol. 76, pp. 428–442, 2017.

[23] W. Zhang, Y. Wen, K. Guan, D. Kilper, H. Luo, and D. O. Wu, "Energy-optimal mobile cloud computing under stochastic wireless channel," *IEEE Transactions on Wireless Communications*, vol. 12, no. 9, pp. 4569–4581, 2013.

[24] S. Esfandiarpoor, A. Pahlavan, and M. Goudarzi, "Structure-aware online virtual machine consolidation for datacenter energy improvement in cloud computing," *Computers and Electrical Engineering*, vol. 42, pp. 74–89, 2015.

[25] X. Zhu, D. Young, B. J. Watson et al., "1000 Islands: integrated capacity and workload management for the next generation data center," in *Proceedings of the 5th International Conference on Autonomic Computing, ICAC '08*, pp. 172–181, June 2008.

[26] R. Nathuji and K. Schwan, "VirtualPower: Coordinated power management in virtualized enterprise systems," *ACM SIGOPS Operating Systems Review*, vol. 41, no. 6, pp. 265–278, 2007.

[27] P. Ranganathan, P. Leech, D. Irwin, and C. Jeffrey, "Ensemble-level power management for dense blade servers," *ACM SIGARCH Computer Architecture News*, vol. 34, no. 2, pp. 66–77, 2006.

[28] V. Venkatachalam, M. Franz, and C. W. Probst, "A new way of estimating compute-boundedness and its application to dynamic voltage scaling," *International Journal of Embedded Systems*, vol. 3, no. 1-2, pp. 17–30, 2007.

[29] X. Fan, W.-D. Weber, and L. A. Barroso, "Power provisioning for a warehouse-sized computer," *ACM SIGARCH Computer Architecture News*, vol. 35, no. 2, pp. 13–23, 2007.

[30] C. Lange, D. Kosiankowski, R. Weidmann, and A. Gladisch, "Energy consumption of telecommunication networks and related improvement options," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 17, no. 2, pp. 285–295, 2011.

[31] Y. Shang, D. Li, and M. Xu, "Energy-aware routing in data center network," in *Proceedings of the 1st ACM SIGCOMM workshop on Green networking*, pp. 1–8, ACM, August 2010.

[32] I. Takouna, R. Rojas-Cessa, K. Sachs, and C. Meinel, "Communication-aware and energy-efficient scheduling for parallel applications in virtualized data centers," in *Proceedings of the IEEE/ACM 6th International Conference on Utility and Cloud Computing, UCC '13*, pp. 251–255, IEEE Computer Society, December 2013.

[33] R. Liu, H. Gu, X. Yu, and X. Nian, "Distributed flow scheduling in energy-aware data center networks," *IEEE Communications Letters*, vol. 17, no. 4, pp. 801–804, 2013.

[34] S. C. Shah, "Recent Advances in Mobile Grid and Cloud Computing," *Intelligent Automation and Soft Computing*, pp. 1–13, 2017.

[35] D. Kusic, J. O. Kephart, J. E. Hanson, N. Kandasamy, and G. Jiang, "Power and performance management of virtualized computing environments via lookahead control," *Cluster Computing*, vol. 12, no. 1, pp. 1–15, 2009.

[36] C.-H. Hsu and S. W. Poole, "Power signature analysis of the SPECpower_ssj2008 benchmark," in *Proceedings of the IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS '11)*, pp. 227–236, Austin, Tex, USA, April 2011.

[37] Y. Susanti, H. Pratiwi, H. Sulistijowati, and T. Liana, "M Estimation, S estimation, and MM estimation in robust regression," *International Journal of Pure and Applied Mathematics*, no. 3, pp. 349–360, 2014.

[38] H. Edelsbrunner and D. L. Souvaine, "Computing least median of squares regression lines and guided topological sweep," *Journal of the American Statistical Association*, vol. 85, no. 409, pp. 115–119, 1990.

[39] J. Fox, *Applied Regression Analysis and Generalized Linear Models*, Sage, 2015.

[40] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. de Rose, and R. Buyya, "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software: Practice and Experience*, vol. 41, no. 1, pp. 23–50, 2011.

[41] K. Park and V. S. Pai, "CoMon: a mostly-scalable monitoring system for PlanetLab," *ACM SIGOPS Operating Systems Review*, vol. 40, no. 1, pp. 65–74, 2006.

Hindawi
Wireless Communications and Mobile Computing
Volume 2017, Article ID 5471721, 21 pages
https://doi.org/10.1155/2017/5471721

Research Article

Distributed Image Compression Architecture over Wireless Multimedia Sensor Networks

Sovannarith Heng,[1,2] Chakchai So-In,[1] and Tri Gia Nguyen[3]

[1]*Applied Network Technology (ANT) Laboratory, Department of Computer Science, Faculty of Science, Khon Kaen University, Khon Kaen, Thailand*
[2]*Department of Computer Science, Faculty of Science, Royal University of Phnom Penh, Phnom Penh, Cambodia*
[3]*Department of Information Technology, Duy Tan University, Da Nang, Vietnam*

Correspondence should be addressed to Chakchai So-In; so-in@ieee.org

Received 25 August 2017; Accepted 15 November 2017; Published 13 December 2017

Academic Editor: Santiago Mazuelas

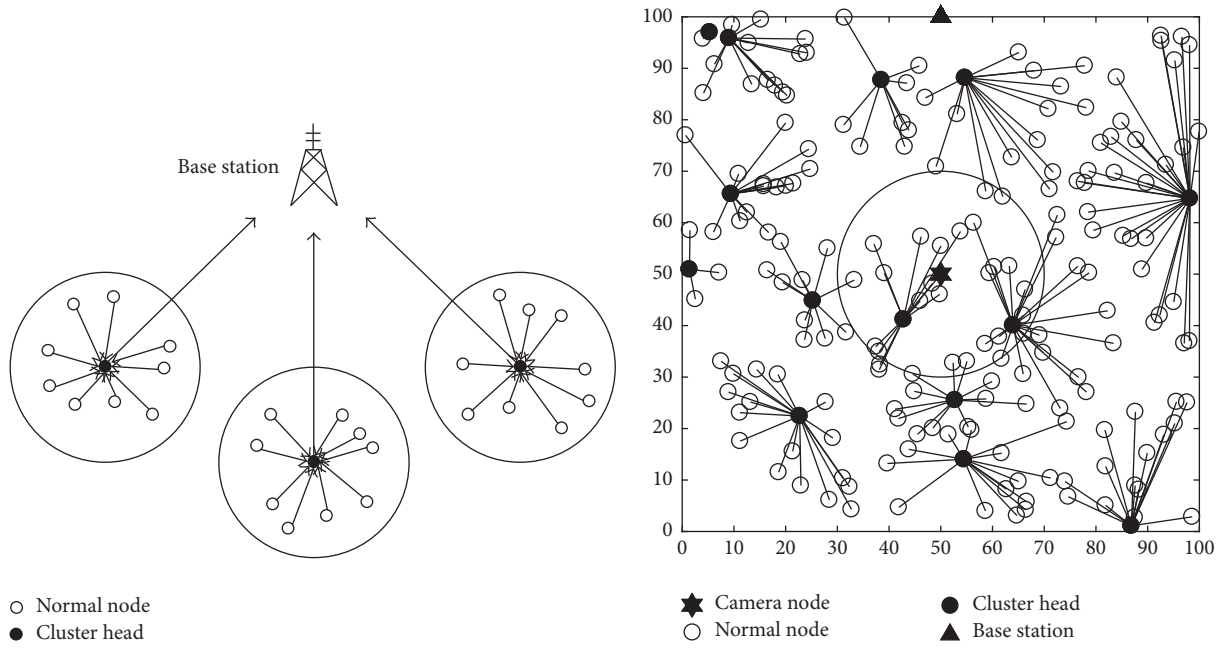
In a wireless multimedia sensor network (WMSN), the minimization of network energy consumption is a crucial task not just for scalar data but also for multimedia. In this network, a camera node (CN) captures images and transmits them to a base station (BS). Several sensor nodes (SNs) are also placed throughout the network to facilitate the proper functioning of the network. Transmitting an image requires a large amount of energy due to the image size and distance; however, SNs are resource constrained. Image compression is used to scale down image size; however, it is accompanied by a computational complexity trade-off. Moreover, direct image transmission to a BS requires more energy. Thus, in this paper, we present a distributed image compression architecture over WMSN for prolonging the overall network lifetime (at high throughput). Our scheme consists of three subtasks: determining the optimal camera radius for prolonging the CN lifetime, distributing image compression tasks among the potential SNs to balance the energy, and, finally, adopting a multihop hierarchical routing scheme to reduce the long-distance transmission energy. Simulation results show that our scheme can prolong the overall network lifetime and achieve high throughput, in comparison with a traditional routing scheme and its state-of-the-art variants.

1. Introduction

The Internet plays a vital role in communication by connecting people around the world through millions of networking devices [1]. The availability of advanced low-cost devices has led to the development of the Internet of Things (IoT). The Internet is moving toward the IoT and Cloud Computing, including Big Data, which has become the most important global technology platform for the future [2]. Typically, the IoT is composed of smart things (or smart devices), which have the ability to detect (sense) or cooperate (communicate) with a physical environment and other devices over large-scale and easy-to-deliver channels (wireless) with optional preprocessing or control those devices to deliver some particular tasks (processing) [1].

The wireless sensor network (WSN) is another key part of the IoT, which is used in many fields, such traffic, military, industry, and environmental monitoring and control [2, 3]. In general, a single wireless sensor node (SN) consists of a readily available and inexpensive sensor, a data processor, memory, a receiver/transceiver, and power units for capturing scalar data, such as temperature, pressure, humidity, velocity, acceleration, and location [4]. Recently, information has been changing from scalar data to multimedia data, such as image, video, and audio data. The IoT and Cloud Computing also provide a backend solution for processing such huge multimedia data streams, which have high computational requirements; for example, Wireless Multimedia Sensor Networks (WMSNs) are used for video streaming, video conferencing, and video surveillance [5].

A WMSN extends a typical WSN by adding multimedia services and devices, such as a Complementary Metal Oxide Semiconductor (CMOS) camera and a microphone, to capture images, video, or audio, which allows the device to retrieve not only scalar data but also multimedia streams. To facilitate understanding, for all devices or sensor nodes



(a) Transmission cluster (WSN)

(b) Camera cluster (WSMN)

Figure 1: Sensor node deployment: example.

(SN) within a WMSN, there are two main types of nodes: camera nodes (CNs) and typical sensor nodes, which are called normal nodes (NNs) and are traditionally used to aid the transmission or perform other scalar-based sensing tasks. Typically, a CN captures the multimedia content and sends it to an NN for processing or transmission to its base station (BS). Although there are various practical transmission schemes [6–9], a hierarchical structure is typically applied, such as a cluster-based approach [10], which has the key advantages of direct-transmission energy optimization and zoning, which prolong the network lifetime.

Figure 1 shows an example of SN deployment; here, there are two possible clusters: (1) a transmission cluster of the NN (here, its representation as cluster head is included) and (2) a camera cluster. As shown in Figure 1(a), a transmission cluster will be normally formed according to criteria that are related to the energy usage, such as distance and remaining power. In each cluster, the sensor node members will send the data to its corresponding cluster head (CH) before further transmitting the aggregated data toward the BS. Similarly, in Figure 1(b), with the introduction of a CN, a camera cluster is a cluster that has one CN and many nearby NNs, which typically function as relay nodes toward the BS. In this cluster, camera cluster optimization is difficult to achieve since the energy of each node has been changing over time. In addition, having too small of a coverage area may lead to having insufficiently many sensor nodes for processing or transmitting data, whereas having too large of a coverage area may cause more energy to be required for communication to its node members.

In general, an SN (NN or CN) comes with resource constraints, such as limited memory, buffer size, data processing, data transmission, and capacity of battery that is very difficult to recharge [11]. Moreover, multimedia data require a large amount of bandwidth and more energy for data delivery, which causes SNs to deplete their energy quickly and shortens the network lifetime; most of the energy of the WMSN is used for data transmission. One possible solution is to compress such data to reduce their size. An image can be compressed because it is composed of many pixels that are closely correlated to one another. By exploiting this correlation, image compression techniques can reduce the size of the image efficiently [12].

Although many image compression techniques have been proposed, most are very complicated and have high resource requirements; thus, they cannot perform well in a resource-constrained WMSN [13–16]. If the image compression process relies on a single SN, this sensor node may quickly run out of energy. A few dead SNs can cause the failure of the entire network, especially SNs that break or divide the network into an unbalanced structure. Accordingly, in the design of a WMSN routing protocol, distributed image compression must be considered [17]. However, designing a distributed scheme that overcomes the limitations of the SN to prolong the network lifetime is challenging since there is always a trade-off between compression quality and energy (such as computing and transmission power).

Furthermore, long-distance data transmission requires more energy than short-distance transmission and causes source SNs to drain their energy quickly; therefore, direct data communication from each individual SN is not efficient in a WMSN; thus, multihop communication is recommended. However, selecting a relay sensor node among all

sensor nodes for forwarding to optimally use the energy of the network is also a complicated task [18].

To overcome the abovementioned issues, we propose a novel distributed architecture of multihop image compression, namely, DICA, to extend the network lifetime and improve throughput using a combination of the fuzzy logic system (FLS) and a distribution-based computation approach. There are three main contributions of this paper: (1) we apply FLS to determine the optimal camera cluster size; (2) we design a distributed image compression technique that divides compression tasks among typical SNs (not camera nodes); and (3) we implement a hierarchical multihop routing technique to partition the network into layers, and similarly, FLS is adopted for optimum relay node selection.

The remainder of this paper is organized as follows: Section 2 highlights some key related works on distributed architectures for image compression in WMSNs. Section 3 describes our system models in detail. Section 4 provides details on our proposed architecture (DICA). Section 5 provides details on the simulation configuration and then presents and discusses the results. The performance of our proposed architecture is evaluated with respect to various metrics, in comparison with an energy-efficient distributed image compression scheme in resource-constrained multihop wireless networks (EEDIC) [14], a two-hop clustered image transmission scheme for maximizing network lifetime in wireless multimedia sensor networks (2HCIT) [19], and a traditional and simplified routing protocol, namely, low-energy adaptive clustering (LEACH) [6]. Finally, the conclusions and the future research directions are discussed in Section 6.

2. Related Work

Typically, a WSN requires a routing protocol to minimize the energy consumption of data transmission. As a clustering-based approach, low-energy adaptive clustering hierarchy (LEACH) is a well-known routing protocol in WSNs. This protocol is an adaptive and self-organizing clustering protocol that generates a random value to select a CH in rotation among SNs. It was originally designed as a single-hop routing protocol, through which SNs transmit data to their CH, which then forwarded the aggregated data toward the BS. In this way, LEACH can distribute energy among SNs in the network to achieve energy reduction, thereby prolonging the network lifetime [6, 20].

Unfortunately, when the scale of the network is increased, LEACH becomes inefficient since the distant CHs deplete their energies faster as they transmit the data to the BS; therefore, LEACH is unsuitable for a large-scale network. To mitigate this problem, multihop LEACH, which is an extension of LEACH, is introduced. In multihop LEACH, instead of sending data directly to the BS, the CH sends data through intermediate CHs, which are closer to the BS. This approach dramatically reduces the energy and extends the lifetime of the WSN [21]. Note that there are also several LEACH derivatives for the purpose of network lifetime optimization [22].

Many researchers have applied LEACH in WSNs. However, LEACH does not consider the effects of multimedia data in WMSNs, such as images with huge size and other considerations, such as computing power and space constraints. Fortunately, as previously stated, an image normally has a large amount of redundant information, which can be removed by using image compression techniques to reduce the image size so that image transmission over multi-hop WMSNs is improved. Over the past decades, many compression techniques have been proposed and can be divided into two categories: discrete cosine transform (DCT) methods and discrete wavelet transform (DWT) methods [21]. As an example of the former, JPEG is a well-known and widely used DCT-based image compression algorithm because it is very fast and energy efficient and requires less memory than other algorithms [23].

In contrast, the latter algorithms perform much better than those in the former algorithms in terms of compression ratio and image quality but with some trade-off as high computational complexity [21]. JPEG2000 is another well-known DWT-based algorithm with the key advantage of providing a better image compression ratio, better image quality, and higher resistance to data transmission error and decoding error; therefore, high-complexity image transmission is not required for prolonging the network lifetime in the context of a WSN or WMSN.

JPEG2000 has been adopted in WMSNs in some works; for example, Zuo et al. [19] presented a two-hop clustered image transmission scheme that was based on traditional LEACH. They divided the clusters into two categories: camera clusters and normal clusters. The camera cluster forms during the first round; a fixed camera radius is carefully considered to ensure that the camera cluster has an adequate number of nodes to receive the image from the CN and then compress the image before sending it to the BS via the CH. At the beginning, the fixed optimized camera cluster radius is determined based on transmission radius adjustment. If the camera radius is too small, there would not be an adequate number of sensor nodes for the camera cluster, whereas if the radius is too large, the image transmission may require more energy than the image compression. In addition, conflicts between camera clusters can occur.

After that, the camera-equipped node assigns the image compression tasks to the nodes that have maximum residual energy within its radius to balance the energies of the camera cluster node members. Then, the camera-equipped node formulates the TDMA schedule based on the compression time, the transmission time of the compressed image, and the transmission time of the original image. JPEG2000 has also been used to make the image size smaller. The experimental results show that this technique can prolong the network lifetime by balancing the energy. However, setting up the camera cluster only once during network initialization is not effective since the energies of the camera cluster member nodes keep changing over time due to many factors. In the worst case, the CN still has energy, but there are no live nodes in the cluster to process and transmit the image to the destination. Furthermore, JPEG2000 is complicated for

a single node to compress an image and the network cannot be scaled using conventional LEACH.

The closest work to ours was presented by Wu et al. [14]. In this study, the wavelet-based image compression standard JPEG2000 is chosen as an image compression technique. The authors distributed the computational load of image compression tasks among SNs with the aim of prolonging the network lifetime. The main idea of the method that is presented in this paper is to distribute the wavelet transform tasks to numerous groups of SNs. To reduce the computational complexity and requirements, two methods were used. In the first method, the image is split into rows and columns with specific sizes, whereas in the second method, the image is cropped into a grid of the same size. Both methods send and distribute wavelet transform of these small images among SNs from source to destination. Nevertheless, in the first method, two round-trip wireless transceivers are required for the SNs who join in each level of the wavelet transform, which may lead to higher power consumption, while in the second method, there is no clear discussion about cluster formation for network load balancing.

To cope with the problems of [14], Tian et al. [15] suggested an algorithm for CH selection that is based on the distance and load factor for improving the centralized encoding algorithm of JPEG2000. First, after being captured by the CN, an image is divided into M blocks of images. Each CN sends those blocks to the M nodes with the highest remaining energy in the cluster to perform the first level of wavelet decomposition. As a result, four subbands are generated: low-low (LL), low-high (LH), high-low (HL), and high-high (HH). Then, the SNs that have participated in the wavelet transform transfer the data to the CH of the next cluster. After receiving the data, the CH forwards only one subband (LL) to a group of SNs for the next level of the wavelet transform, whereas the remaining subbands (LH, HL, and HH) are sent to three other SNs in the cluster for independent encoding. This process is repeated until the desired image compression ratio is achieved and the compressed image has arrived at the BS. This algorithm can decrease the energy consumption, offer load balancing of the network, and prolong the network lifetime. Unfortunately, the quality of the image is not satisfactory, comparing with the original JPEG2000.

Nasri et al. [24] proposed an effective JPEG2000 image compression method for multihop WSNs. This proposed method distributes the image compression tasks among SNs along the way to the BS, similar to the abovementioned studies. First, after receiving the query from the CN from the source SN, CH_1 selects multiple sensor nodes SN_{1i} ($i = 1, \ldots, 4$). The source SN divides the original images into many tiles and sends them to this group of SNs (SN_{11}, SN_{12}, SN_{13}, and SN_{14}) for 1D of DWT decomposition (vertical decomposition). This group of SNs sends the result to CH_2 and then CH_2 sends this result to other sets of SNs (SN_{21}, SN_{22}, SN_{23}, and SN_{24}) for the other 1D of DWT decomposition (horizontal decomposition). As a result, level 1 of the data is retrieved. Next, CH_3 chooses LL_1 and distributes it to a group of SNs (SN_{31}, SN_{32}, SN_{33}, and SN_{34}) for 1D wavelet transform.

After completing the 1D wavelet transform, the SNs in this group send the intermediate result back to CH_3 for the other 1D wavelet transform. CH_3 sends the level 2 data to CH_4 for quantization. Then, CH_4 sends the level 2 data to a single sensor node, namely, SN_{4i}, for quantization, whereas the other SNs in this cluster are awakened. The number of levels depends on the compression target. After quantization, SN_{4i} sends the data to CH_5. Then, CH_5 divides the quantized subbands into multiple smaller code blocks of equal size and sends them to a set of nodes, namely, SN_{5i} (SN_{53}, SN_{52}, SN_{53}, and SN_{54}), for independent code block entropy encoding to produce compressed bitstreams. This procedure continues until the compressed image reaches the BS. This proposed scheme can reduce the memory consumption of SNs by dividing the original image into tiles and blocking quantized subbands into multiple smaller code blocks for entropy encoding. This scheme can also distribute the images to many nodes but needs to transmit uncompressed image along many clusters before the image is completely compressed; therefore, a large amount of energy is used during data transmission.

Another energy-efficient JPEG2000 image compression architecture is proposed by Lu et al. [25]. In this proposed method, there are also two types of clusters: camera clusters and normal clusters. A camera cluster has a CN as its CH. First, the CN captures the image and divides it into tiles. Then, the tiles are sent to the member nodes of the camera cluster to share the compression tasks to minimize the total energy consumption. After finishing the compression tasks, the SNs send the compressed image to the normal CH. Then, the CH forwards the compressed image to the BS. By comparing with the scheme in which the CN performs all compression tasks and sends the compressed image out, this proposed method can reduce the energy consumption of the CN and improve the lifetime of the network. Unfortunately, in the experiment, the network lifetime is improved only in the case of dense SN deployment.

Aside from the routing protocol and image compression, as one of the pioneering forms of computational intelligence (CI) [26], fuzzy logic system (FLS) can be used in many fields to optimize the system when there are uncertainties in the system without requiring the complete information. Many studies have adopted FLS in WSNs, in particular, for routing and clustering. For example, Brante et al. [27] developed a fuzzy-logic-based relay selection scheme for multihop WSNs. The instantaneous channel conditions and the residual energy were used as fuzzy input to select the relay node. Unfortunately, when the density of nodes in the network is increased and relay selection grows dramatically, the end-to-end performance is impacted.

In addition, as stated in [26], the authors determined that, during clustering, the CH should be selected based on the residual energy and rotated periodically to balance the energy consumption. As a result, CHs that are far away from the BS die sooner than those that are nearer to the BS. Here, FLS-based unequal clustering was proposed to partition the network into clusters of different sizes. The clusters that are near the BS have smaller sizes than the distant clusters. FLS with three variables, namely, the distance to the BS, the residual energy, and the node degree, is used to compute

the competitive radius, whereas each SN in the competitive radius computes the fuzzy choice using FIS with two input variables, namely, the distance and the node degree of the CH, to decide whether to join the cluster or not. However, as the network size increases, the clusters in WSN become imbalanced, which can reduce the performance of network since the node degrees and the remaining energies of the CHs are taken into account when SNs are assigned to the cluster.

Bagci and Yazici [28] also proposed the fuzzy-based clustering method. In their paper, each SN generates a random number between 0 and 1. The SN whose random number is below the predefined threshold T becomes a desired tentative CH. Each tentative CH computes the competitive radius using FLS with two input variables: the distance to the BS and the residual energy. A broadcasting message is sent by each tentative CH to compete with the other tentative CHs locally. If a tentative CH receives a broadcasting message from another tentative CH that has higher energy than its remaining energy within its competitive range, the tentative CH broadcasts the quit message; however, if the tentative CH receives broadcasting messages from tentative CHs whose energies are lower than its remaining energy, it becomes the CH. Compared to other unequal and equal clustering algorithms, this algorithm can perform better; unfortunately, it may cause imbalanced energy consumption of SNs in the WSN if the randomly selected CHs have lower remaining energy.

3. System Model

3.1. Network Model. In this paper, we make the following network assumptions:

(i) All sensor nodes (SN) are deployed randomly and uniformly in a targeted area ($M \times M$).

(ii) The network is composed of two types of SNs: camera sensor nodes or camera nodes (CNs) and typical sensor nodes or normal nodes (NNs).

(iii) The number of the CNs is far less than the number of NNs.

(iv) All NNs are homogenous in each type; therefore, they have the same initial energy, computational power, memory, and so on.

(v) All SNs are unaware of their location and become static after deployment.

(vi) All SNs have the same communication range (radius $= R_c$).

(vii) Each SN can approximate its distances to other SNs based on the received signal strength index (RSSI) [29].

(viii) A single BS is located at a specific position that is outside the sensor field and has sufficient hardware, software, and constant power supply.

(ix) The operation is broken into rounds. The transmission of a full image from a CN to the BS is considered one round.

3.2. Energy Consumption Model. While transmitting and receiving data, that is, packets that are based on the IEEE 802.15c framework, in WSNs or WMSNs, each SN consumes energy. The energy consumption is based on the size of the packet and the distance between sender and receiver. This paper adapts a simplified model that is used in LEACH for the communication energy consumption model. The energy consumption for transmitting and receiving l bits of data are given in (1) and (2), respectively [19].

$$E_{\text{tx}}(l,d) = \begin{cases} l \times E_{\text{elec}} + l \times \varepsilon_{\text{fs}} \times d^2, & \text{if } d < d_0 \\ l \times E_{\text{elec}} + l \times \varepsilon_{\text{mp}} \times d^4, & \text{if } d \geq d_0 \end{cases} \quad (1)$$

$$E_{\text{tx}}(l,d) = E_{\text{elec}} \times l, \quad (2)$$

where E_{elec} is the energy that is consumed by the circuit per bit; d is the distance between sender and receiver. ε_{fs} corresponds to free space, whereas ε_{mp} corresponds to multipath fading; they are the energies that are depleted by the amplifier for short and long distances, respectively. $d_0 = \sqrt{\varepsilon_{\text{fs}}/\varepsilon_{\text{mp}}}$ is the reference distance between sender and receiver. If this distance is less than d_0, then ε_{fs} is used; otherwise, ε_{mp} is applied.

The energy consumption of an SN when it receives a k-bit packet is as follows:

$$E_R = k \times E_{\text{elec}}. \quad (3)$$

After receiving data, the SN requires energy for data aggregation. The energy consumption of data aggregation is computed as follows:

$$E_{\text{agg}} = E_{\text{DA}} \times n \times k, \quad (4)$$

where n is the number of messages and E_{DA} is the energy that is used for aggregating some data.

JPEG2000, which is a wavelet-based image compression scheme, is used in this study. The energy consumption is divided into two parts: discrete wavelet transform (DWT) energy consumption and encoding energy consumption. The image compression energy consumption per bit, which is denoted as E_{comp}, is calculated as follows [30]:

$$E_{\text{comp}} = E_{\text{DWT}} \times \sum_{\text{level}=1}^{\text{DWT\_Level}} \left(\frac{1}{4}\right)^{\text{level}-1} + E_{\text{encode}}, \quad (5)$$

where DWT_Level is the desired decomposition level of the wavelet transform.

In an implementation on a StrongARM SA-1100 206 MHz device, E_{elec}, ε_{fs}, and ε_{mp} are measured to be approximately 50 nj/bit, 10 pj/bit, and 0.0013 pj/bit/m^4, respectively. E_{DWT}, which denotes the discrete wavelet transform energy consumption, costs approximately 220 nj/bit, whereas E_{encode} consumes an energy of approximately 20 nj/bit [14, 19].

3.3. Definition and Notation. Before proceeding to the proposed algorithm, the following definitions and notations are presented.

Definition 1 (sensor nodes). The wireless sensor network consists of N sensor nodes. We denote $\mathrm{SN} = \{\mathrm{SN}_1, \mathrm{SN}_2, \mathrm{SN}_3, \ldots, \mathrm{SN}_N\}$, where $|\mathrm{SN}| = N$ and SN_i represents the ith sensor node.

Definition 2 (neighboring nodes). The neighbor set of sensor nodes SN_i is defined as $\mathrm{NB}_i = \{j \epsilon N \mid d_{i,j} \leq 2R_c, i \epsilon N, j \neq i\}$, where N is a set of sensor nodes that are deployed in the area $(M \times M)$, R_c denotes the broadcasting range (communication radius) of the SNs, and $d_{i,j}$ is the Euclidean distance between SN_i and SN_j, which is approximated using RSSI.

Definition 3 (node degree). The node degree is the number of neighbors of an SN in the field within broadcasting radius R_c. The degree of sensor node SN_i is denoted as $\mathrm{Deg}(\mathrm{SN}_i)$. The higher the degree of a sensor node SN_i is, the more sensor nodes that surround SN_i and the better the coverage if SN_i is a cluster head.

Definition 4 (member nodes). Any SN can be a member of a cluster. After a cluster head is elected, any sensor node SN_i that is in the coverage area of cluster head CH_j or whose distance from CH_j is less than R_c has a chance to be a member node of the cluster.

Definition 5 (competitive radius). The competitive radius of a camera node is determined by an appropriate camera cluster size. The competitive radius of camera node CN_i is defined as R_{CN_i} and should satisfy

$$R_{\mathrm{CN}_i} \leq R_c. \tag{6}$$

Definition 6 (communication cluster). For energy efficiency, all SNs are grouped into sets, which are called clusters. A set of SNs consists of many SNs, of which only one is elected to be the cluster head (CH), while the rest are just member nodes. Member nodes send the data to their CH, which then forwards the data to the next destination.

Definition 7 (camera cluster). In a camera cluster, there is one camera node CN and its neighboring nodes $\mathrm{NB}_{\mathrm{CN}}$. This cluster is formed to share the processing and transmission task of the camera node. The camera node acts as the cluster head of the camera cluster.

Definition 8 (image compression cluster). The image compression cluster is formed during data transmission. This cluster is used to compress the image before transmitting it to the BS. The highest-energy node in the camera cluster is selected to be the cluster head of image compression cluster ICH. The neighboring nodes of image compression cluster head $\mathrm{NB}_{\mathrm{ICH}}$ are members of the image compression cluster.

In our algorithm, we use nine different control messages:

(i) *CH-Msg (ID)*: this message is sent by the communication CH to its neighbors to announce itself as a CH and contains only the CH's ID.

(ii) *CL-Join-Request-Msg (CH_ID, ID)*: this message is sent by neighboring nodes of the communication CH to request to join the communication cluster. The message includes the ID of the CH and the ID of the SN that wants to join the cluster.

(iii) *Level-Msg (ID, routing_level)*: the BS and communication CHs broadcast this message, which contains their IDs and routing level information, to their neighbors to build a hierarchical routing structure.

(iv) *Level-Request-Msg (ID)*: during a specific time, communication CHs, which do not receive any *Level-Msg*, broadcast this message to their neighboring SNs to request routing level information. The ID of the requesting CH is stored in this message.

(v) *Level-Reply-Msg (ID, routing_level)*: after receiving a *Level-Request-Msg* from the requesting communication CH, the SN replies with a *Level-Request-Msg,* which contains its ID and routing level information.

(vi) *Cam-Msg (ID, cam_energy, node_degree)*: the CN sends a *Cam-Msg,* which contains its ID, remaining energy, and node degree, to its neighboring nodes to form a camera cluster.

(vii) *Cam-Join-Request-Msg (ID, energy)*: this message is sent by a neighboring node of the CN to join the camera cluster. The message contains the ID and remaining energy of the neighboring node of the CN.

(viii) *Relay-Msg (ID)*: this message is sent by the source SN to its neighbors to find the relay node and contains the ID of the source node.

(ix) *Relay-Reply-Msg (ID, routing_level, energy)*: after receiving *Relay-Msg*, the neighboring nodes of the source node send this message back to source node. This message contains their IDs, routing level information, and remaining energies.

4. Distributed Image Compression Architecture over Multihop Wireless Multimedia Sensor Networks

In this section, our proposed distributed image compression architecture over multihop wireless multimedia sensor networks, namely, DICA, is described in detail. Our proposed architecture is divided into five phases per round: communication cluster setup, multihop hierarchical routing setup, camera cluster setup, image compression cluster setup, and image transmission (see Figure 2).

4.1. Communication Cluster Setup. In our proposed architecture, before proceeding to the other phases, the communication clusters are established. For the purpose of forwarding images to the BS, the WMSN is divided into many clusters, which are called communication clusters. Each communication cluster has its own CH and member nodes. Member nodes send data to their CHs and then forward the data to the destination. We applied LEACH, which is a pioneering cluster-based routing algorithm in terms of effective energy consumption in WMSNs, for our communication cluster setup. To preserve the energy of the

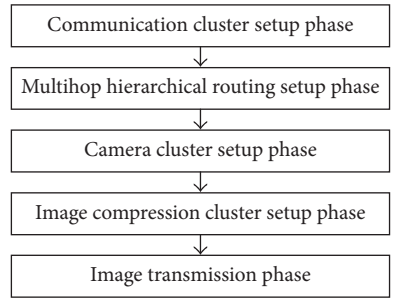


Figure 2: Phases of the proposed architecture.

CNs, CNs cannot function as the CH since the CH depletes more energy for data transmission and aggregation.

4.2. Multihop Hierarchical Routing Setup. After the communication cluster has been set up, the next step is the multihop hierarchical routing setup phase. Here, a level of the hierarchical routing information structure is constructed to offer better routing information for multihop communications. Each SN in the field has its own routing level. The routing level is a number that is used to identify how close the SN is to the BS. The routing level starts from 1 and a low routing level indicates that the SN is near the BS. This setup phase is conducted only in the first round of the network, after the communication cluster setup is completed. This algorithm relies on the communication of CHs to broadcast the routing level information, which is much more efficient than whole-network broadcasting. The details of the multihop hierarchical routing setup are described in Algorithm 1.

In this phase, initially, the routing levels of all SNs in the field are set to 0. The BS broadcasts *Level-Msg* with its ID and routing level (=1) to all SNs that are within its communication radius. After receiving this broadcasting message from the BS, all SNs save the routing level information. In this case, the broadcasting message is received by the CH, and the CH not only saves the routing level information but also increases the routing level by 1 and finally rebroadcasts to its neighbors. This process is repeated until no more CHs receive the broadcasting message. If an SN receives more than one message, it checks the existing routing information that was saved previously against the new information. The lower routing level is accepted and the higher one is discarded, so the routing level of the node is updated with the lowest one.

CHs that do not receive any broadcasting messages for a specific time T_1 are considered far away from other CHs; however, it is possible that their neighbors have the routing level information. Therefore, these CHs broadcast *Level-Request-Msg* to their neighbors to request the routing level information. If there is a neighboring node that receives *Level-Request-Msg* and its routing level is not equal to 0, it replies with *Level-Request-Reply-Msg,* which contains its routing level information to the source CH. After receiving the replies from its neighbors, the CH saves the received routing level information for future use.

If more than one reply message is received, the CH chooses the lowest routing information level. Then, this CH increases the routing level information by 1 and broadcasts it to its neighbors. If the source CH does not receive any replies from its neighbors during a specific time T_2, it waits for another specific time T_3. If, in the worst case, there is still no any reply during time T_3 to the source CH, the source CH sets its routing level to the maximum value (here, 100). For the NNs, after waiting for a specific time T_4 and not receiving any routing level information, these nodes also set their routing levels to the maximum value.

4.3. Camera Cluster Setup. The camera cluster consists of one CN, which acts as the CH of this cluster and nearby member nodes. Once communication cluster setup and multihierarchical routing setup are completed, the camera cluster setup phase is conducted. Since the energies of the member nodes are changing all the time, this phase must be implemented in every round in the network to ensure that the camera cluster has enough member nodes for image processing and transmission.

The detailed algorithm of this phase is described in Algorithm 2. Here, the CN broadcasts the *Cam-Msg* to its neighbors within its competitive radius to form the camera cluster. The competitive radius computation will be discussed in the next subsection. *Cam-Msg* contains the ID, remaining energy, and node degree of the CN in the previous round.

After receiving this message, the neighboring nodes of the CN calculate the camera cluster membership, which will be discussed later. In the camera cluster membership calculation, an FLS cost is calculated. After calculating this cost, whose value is between 0 and 1, the neighboring node generates a random number between 0 and 1, which is called the local chance. If the local chance is lower than the fuzzy cost, the neighboring node sends *Cam-Join-Request-Msg,* which contains the ID and remaining energy of this node, to the CN. Then, this node becomes a member of the camera cluster.

4.3.1. Fuzzy Logic System. Fuzzy logic or set, which was developed by Zadeh [31], is an effective technique for improving decision-making in resource-constrained networks such as WSNs because FLS can reduce the resource consumption while maintaining effective performance and offering good solutions for many control problems by imitating the human thought process. FLS consists of four key components: fuzzier, inference engine, fuzzy rules, and defuzzifier, as shown in Figure 3 [32].

```
(1)  if round == 1 then
(2)      BS broadcasts Level-Msg containing routing_level = 1.
(3)      if SN receives the broadcasting message then
(4)          if Message is Level-Msg then
(5)              if routing_level of node != 0 then
(6)                  if routing_level of node > routing_level within received message then
(7)                      Update routing_level of SN
(8)                      if Node is CH then
(9)                          routing_level ← routing_level + 1
(10)                         Broadcast new Level-Msg (ID, routing_level)
(11)                     end if
(12)                 end if
(13)             Else
(14)                 Update routing_level of SN with routing_level within received message
(15)             end if
(16)         else if Message is Level-Request-Msg then
(17)             if routing_level of node != 0 then
(18)                 Send Reply-Request-Msg (ID, routing_level) to the requested SN
(19)             end if
(20)         End
(21)     else
(22)         Waiting for Level-Msg when T1 is expired
(23)         if Node is CH then
(24)             Broadcast Level-Request-Msg (ID)
(25)             Waiting for Level-Reply-Msg when T2 is expired
(26)             if Node receive Level-Reply-Msg then
(27)                 if Node is CH then
(28)                     Update routing_level of SN with routing_level within received message
(29)                     routing_level ← routing_level + 1
(30)                     Broadcast new Level-Msg (ID, routing_level)
(31)                 end if
(32)             Else
(33)                 Waiting for Level-Reply-Msg when T3 is expired
(34)                 routing_level ← 100
(35)             end if
(36)         Else
(37)             Waiting for Level-Msg when T4 is expired
(38)             routing_level ← 100
(39)         end if
(40)     end if
(41) end if
```

Algorithm 1: Multihop hierarchical routing level setup.

The fuzzifier gets crisp inputs and converts them to a fuzzy set, which is represented by a linguistic term, such as "near," "medium," or "far," using a membership function. This process is known as fuzzification. Used for quantifying linguistic terms, the membership function maps nonfuzzy input values onto fuzzy linguistic terms, and vice versa. There are different forms of membership functions, such as triangular, trapezoidal, Gaussian, piecewise linear, and singleton; however, there are a few membership functions that are commonly used, that is, triangular, trapezoidal, and Gaussian. The choice of membership functions depends on the experience of and assessment by the researcher. The membership function is applied in both fuzzification and defuzzification. After the fuzzification process, the inference is made by inference system based on a set of rules, which are stored in the rule base. Each of these rules is typically an IF-THEN rule with conditions and a conclusion. Lastly, the output of FLS is defuzzified using the membership function. Defuzzification transforms the fuzzy set back into a crisp output value [33].

4.3.2. Competitive Radius. The competitive radius is one of the most important factors for the camera cluster since it is used to limit the SNs that can be members of the camera cluster. If the competitive radius is too large, more energy may be required for the CN to communicate with its node members, whereas if it is too small, there may be an inadequate number of SNs to join the camera cluster as well as processing and transmiting the images to the BS. Based on RSSI, the competitive camera radius is computed using the following formula:

$$R_{\mathrm{CN}} = \propto \mathrm{RSSI}_{\max}, \tag{7}$$

```
(1)  if Node is CN then
(2)      if round = = 1 then
(3)          node_degree = 0;
(4)          competitve_radius = RSSI_max
(5)      else
(6)          node_degree ← calculate node degree
(7)          competitive_radius ← calculate competitive radius based on algorithm in Section 4.3.2
(8)      end
(9)      Broadcast Cam-Msg (ID, cam_energy, node_degree) within its competitive radius
(10) Else
(11)      if Node receives Cam-Msg then
(12)          local_chance ← generate random value
(13)          fuzzy_cost ← calculate Fuzzy_Cost (cam_energy, RSSI, node_degree) based on
              algorithm in Section 4.3.3
(14)          if local_chance ≦ fuzzy_cost then
(15)              Sends Cam-Join-Request-Msg (ID, energy) to CN
(16)          end if
(17)      end
(18) end if
```

Algorithm 2: Fuzzy logic-based camera cluster setup.

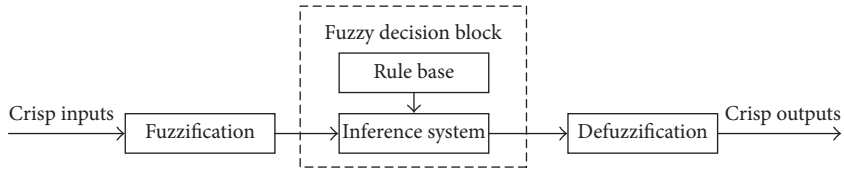


Figure 3: Basic block diagram of a fuzzy system.

where $\mathrm{RSSI}_{\max}$ denotes the maximum RSSI of the SNs and $\propto$ is the weight of the camera radius, which is used to adjust the camera radius. This weight of the camera radius $\propto$ ranges between 0 and 1. In the first round, $\propto$ is equal to 1.

To determine the weight of the camera radius $\propto$, FLS is applied. In FLS, three input variables are used. First, the CN energy is the remaining energy of the CN, as a percentage: $\mathrm{CN\_}Energy_i = \mathrm{CN\_}Remaining\_Energy_i/\mathrm{CN\_}Init\_Energy_i$. Second, the camera node degree is the number of other CNs that are within the competitive radius of the CN, divided by the total number of CNs in the network: $CamDeg(\mathrm{CN}_i) = |CamNB_{\mathrm{CN}_i}|/\#\mathrm{CNs}$. Finally, the node degree of the camera cluster is the number of SNs in the camera cluster in the previous round, divided by the total number of SNs: $Node\_degree = \mathrm{Deg}(\mathrm{CN}_i)/\#\mathrm{SNs}$. Here, $\mathrm{Deg}(\mathrm{CN}_i)$ is set to 0 in the first round of the network.

As shown in Figure 4, all three variables have the same linguistic variables for the fuzzy set: low, medium, and high. Since trapezoidal and triangular membership functions are suitable for real-time operation and their computations are not complex [34], we applied them to our fuzzy input and output variables. All fuzzy input parameters of the membership function are formulated based on [32] and our experiment. Mamdani's method is chosen as the fuzzy inference technique that is used to map the set of input linguistic variables to the output set [35]. A set of 27 fuzzy rules and output membership functions are shown in Table 1 and Figure 5, respectively.

4.3.3. Camera Cluster Membership. The camera cluster membership algorithm also adopts FLS to determine the size of the camera cluster. Not all SNs within the competitive radius can be members of the camera cluster; only the qualified ones are selected to join this cluster. As shown in Table 2, three fuzzy logic input variables are used for camera member node selection. Each of them has its own membership function. The first and second variables are camera node energy and node degree, respectively, which are the same as the input variables of FLS of competitive radius, as shown in Figure 4; therefore, we discuss only the third variable, namely, RSSI, which is a measurement of the power of the radio signal from the camera node (see also Figure 6). Weak, medium, and strong are the linguistic input variables for the fuzzy set. The triangular membership function is used for weak and strong, whereas the trapezoidal membership function is applied to medium [36].

Fuzzy inference relies on 27 rules, which are generated from three fuzzy input variables, as described in Table 2. The output, as depicted in Figure 7, is the fuzzy cost, which determines the possibility that nodes can be members of the camera cluster.

4.4. Image Compression Cluster Setup. The image compression cluster is formed to prepare the SNs for distributed image compression. In the image compression cluster setup phase, initially, the CN selects the highest-energy SN in

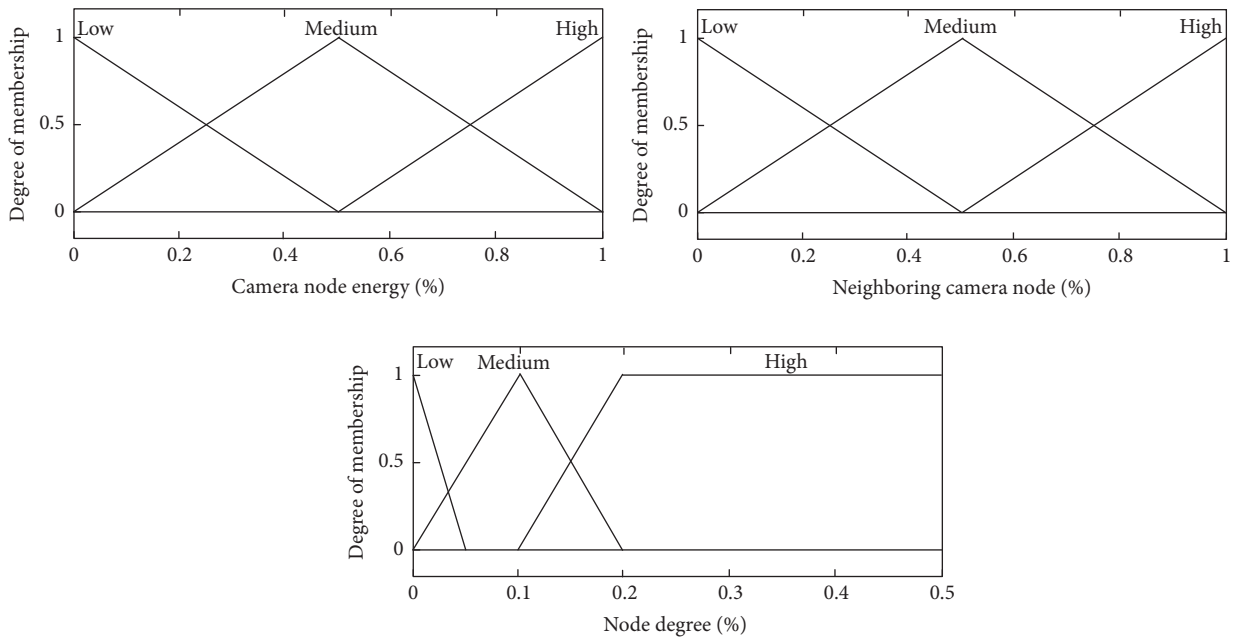


Figure 4: Fuzzy input variables of competitive radius.

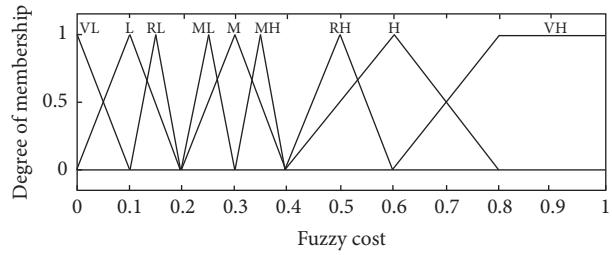


Figure 5: Fuzzy output variable of fuzzy cost in competitive radius calculation. VL = very low, L = low, RL = rather low, ML = medium low, M = medium, MH = medium high, RH = rather high, H = high, and VH = very high.

the camera cluster. This highest-energy SN becomes the image compression cluster head (ICH). Not all neighbors of the ICH become members of the image compression cluster automatically. The ICH chooses a set of SNs that have the highest energies as its image compression cluster to participate in image compression. The number of SNs depends on the compression target; however, the ICH typically chooses the four highest-energy SNs within its communication radius to be members of the image compression cluster.

4.5. Image Transmission. The purpose of the image transmission phase is to ensure that the image has been captured, compressed, and sent hop by hop to the BS successfully. In this phase, there are two main subphases: distributed image compression and relay node selection.

4.5.1. Distributed Image Compression. In this paper, we distribute the discrete wavelet transform level and encoding to different sensor nodes within the image compression cluster. At the beginning, a camera node captures an image and then divides this image equally into several small images (here, 4).

(1) Discrete Wavelet Transform (DWT). As briefly discussed, the discrete wavelet transform is an effective technique for signal analysis and has been widely used in image processing, especially image compression. Here, JPEG2000 adopts DWT in its compression technique. DWT decomposes the image into a set of subbands and the level of decomposition produces four subbands: low-low or image approximate (LL), high-low or diagonal details (HL), low-high or horizontal details (LH), and high-high or vertical details (HH). LL is lower scale of the image, whereas the other subbands are the image details. LL can be decomposed further to produce another level of subbands. The higher the level of the decomposition, the smaller the image compression size is; however, the image compression size does not improve when the decomposition level reaches 5-6. An example of 3 levels of DWT decomposition is shown in Figure 8.

(2) JPEG2000. JPEG2000 can be either lossy or lossless, depending on the wavelet transform and quantization method that are applied. It utilizes the Embedded Block Coding with Optimized Truncation (EBCOT) image compression technique. The difference between JPEG2000 and

Table 1: Fuzzy rules of camera cluster radius.

Camera node energy	Neighboring camera node	Node degree	Fuzzy cost
Low	Low	Low	Very high
Low	Low	Medium	Low
Low	Low	High	Very low
Low	Medium	Low	High
Low	Medium	Medium	Low
Low	Medium	High	Very low
Low	High	Low	High
Low	High	Medium	Low
Low	High	High	Very low
Medium	Low	Low	High
Medium	Low	Medium	Low
Medium	Low	High	Very low
Medium	Medium	Low	Rather high
Medium	Medium	Medium	Low
Medium	Medium	High	Very low
Medium	High	Low	Medium high
Medium	High	Medium	Low
Medium	High	High	Very low
High	Low	Low	Medium
High	Low	Medium	Low
High	Low	High	Very low
High	Medium	Low	Medium low
High	Medium	Medium	Low
High	Medium	High	Very low
High	High	Low	Rather low
High	High	Medium	Low
High	High	High	Very low

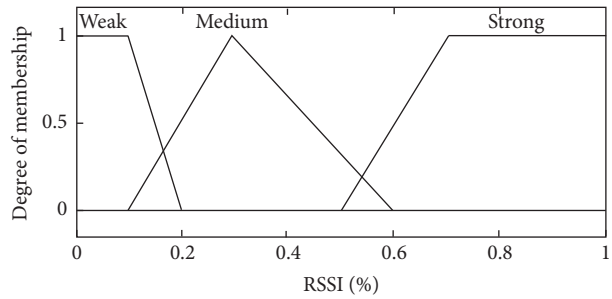


Figure 6: RSSI input variable of camera cluster membership.

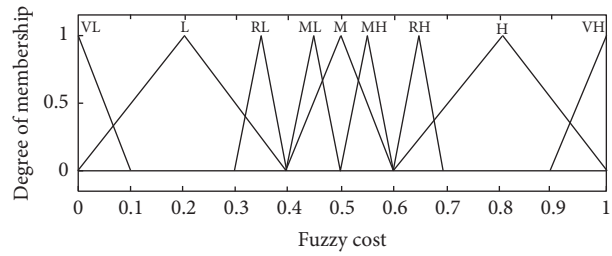


Figure 7: Fuzzy output variable of fuzzy cost in camera cluster membership. VL = very low, L = low, RL = rather low, ML = medium low, M = medium, MH = medium high, RH = rather high, H = high, and VH = very high.

Table 2: Fuzzy rules of camera cluster membership.

Node energy	Node degree	RSSI	Fuzzy cost
Low	Low	Strong	Very high
Low	Low	Medium	Very high
Low	Low	Weak	Very high
Low	Medium	Strong	Very high
Low	Medium	Medium	High
Low	Medium	Weak	Rather high
Low	High	Strong	Very high
Low	High	Medium	Medium
Low	High	Weak	Low
Medium	Low	Strong	Very high
Medium	Low	Medium	High
Medium	Low	Weak	Medium
Medium	Medium	Strong	Very high
Medium	Medium	Medium	Medium
Medium	Medium	Weak	Medium
Medium	High	Strong	Very high
Medium	High	Medium	Low
Medium	High	Weak	Very low
High	Low	Strong	Very high
High	Low	Medium	Rather high
High	Low	Weak	Medium
High	Medium	Strong	Very high
High	Medium	Medium	Low
High	Medium	Weak	Medium low
High	High	Strong	Very high
High	High	Medium	Rather low
High	High	Weak	Very low

Figure 8: Discrete wavelet transform of a grayscale image.

JPEG is that JPEG2000 uses DWT, whereas JPEG exploits DCT. JPEG2000 is far superior to its ancestor since it can offer a higher compression ratio. Moreover, wavelet-based compression techniques such as JPEG2000 are more robust to transmission and decoding errors. Here, we select JPEG2000 for a resource-constrained network in WMSN [14, 37, 38], as shown in Figure 9. For encoding, JPEG2000 compression initially applies DWT and then quantizes and encodes the transform coefficients to generate the output codestream (compressed image). For decoding processes, the codestream is decoded and dequantized. Finally, the inverse DWT is used to reconstruct the image.

(3) Distributed JPEG2000 Compression. As shown in Figure 10, before sending each small image out, the CN forms the image compression cluster based on the algorithm that was discussed in Section 4.4. The detailed algorithm of distributed image compression is described in Algorithm 3. Once the

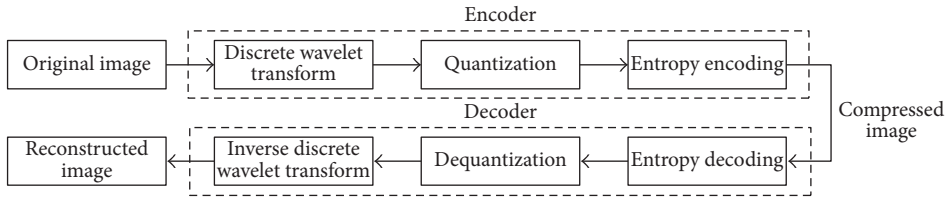


Figure 9: Flow diagram of JPEG2000 image compression.

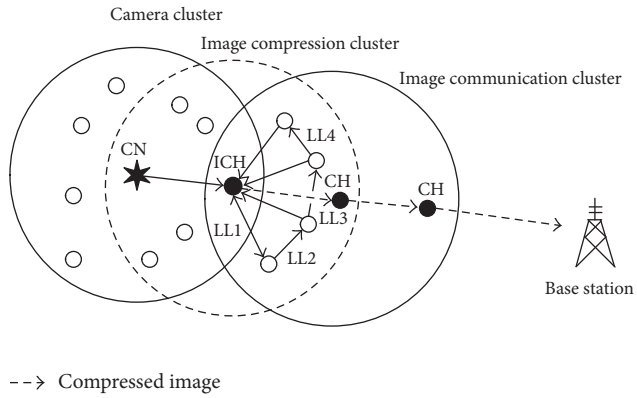


Figure 10: Distributed image compression.

image compression cluster setup is completed, the CN sends each small image to the ICH. After receiving the image from the CN, the ICH applies the first level of DWT on the image and obtains subbands LL, HL, HL, and HL. The process continues by encoding HL, HL, and HL and sends the compressed data to its communication CH, whereas the remaining LL is sent to the nearest SN in the image compression cluster for further processing.

The nearest SN performs the same steps as in the previous process. After receiving the LL, the nearest SN implements another level of DWT; encodes the new HL, HL, and HL; and then sends the compressed data to its ICH, whereas the new LL is sent to the next-nearest SN in the image compression cluster. The processes are repeated until the target compression is achieved. If there are not enough SNs in the image compression cluster, the last SN in the image compression cluster performs the remaining image compression tasks. In this case, there is no member in the image compression cluster, and the ICH performs all image compression tasks. After distributed image compression, the ICH aggregates the compressed image and sends it to its communication CH. Then, the CH forwards the compressed image via other CHs hop by hop to the BS based on the relay node selection algorithm that is presented in Section 4.5.2, until the compressed image reaches the BS.

Since there is a trade-off between compression and transmission complexity, to preserve the energy consumption of the SNs that are participating in compression tasks before processing their normal compression tasks as mentioned above, each SN is required to perform the following preprocessing calculations:

(i) If the node is the ICH, it is required to calculate the energy consumption of implementing an entire compression task and forwarding the compressed image to its CH.

(ii) If the node is a member node in the image compression cluster and is selected to participate in image compression tasks, it needs to calculate the energy consumption of processing all remaining image compression tasks and forwarding the compressed data to the ICH.

The SNs in the image compression cluster choose to implement the preprocessing task if this task consumes less energy than their normal tasks; otherwise, the SNs perform their normal tasks.

4.5.2. Relay Node Selection. In this study, a relay node selection algorithm is used to select the next CHs to forward the compressed image hop by hop to the BS.

We describe the relay node algorithm, which is given in Algorithm 4, as follows: first, the source CH broadcasts a *Relay-Msg,* which contains its ID, to its neighboring CHs within its communication radius to find a relay CH. Once the neighboring CHs receive the broadcasting message, the CHs reply to the source CH by sending a *Relay-Reply-Msg* directly, which contains their IDs, remaining energies, and routing levels. After receiving the reply messages from the neighboring CHs, the source CH chooses the group of neighboring CHs that has lower routing level than itself to calculate the fuzzy cost using FIS, as described later. If there are no neighboring CHs that have lower routing levels, the source CH chooses those with the same routing level as itself. The selected neighboring CH who has the highest fuzzy cost is chosen as the relay CH to forward compressed image to the BS. This process is repeated until the compressed image reaches the BS.

```
(1)  CN selects the highest-energy SN in the camera cluster as the image compression cluster head (ICH)
(2)  if Node is ICH then
(3)        if Node receives image from the CN then
(4)              Select a specified number of SNs that have the highest energy
                 in the image compression cluster as the members
(5)              if Has member in image compression cluster and the shortcut tasks are
                 greater than its normal tasks then
(6)                    ICH applies one level of DWT on the image
(7)                    ICH encodes HH, HL, and LH
(8)                    ICH sends the LL to nearest member node in the image compression cluster
(9)              Else
(10)                   ICH performs all compression tasks
(11)                   ICH sends the compressed data to the ICH's communication CH
(12)             end if
(13)       end if
(14)       if Compression tasks are finished then
(15)             ICH combines all compressed data and sends them to the ICH's communication CH
(16)       end if
(17) else if Node is member of the image compression cluster then
(18)       if Node receives LL from another SN in the image compression cluster then
(19)             Find the next-nearest SN in the image compression cluster
(20)             if Has the next-nearest node in image compression cluster then
(21)                   Node applies one level of DWT on LL of the previous level
(22)                   Node encodes HH, HL, and LH and sends the compressed data to the ICH
(23)                   Node sends the LL of this level to the nearest member node in the image
                       compression cluster
(24)             Else
(25)                   Node performs the remaining compression tasks
(26)                   Node sends the compressed data to the ICH
(27)             end If
(28)       end If
(29) end if
```

Algorithm 3: Distributed image compression.

```
(1)  while Compressed image has not arrived at the base station
(2)  if Node is CH then
(3)        if CH has a neighboring CH then
(4)              Broadcast Relay-Msg (ID)
(5)              Receive Relay-Reply-Msg (ID, routing_level, energy) for neighboring CHs
(6)              Select a group of neighboring CHs that have lowest routing_level
(7)              Fuzzy_cost ← empty
(8)              For each Relay-Reply-Msg
(9)                    Fuzzy_cost[]← calculate Fuzzy (Remaining_energy, RSSI)
(10)             end for
(11)             Next_CH ← Select the CH that has the highest fuzzy cost
(12)             Source CH sends the compressed image to Next_CHs.
(13)       Else
(14)             Source CH sends the compressed image directly to the base station.
(15)       end if
(16)       if Node receives Relay-Msg then
(17)             CH replies to the source CH with Relay-Reply-Msg (ID, routing_level, energy).
(18)       end if
(19) end if
(20) end while
```

Algorithm 4: Relay node selection.

Table 3: Relay node selection fuzzy rules.

Remaining energy	RSSI	Fuzzy cost
Low	Strong	Very low
Low	Medium	Very low
Low	Weak	Low
Medium	Strong	High
Medium	Medium	Medium
Medium	Weak	High
High	Strong	Very high
High	Medium	Very high
High	Weak	High

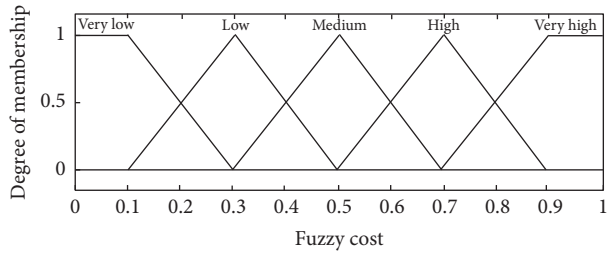


Figure 11: Fuzzy set for the fuzzy cost output variable in relay node selection.

This relay node algorithm also uses FLS as a decision system for choosing the relay CH. In this part, the procedure for obtaining the fuzzy cost of the relay node algorithm is explained in detail. There are two fuzzy input variables for each fuzzy set. The first fuzzy input variable is the remaining energy of the neighboring CHs, which has three linguistic input values: low, medium, and high. For evaluation, the triangular membership function is also applied for medium, whereas the trapezoidal membership function is used with low and high, as shown in Figure 4. The second variable is the RSSI of the neighboring CHs, which has three linguistic input values: strong, medium, and weak. All of them use the triangular membership function, as shown in Figure 6. We also use Mamdani's method as a fuzzy inference technique. There are 9 fuzzy rules for relay node selection, as depicted in Table 3. The output fuzzy variable, as shown in Figure 11, is the fuzzy cost, which determines the next relay node. The highest-fuzzy-cost CH is selected to be the relay node.

A flowchart of the overall processes of DICA is shown in Figure 12.

5. Experimental Results and Discussion

5.1. Experimental Setting. Our algorithm is simulated using MATLAB. The SNs are randomly deployed in a topographical area $A$of dimension 300 m × 300 m. To evaluate the efficiency of our methods, the experiments were conducted on both sparse and dense networks. The number of sensor nodes N (including the NN and the CN) is varied from 100 up to 200. Here, the number of CNs is set to 1 for comparative purposes (randomly placed) [19]. The CN and NNs have initial energies of 15 J and 10 J, respectively. The communication radius R_c of all sensor nodes is 75 m [39, 40]. The CN captures grayscale images of 512 × 512 pixels, each of which is divided equally into 4 blocks before transmission.

Here, there are 5 levels of DWT. The quantization value of image compression is set to 10 [16]. We compared our algorithm with a traditional WSN routing algorithm and the two state-of-the-art distributed image routing architectures, namely, LEACH, 2HCIT [19], and EEDIC [14]. In the LEACH approach, the CN compresses a whole image and sends the compressed image to its communication CH. Then, the compressed image is forwarded to the BS. In other words, the traditional LEACH approach is unaware of multimedia transmission. The simulation parameters are summarized in Table 4. In our simulation, there are two scenarios. The first scenario is conducted with 100 SNs, whereas the second scenario is implemented with 200 SNs. We performed ten trials and averaged the results.

5.2. Results and Discussions. In this section, we present and evaluate the results of simulations of our distributed image compression architecture over WMSNs. We compare the performance of our proposed method with those of EEIC, 2HCIT, and LEACH in terms of the energy consumption, network lifetime, and throughput in networks of different scales.

5.2.1. Hierarchical Routing Level Evaluation. First, examples of the hierarchical routing algorithm in both spare and dense networks are shown Figure 13. In these examples, the whole networks are divided into 4 layers, which correspond to 3 routing levels. The closest SNs to the BS have routing level 1 and the furthest have routing level 3. All of the SNs, whether in sparse or dense networks, have their own routing levels.

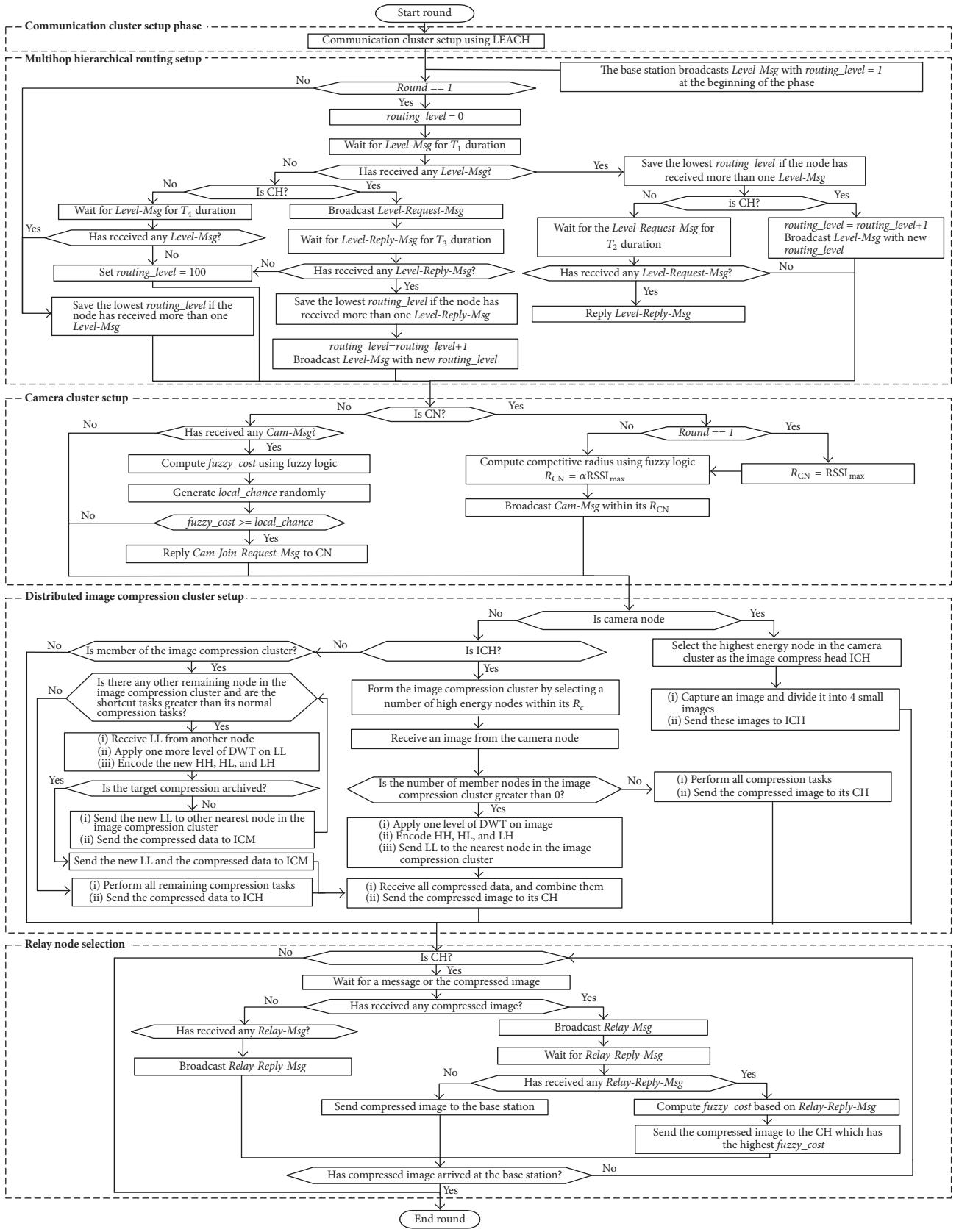
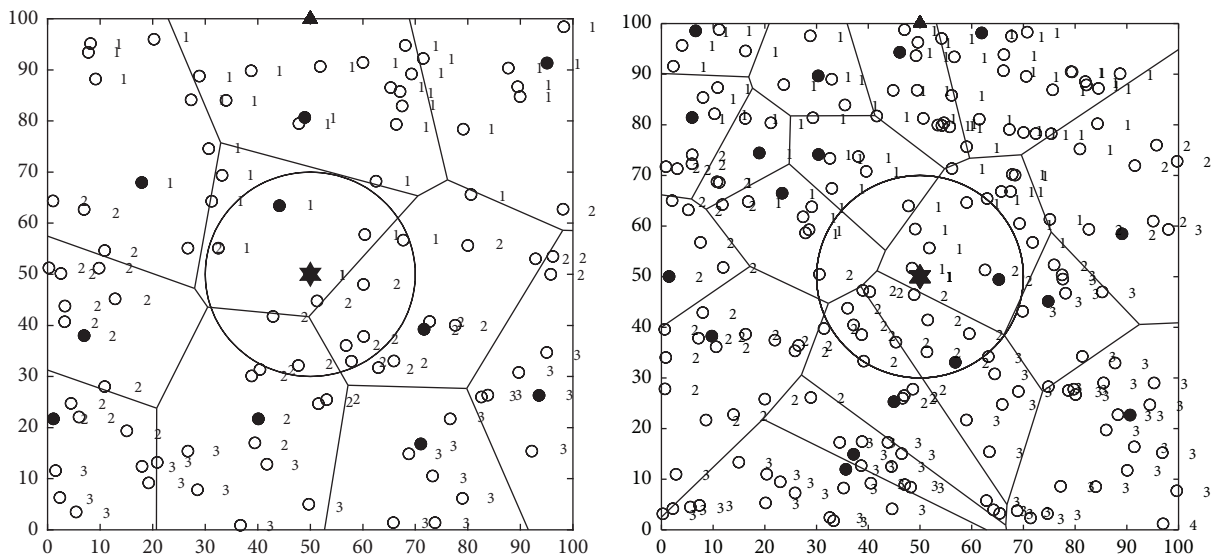


Figure 12: Overall processes of the proposed architecture.

Table 4: Simulation parameters.

Parameter	Symbol	Value
Network size	A	300 m × 300 m
Number of sensor nodes	N	100, 200
Number of camera nodes	-	1
Percentage of communication clusters	-	20%
Initial normal node energy	E_{init}	10 J
Initial camera node energy	E_{cam_init}	15 J
E_{Tx} and E_{Rx}	E_{elec}	50 nJ/bit
Free space	ε_{fs}	10 pJ/bit/m2
Multipath fading	ε_{fs}	0.0013 pJ/bit/m4
Energy for aggregation	E_{DA}	5 nJ/bit/message
Data packet size [41]	l_d	128 bytes
Header size	l_h	8 bytes
Payload size	l_p	120 bytes
Control packet size	l_c	25 bytes
Communication range	R_c	75 m
Image size (Lenna grayscale) [42]	-	512 × 512 pixels
DWT level	-	5
Quantization level	-	10



(a) Hierarchical routing of a sparse network

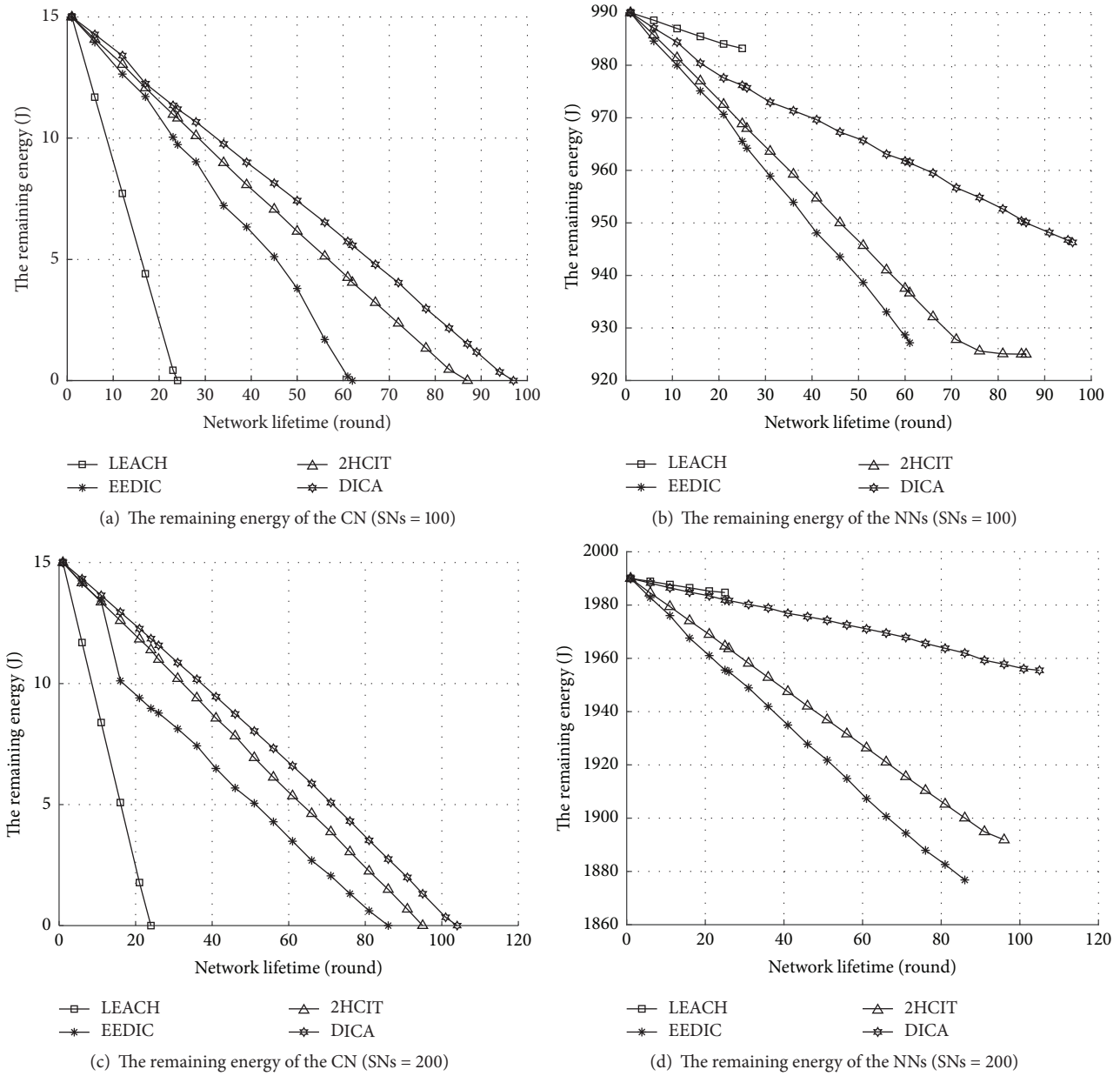
(b) Hierarchical routing of a dense network

Figure 13: Examples of hierarchical routing in sparse and dense networks.

5.2.2. Network Remaining Energy Evaluation. The remaining energies of both the CN and NNs with respect to the number of rounds are shown in a set of subfigures (Figure 14). Our proposed method outperforms the LEACH, EEDIC, and 2HCIT approaches in terms of balancing the network energy consumption and prolonging the network lifetime. In Figure 14, as the number of rounds increases, our proposed method outperforms the other methods. The energies of both the CN and NNs were reduced slightly in our method, followed by those in the EEDIC and 2HCIT approaches, while in LEACH approach, the remaining energy of the SNs decreased dramatically, especially that of the CN.

These results indicate that our proposed method could distribute the compression and transmission tasks among the SNs and better balance the energy consumption of the SNs in the network compared to the other two approaches. Moreover, our multihop hierarchical routing algorithm performs well by reducing the transmission energies of the SNs. In both sparse and dense networks, our proposed method can yield SN energy savings of approximately 80%, 40%, and 10% compared to LEACH, EEDIC, and 2HCIT, respectively.

5.2.3. Network Lifetime Evaluation. In WMSNs, the entire network lifetime evaluation is important for proving



(a) The remaining energy of the CN (SNs = 100)

(b) The remaining energy of the NNs (SNs = 100)

(c) The remaining energy of the CN (SNs = 200)

(d) The remaining energy of the NNs (SNs = 200)

Figure 14: Remaining energies of the CN and the NNs.

energy-efficient performance. Figure 15 provides an insight into the network lifetime in terms of the number of rounds with respect to the percentage of energy that remains in the CN. The results indicate that our proposed method can enhance the network lifetime for both sparse and dense SNs. As the number of NNs increases, our algorithm still performs well. Our proposed method improves the network lifetime by approximately 80% over LEACH. Comparing to 2HCIT and EEDIC, our method improves the network lifetime by approximately 20%–40% in both networks.

5.2.4. Throughput Evaluation. In this subsection, we discuss the throughput of an overall network, which is one of the most important measures of network performance. Throughput is defined as the number of sequential packets that contain a full (complete) image that are sent to the BS successfully. As shown in Table 5, our proposed algorithm archives much better throughput than the other algorithms since our algorithm relies on minimizing the CN energy consumption by distributing the compression task and NN energy by (preferred) transmission over a short distance hop by hop to the BS. In both spare and dense networks, on average, our proposed method achieves 10%, 70%, and 80% higher throughput compared to 2HCIT, EDDIC, and LEACH, respectively.

6. Conclusions

A distributed image compression architecture (DICA) over WMSNs is proposed in this paper. Three main contributions

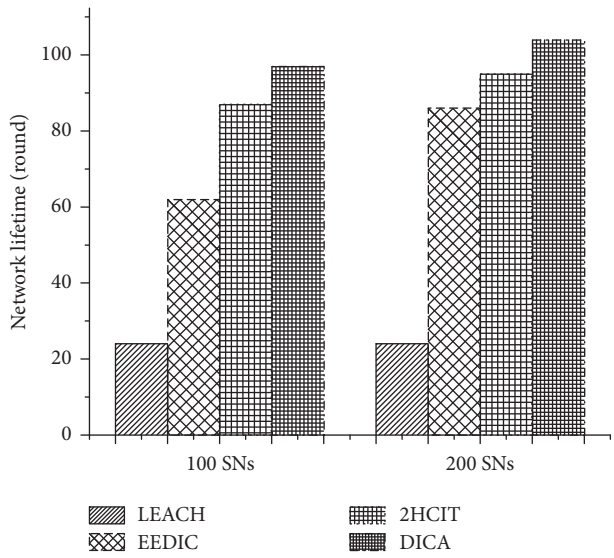


Figure 15: Network lifetimes of 100 and 200 SNs.

Table 5: The throughput results (# of complete images).

Number of SNs	LEACH	EEDIC	2HCIT	DICA
100	20	28	85	96
200	22	35	95	102

have been discussed. First, the optimal camera cluster is determined by using FLS. Second, image compression tasks are distributed among SNs, which are close to each other within the image compression cluster to save energy. Lastly, multihop hierarchical routing has been developed to preserve and balance energy in the network. According to simulation experiments, our algorithm can improve the energy consumption efficiency, on average, by 10%, 40%, and 80% compared to 2HCIT, EEDIC, and LEACH, respectively, thereby prolonging the network lifetime and increasing the throughput compared to the above-discussed algorithms for both sparse and dense networks.

Even though DICA can achieve a high degree of energy efficiency and throughput, more investigations, assumptions, and constraints could be further explored, such as quality-of-service awareness mechanisms and sophisticated data compression and aggregation techniques. With respect to other factors that likely affect the high volume of data transmission, comprehensive simulation and analysis could be intensively investigated, such as a scalability consideration, that is, network density and diversity, network dimension, routing selection, mobility, irregular topology, various signal propagation models, and heterogeneous data traffic considering additional transmission protocol overheads. In future work, we will focus on the modification of the image compression technique to reduce its computational complexity, memory usage, and energy consumption so that image compression can be better performed in resource-constrained wireless sensor networks. Furthermore, we plan to extend our proposed scheme to video compression and transmission over WMSNs.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by grants from Khon Kaen University via ASEAN and GMS Countries' Personnel Programs 2016–2019 and an Interdisciplinary Grant (CSKKU2559) from the Department of Computer Science, Khon Kaen University.

References

[1] S. A. Alvi, B. Afzal, G. A. Shah, L. Atzori, and W. Mahmood, "Internet of multimedia things: vision and challenges," *Ad Hoc Networks*, vol. 33, pp. 87–111, 2015.

[2] S. Agrawal and M. L. Das, "Internet of things—a paradigm shift of future Internet applications," in *Proceedings of the Nirma University International Conference on Engineering (NUiCONE '11)*, pp. 1–7, IEEE, Ahmedabad, India, December 2011.

[3] T. G. Nguyen, C. So-In, N. G. Nguyen, and S. Phoemphon, "A novel energy-efficient clustering protocol with area coverage awareness for wireless sensor networks," *Peer-to-Peer Networking and Applications*, vol. 10, no. 3, pp. 519–536, 2017.

[4] W. Dong et al., "QoS routing algorithm for wireless multimedia sensor networks, in advances in computation and intelligence," in *Proceedings of the 4th International Symposium, ISICA 2009 Huangshi*, Z. Cai and et al., Eds., vol. 5821, pp. 517–524, Springer, Berlin, Germany.

[5] K. Chang, C. Chen, J. Chen, and H. Chao, "Internet of things and cloud computing for future internet, in security-enriched urban computing and smart grid," in *Proceedings of the Second International Conference (SUComS '11)*, R.-S. Chang, T.-h. Kim, and S.-L. Peng, Eds., vol. 223, pp. 1–10, Springer Berlin Heidelberg, Berlin, Germany.

[6] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless micro-sensor networks," in *Proceedings of the 33rd Annual Hawaii International Conference on System Siences (HICSS '00)*, January 2000.

[7] R. Mahakud, S. Rath, M. Samantaray et al., "Energy Management in Wireless Sensor Network Using PEGASIS," in *Proceedings of the 2nd International Conference on Intelligent Computing, Communication and Convergence, ICCC 2016*, pp. 207–212, India, January 2016.

[8] K. Jung, J.-Y. Lee, and H.-Y. Jeong, "Improving adaptive cluster head selection of teen protocol using fuzzy logic for WMSN," *Multimedia Tools and Applications*, pp. 1–16, 2016.

[9] H. Wang, Q. Liu, and M. Liu, "NHEED: An energy-efficient multi-hop routing protocol based on HEED," in *Proceedings of the 12th World Congress on Intelligent Control and Automation, WCICA 2016*, pp. 3219–3225, China, June 2016.

[10] V. K. Arora, V. Sharma, and M. Sachdeva, "A survey on LEACH and other's routing protocols in wireless sensor network," *Optik - International Journal for Light and Electron Optics*, vol. 127, no. 16, pp. 6590–6600, 2016.

[11] M. Macit, V. C. Gungor, and G. Tuna, "Comparison of QoS-aware single-path vs. multi-path routing protocols for image transmission in wireless multimedia sensor networks," *Ad Hoc Networks*, vol. 19, pp. 132–141, 2014.

[12] H. Zaineldin, M. A. Elhosseini, and H. A. Ali, "Image compression algorithms in wireless multimedia sensor networks: A survey," *Ain Shams Engineering Journal*, vol. 6, no. 2, pp. 481–490, 2015.

[13] T. Sheltami, M. Musaddiq, and E. Shakshuki, "Data compression techniques in Wireless Sensor Networks," *Future Generation Computer Systems*, vol. 64, pp. 151–162, 2016.

[14] H. Wu and A. A. Abouzeid, "Energy efficient distributed image compression in resource-constrained multihop wireless networks," *Computer Communications*, vol. 28, no. 14, pp. 1658–1668, 2005.

[15] F. Tian, J. Liu, E. Sun, and C. Wang, "An energy efficient and load balancing distributed image compression algorithm in WMSNs," in *Proceedings of the 2011 International Conference on Advanced in Control Engineering and Information Science, CEIS 2011*, pp. 3421–3427, China, August 2011.

[16] M. Nasri, A. Helali, H. Sghaier, and H. Maaref, "Trade-off analysis of energy consumption and image quality for multihop wireless sensor networks," *International Journal of Wireless Information Networks*, vol. 19, no. 3, pp. 254–269, 2012.

[17] S.-K. Jo, M. Ikram, I. Jung, W. Ryu, and J. Kim, "Power efficient clustering for wireless multimedia sensor network," *International Journal of Distributed Sensor Networks*, vol. 2014, Article ID 148595, 9 pages, 2014.

[18] J. Luo, D. Wu, C. Pan, and J. Zha, "Optimal Energy Strategy for Node Selection and Data Relay in WSN-based IoT," *Mobile Networks and Applications*, vol. 20, no. 2, pp. 169–180, 2015.

[19] Z. Zuo, Q. Lu, and W. Luo, "A two-hop clustered image transmission scheme for maximizing network lifetime in wireless multimedia sensor networks," *Computer Communications*, vol. 35, no. 1, pp. 100–108, 2012.

[20] C. So-In et al., "Performance Evaluation of LEACH on Cluster Head Selection Techniques in Wireless Sensor Networks," in *Proceedings of the in The 9th International Conference on Computing and InformationTechnology (IC2IT2013): 9th-10th*, P. Meesad, H. Unger, and S. Boonkrong, Eds., pp. 51–61, Springer Berlin Heidelberg, Berlin, Germany, May 2013.

[21] T. Ma et al., "A survey of energy-efficient compression and communication techniques for multimedia in resource constrained systems," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 3, pp. 963–972, 2013.

[22] S. K. Singh, P. Kumar, and J. P. Singh, "A Survey on Successors of LEACH Protocol," *IEEE Access*, vol. 5, pp. 4298–4328, 2017.

[23] D.-U. Lee, H. Kim, M. Rahimi, D. Estrin, and J. D. Villasenor, "Energy-efficient image compression for resource-constrained platforms," *IEEE Transactions on Image Processing*, vol. 18, no. 9, pp. 2100–2113, 2009.

[24] M. Nasri, A. Helali, H. Sghaier, and H. Maaref, "Efficient JPEG 2000 image compression scheme for multihop wireless networks," *TELKOMNIKA (Telecommunication, Computing, Electronics and Control)*, vol. 9, 2011.

[25] Q. Lu, X. Ye, and L. Du, "An architecture for energy efficient image transmission in WSNs," in *Proceedings of the International Conference on Networks Security, Wireless Communications and Trusted Computing, NSWCTC 2009*, pp. 296–299, China, April 2009.

[26] R. Logambigai and A. Kannan, "Fuzzy logic based unequal clustering for wireless sensor networks," *Wireless Networks*, vol. 22, no. 3, pp. 945–957, 2016.

[27] G. Brante, G. De Santi Peron, R. D. Souza, and T. Abrao, "Distributed fuzzy logic-based relay selection algorithm for cooperative wireless sensor networks," *IEEE Sensors Journal*, vol. 13, no. 11, pp. 4375–4386, 2013.

[28] H. Bagci and A. Yazici, "An energy aware fuzzy approach to unequal clustering in wireless sensor networks," *Applied Soft Computing*, vol. 13, no. 4, pp. 1741–1749, 2013.

[29] A. De San Bernabe, J. R. Martinez-de Dios, and A. Ollero, "Efficient integration of RSSI for tracking using Wireless Camera Networks," *Information Fusion*, vol. 36, pp. 296–312, 2017.

[30] D. S. Taubman and M. W. Marcellin, *JPEG 2000: Image Compression Fundamentals, Standards and Practice*, vol. 776, Kluwer Academic Publishers, 2002.

[31] L. A. Zadeh, "Fuzzy sets," *Information and Computation*, vol. 8, no. 3, pp. 338–353, 1965.

[32] B. Balakrishnan and S. Balachandran, "FLECH: fuzzy logic based energy efficient clustering hierarchy for nonuniform wireless sensor networks," *Wireless Communications and Mobile Computing*, vol. 2017, 13 pages, 2017.

[33] I. S. AlShawi, L. Yan, W. Pan, and B. Luo, "Lifetime enhancement in wireless sensor networks using fuzzy approach and A-star algorithm," in *Proceedings of the IET Conference on Wireless Sensor Systems, WSS 2012*, UK, June 2012.

[34] J. M. Mendel, "Fuzzy logic systems for engineering: a tutorial," *Proceedings of the IEEE*, vol. 83, no. 3, pp. 345–377, 1995.

[35] E. H. Mamdani, "Application of fuzzy logic to approximate reasoning using linguistic synthesis," *Fuzzy Sets and Systems*, vol. 26, no. 12, pp. 1182–1191, 1977.

[36] A. Alomari et al., "Dynamic fuzzy-logic based path planning for mobility-assisted localization in wireless sensor networks," *Sensors*, vol. 17, no. 8, p. 1904, 2017.

[37] C. Christopoulos, A. Skodras, and T. Ebrahimi, "The JPEG2000 still image coding system: an overview," *IEEE Transactions on Consumer Electronics*, vol. 46, no. 4, pp. 1103–1127, 2002.

[38] T. Xiang, C. Yu, and F. Chen, "Secure MQ coder: An efficient way to protect JPEG 2000 images in wireless multimedia sensor networks," *Signal Processing: Image Communication*, vol. 29, no. 9, pp. 1015–1027, 2014.

[39] Wicaz Wireless Measurement System. Available from: http://www.openautomation.net/uploadsproductos/micaz_datasheet.pdf.

[40] T. G. Nguyen, C. So-In, and N. G. Nguyen, "Radio irregularity obstacles-aware model for wireless sensor networks," *Journal of Telecommunication, Electronic and Computer Engineering*, vol. 8, no. 3, pp. 121–126, 2016.

[41] S. Kurt, H. U. Yildiz, M. Yigit, B. Tavli, and V. C. Gungor, "Packet Size Optimization in Wireless Sensor Networks for Smart Grid Applications," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 3, pp. 2392–2401, 2017.

[42] Lenna. Available from: http://www.ece.rice.edu/~wakin/images/lena512.bmp.

Hindawi
Wireless Communications and Mobile Computing
Volume 2017, Article ID 9823684, 10 pages
https://doi.org/10.1155/2017/9823684

Research Article

An Adaptive Joint Sparsity Recovery for Compressive Sensing Based EEG System

Hamza Djelouat, Hamza Baali, Abbes Amira, and Faycal Bensaali

College of Engineering, Qatar University, P.O. Box 2713, Doha, Qatar

Correspondence should be addressed to Hamza Djelouat; hamza.djelouat@qu.edu.qa

Received 28 July 2017; Revised 14 October 2017; Accepted 2 November 2017; Published 29 November 2017

Academic Editor: Gonzalo Vazquez-Vilar

The last decade has witnessed tremendous efforts to shape the Internet of things (IoT) platforms to be well suited for healthcare applications. These platforms are comprised of a network of wireless sensors to monitor several physical and physiological quantities. For instance, long-term monitoring of brain activities using wearable electroencephalogram (EEG) sensors is widely exploited in the clinical diagnosis of epileptic seizures and sleeping disorders. However, the deployment of such platforms is challenged by the high power consumption and system complexity. Energy efficiency can be achieved by exploring efficient compression techniques such as compressive sensing (CS). CS is an emerging theory that enables a compressed acquisition using well-designed sensing matrices. Moreover, system complexity can be optimized by using hardware friendly structured sensing matrices. This paper quantifies the performance of a CS-based multichannel EEG monitoring. In addition, the paper exploits the joint sparsity of multichannel EEG using subspace pursuit (SP) algorithm as well as a designed sparsifying basis in order to improve the reconstruction quality. Furthermore, the paper proposes a modification to the SP algorithm based on an adaptive selection approach to further improve the performance in terms of reconstruction quality, execution time, and the robustness of the recovery process.

1. Introduction

Nowadays, a huge interest has been dedicated to the development of Internet of things (IoT) based connected health platforms. These platforms are empowered by several wearable battery-driven sensors that collect and record different vital signs for a long period. The collected data is sent using low-power communication protocols to a nearby gateway. The gateway then delivers the data to the host cloud. At the cloud level, various signal processing and data analysis techniques are performed to provide computer-aided medical assistance. However, the performance of these platforms is bottlenecked mainly by the limited lifespan of wearable sensors. Therefore, exploring data compression techniques can reduce the number of the data transmitted from the sensors to the gateway, hence prolonging the sensor's lifespan. Compressive sensing (CS) theory has proved to be a reliable compression technique which provides the best trade-off between reconstruction quality and low-power consumption compared to conventional compression approaches such as transform coding or segmentation and labeling techniques [1].

CS is a novel data sampling paradigm that merges the acquisition and the compression processes into one operation. CS relies on the signal sparsity/compressibility in order to acquire a compressed form of the signal while maintaining its salient information. CS has been introduced in [2, 3]; the authors have proved that any sparse signal can be recovered exactly from a smaller set of measurements than its original dimension. Therefore, it is possible to acquire sparse signals by taking far fewer random measurements than what the famous Shannon-Nyquist theorem states using well-designed matrices. Despite the fact that CS is a relatively new theory, it has been incorporated in a wide range of emerging applications including image processing, radar, wireless communication, and monitoring-based applications.

Furthermore, to cope with current monitoring systems, an extension to CS has been introduced in [4], namely, distributed compressive sensing (DCS). DCS aims to exploit both the signal intrastructure (the sparsity) and the interstructure of the acquisition system (correlation between the measurements of the different sensors) in order to acquire the

information about the signals of interest using the minimum number of measurements.

Subsequently, by leveraging the sparsity of most biosignals such as electrocardiogram (ECG) and electromyogram (EMG) [5, 6], many efforts have been dedicated to exploit CS and DCS in wireless body area network (WBAN) applications to enable CS-based IoT platforms for connected health. In such platforms, the compressed data is transmitted to a fusion node (gateway) that possesses enough computing and communication abilities. Afterwards, the data is routed to the cloud via the Internet for reconstruction, processing, and analysis. It is worth mentioning that data reconstruction can be performed on the gateway to empower an "IoT-based edge computing platform."

CS-based systems for EMG and ECG monitoring have been thoroughly investigated, where various aspects have been well analyzed, for instance, the comparison between CS and state-of-the-art compression techniques [7], the system design considerations [5], the effect of the sparsifying dictionaries [8], and the best algorithms in terms of quality of reconstruction [9]. In addition, authors in [10] further leveraged the biosignals structure, where, instead of only exploring the signal sparsity in one domain, the authors proposed using all the available structure such as low rank, piecewise smoothness, and the sparsity in more than one domain. Subsequently, the authors in [10] proposed a reconstruction framework that aims to exploit any a priori information about the signals in order to enhance the reconstruction quality.

Moreover, the application of CS in electroencephalogram (EEG) signals has been presented in the literature. The performance of such CS-based systems is controlled by two parameters, the sparsity of the signal, which depends mainly on the sparsifying basis, and the appropriate recovery algorithm adopted. Authors in [11] have shown the possibility of using CS for EEG compression as long as the EEG signal is recorded at least via 22 channels. The major limitation facing the deployment of CS in EEG compression is that it is very hard to find the transform domain where the EEG exhibits a sparse behavior. Therefore, different classes of sparsifying basis and dictionaries have been investigated to determine the best basis that provides the sparsest representation for EEG. Senay et al. have quantified the use of Slepian basis as sparsifying basis for EEG [12]; the obtained result shows a low error rate for the reconstructed EEG signal. In addition, Aviyente has presented a CS-based EEG compression system exploiting Gabor frame as dictionary for EEG signals [13], whereas Gangopadhyay et al. [14] have found that adopting a wavelet transform for EEG is more efficient in terms of quality of reconstruction. Zhang et al. presented in [15] a block sparse Bayesian learning (BSBL) approach to recover EEG raw data that enable both good reconstruction quality and low system complexity by using sparse sensing matrices and wavelet matrices as the sparsifying basis. More recently, authors in [16] introduced an optimization model based on ℓ_0 norm to enhance the cosparsity and to enforce the low-rank structure of the EEG signal. The authors proposed using a second-order difference matrix as the sparsifying dictionary to enhance the sparsity of the EEG signal as well as exploit the collaboration between the cosparsity and the low-rank structure to recover simultaneously a multichannel EEG signal.

Besides selecting the optimum sparsifying matrix, adopting the appropriate reconstruction algorithm plays an important role in the recovery of the EEG data. Greedy algorithms have been widely explored in CS applications due to their low complexity and their superior performance compared to other recovery algorithms, such as convex relaxation approaches. The widely used greedy algorithms are orthogonal matching pursuits (OMP) [17], stage-wise OMP (StOMP) [18], compressive sampling matching pursuit (CoSaMP) [19], and subspace pursuit (SP) [20].

The main task in greedy algorithms is to identify the locations of the largest coefficients in the estimated signal. Greedy algorithms adopt a signal proxy approach at each iteration to identify these locations. If the sensing matrix satisfies the restricted isometry property condition [21], then the signal proxy is very similar to the original signal and the locations of the nonzero elements can be easily identified. OMP and StOMP reconstruct the signal in an iterative approach by locating the largest coefficient at each time. On the other hand, SP and CoSaMP select more than one coefficient at each iteration which allows them to converge to the solution with a lower number of iterations. However, SP and CoSaMP require information about the signal sparsity which is not available a priori in many applications as the sparsity of the signal often changes over time. Moreover, the sparsity parameter k depends not only on the signal structure but also on the space where the data is sparse; hence, the same signal can exhibit different levels of sparsity depending on the sparsifying basis. The required knowledge of sparsity estimate parameter k presents a critical issue with the SP and CoSaMP, where a poor choice of k can remarkably degrade the reconstruction quality. Adaptive sparsity algorithms have been proposed in the literature; authors in [22, 23] have performed various modification to OMP, CoSaMP, and SP algorithms in order to provide an adaptive framework that estimates the best value for the sparsity parameter k. Sparsity adaptive matching pursuit (SAMP) proposed in [22] is considered as a generalization of both OMP and SP by updating k at each iteration until a certain condition is satisfied. The SAMP increases the value of sparsity parameter k using a two-stage verification process until the difference between the norms of the residual for every two successive iterations is below a certain threshold.

In this paper, a CS-based scheme for EEG signal compression and recovery is presented. The contributions of the paper are as follows:

(i) Joint channel reconstruction using SP algorithm is presented. The proposed approach renders a better reconstruction quality than the conventional channel-per-channel recovery.

(ii) The concept of concatenated basis as the sparsifying basis for EEG signals is explored to tackle the problem of the nonsparsity, and the concatenated basis consists of a random selection of elements from both discrete cosine transform matrix (DCT) and discrete wavelet transform matrix (DWT).

(iii) A new adaptive approach is presented to reconstruct the EEG signal. The new algorithm is a modification of the SP algorithm to provide an algorithm that does not require the knowledge of the sparsity of the signal a priori. The new proposed dynamic selection subspace pursuit (DSSP) algorithm performs an adaptive selection at each iteration for the coefficients that capture most of the signal energy. The proposed algorithm promotes two improvements over SP: first, an enhancement of the data reconstruction quality and, second, an increased robustness compared with SP, as the latter would provide a bad reconstruction quality if the sparsity parameter is poorly estimated.

The rest of the paper is organized as follows: CS fundamentals are briefly presented in Section 2. Section 3 addresses the main issue of the paper where the description of joint reconstruction approach and the proposed recovery algorithm is provided. Simulation results and discussion are presented in Section 4. Section 5 concludes the paper.

2. Compressed Sensing

2.1. Acquisition Model. The acquisition model of CS (1) is represented by an inner product between the input sparse signal $\mathbf{x} \in \mathbb{R}^{N\times 1}$ and the sensing matrix $\mathbf{\Phi} \in \mathbb{R}^{M\times N}$ (such that $m < N$) to generate the compressed measured signal $\mathbf{y} \in \mathbb{R}^{M\times 1}$.

$$\mathbf{y} = \mathbf{\Phi x}. \tag{1}$$

In most cases, the input signal $\mathbf{x}$ is not sparse in time domain, yet it can exhibit a sparse behavior under the appropriate transform. Thus, given a set $\{\Psi_i\}_{i=1}^{N}$ that spans $\mathbb{R}$, $\mathbf{x}$ can be expressed as a linear combination between the elements of $\mathbf{\Psi}$ with a vector $\mathbf{s} \in \mathbb{R}^N$ such that $\mathbf{x} = \sum_{i=1}^{N} \boldsymbol{\psi}_i s_i$. The input signal $\mathbf{x}$ is said to be k-sparse if $\mathbf{s}$ has only $k \ll N$ nonzero entries. The set of the indices corresponding to the positions of the nonzero entries of $\mathbf{s}$ is called the support of $\mathbf{s}$ and denoted as Σ_k.

The sensing matrix $\mathbf{\Phi} \in \mathbb{R}^{M\times N}$ which maps the N-length input signal $\mathbf{x}$ to an M-length signal $\mathbf{y}$ has to enable a small number of samples to acquire the salient information in the input signal. Moreover, it should allow acceptable reconstruction quality. Therefore, $\mathbf{\Phi}$ has to satisfy two conditions on the RIP and should be incoherent with sparsifying matrix $\mathbf{\Psi}$ [21].

2.2. Reconstruction Algorithms. Data reconstruction is the crucial task in any CS-based system. Thus, several approaches to recover the original signal $\mathbf{x}$ from the measured signal $\mathbf{y}$ have been proposed in literature. However, there are two main classes of reconstruction algorithms that have been widely explored, namely, convex optimization and greedy algorithms. Convex optimization approaches provide the exact solution if the input signal is completely sparse. Convex optimization algorithms are based on the $\ell 1$ minimization operation; for instance, basis pursuit (BP) algorithm [24] considers the following solution:

$$\begin{aligned} \widehat{\mathbf{x}} &= \arg\min \quad \|\mathbf{x}\|_1 \\ &\text{subject to} \quad \mathbf{y} = \mathbf{\Phi x}. \end{aligned} \tag{2}$$

In the case where the acquisition process is contaminated with noise, two different techniques can be deployed; first, if the noise level is known a priori, basis pursuit denoising (BPDN) [25] can be applied. However, if there is no knowledge about the noise level, least absolute shrinkage and selection operator (LASSO) presents an efficient approach to recover the original signal.

Greedy algorithms provide a suboptimal recovery for sparse signals, yet they outperform convex optimization approaches in the case where the signal of interest is highly sparse [17]. Greedy algorithms solve (1) iteratively by taking locally optimal decisions. These algorithms aim to find the locations of the nonzero coefficients to enable a fast recovery. Greedy algorithms include several variants such as gradient pursuit, matching pursuit (MP) [26], and OMP [17]. OMP offers a fast recovery compared to convex optimization approaches; however, it suffers from bad recovery quality for signals with a low degree of sparsity. Thus, several improved versions of OMP have been proposed, such as CoSaMP [19], SP [20], and StOMP [18].

2.3. Distributed Compressive Sensing. Conventional CS exploits only the sparsity of the data. However, if the same data is collected using different sensing nodes or different channels, their measurements would be highly correlated. In such scenario, the measurements exhibit the same behavior, such as being sparse in a particular domain.

Therefore, in a multichannel CS-based data acquisition system, each sensing node collects and compresses its data individually without taking any considerations about the other nodes. For the recovery, two approaches can be considered to reconstruct the data; first, data reconstruction can be performed on each sensing node individually, this approach ignores the dependency between the measurements of different sensors, and, hence, the quality of the reconstruction depends only on the sparsity of each recording. The second approach exploits the collaboration between all measurements to obtain more information about the data; thus, a better reconstruction quality can be achieved. This process is called joint measurement setting and it has motivated the introduction of the DCS concept.

DCS presents a new distributed coding framework that exploits both the sparsity of the signal and the correlation between the different signals in multisensing architectures. In the DCS acquisition stage, each sensor collects its measurements by taking random projections of the signal without any consideration about the states of the other sensors in the network. However, the reconstruction phase exploits the intersignal correlation by using all of the obtained measurements to recover all the signals simultaneously.

3. CS-Based EEG Compression

EEG is a well-considered framework to measure the electrical activity of the brain; EEG signals are widely used to detect different types of neurological disorders such as comas, epilepsy, and sleep disorders. Moreover, EEG can also be used for nonmedical applications such as brain-computer interface. EEG signals are recorded over a long period of time using a set

of electrodes placed over the head of the subject. EEG signals are considered as a multivariate signal acquired via multiple channels which results in the generation of big EEG data that need to be stored and transmitted. However, several studies have highlighted the limitation of such approach in terms of high energy consumption due to massive raw data streaming. Thus, EEG monitoring platforms would benefit from more power efficient sampling and compression prior to wireless transmission. These limitations motivate the incorporation of CS and DCS to the EEG acquisition and compression.

3.1. Related Work. CS-based EEG monitoring has been investigated in the literature. First, the feasibility of applying CS to EEG acquisition has been addressed in [11, 27]. The authors quantified CS-based EEG monitoring, where CS has been used as a compression technique to reduce both the storage and the processing load. The obtained results revealed that CS does not provide a good reconstruction quality unless an appropriate acquisition scheme is deployed. CS can be applied only if at least 22 channels are deployed to collect the EEG data.

The low sparsity of the EEG raw data in both time and frequency domains presents the main challenge in the design of CS-based EEG monitoring systems. Thus, a great attention was dedicated to providing dictionaries and basis that render a high sparse representation of EEG signals. Subsequently, several dictionaries have been investigated in the literature such as Slepian basis, Gabor frames, and DWT matrices [12–15, 28]. In [12], Senay et al. quantified a CS framework for EEG compression using Slepian functions as a sparsifying dictionary. By projecting the EEG signal into the Slepian basis, a sparse representation is achieved; hence, the EEG can be efficiently compressed with CS at a very low error rate. In addition, Aviyente analyzed a CS framework for EEG compression in terms of the mean square error (MSE) using Gabor frame method as sparsifying basis [13]. The author argued that chirped Gabor dictionary would be very efficient and it can increase the sparsity of the signals; hence, it improves the performance of CS-based EEG monitoring. On the other hand, Gangopadhyay et al. claimed in [14] that wavelet-based dictionaries are more suitable for CS-based EEG compression than the previously mentioned approaches. Author in [11] have provided a detailed performance study for six different sparsifying dictionaries, namely, Gabor, Mexican Hat, cubic Spline, linear Spline, cubic B-Spline, and linear B-Spline. In the paper, intensive sets of simulations were carried out for different reconstruction algorithms in 18 different test conditions. The B-Spline dictionaries proved to be the most promising, yielding best reconstruction quality and achieving the lowest error rates. Furthermore, Liu et al. proposed in [16] a new framework for EEG monitoring based on sparse signal recovery method and simultaneous cosparsity and low-rank (SCLR) optimization approaches. The proposed approach utilizes second-order difference matrix as the sparsifying basis and $\ell 0$ minimization for data reconstruction. Nevertheless, Zhang et al. explored the BSBL which was initially developed in [29] to empower ECG signal monitoring for EEG reconstruction [15]. The idea of the paper is that, instead of finding the optimal sparsifying dictionary, the authors used general dictionary matrices (DWT and DCT) to represent the EEG signal. Yet, they explored BSBL to take advantage of the block structure of the EEG signal. The results revealed an acceptable reconstruction quality for particular sets of applications.

Besides evaluating the sparsity, the metrics for EEG data reconstruction have been investigated as well. For instance, root-mean-squared difference (PRD) has been used to evaluate the reconstruction quality of EEG reconstruction in [30]. However, different thresholds have been established based on the targeted application. In [31], based on energy preservation criterion, authors determined that the maximum PRD which provides an acceptable recovery is 7%. Such PRD value can guarantee that 99.5% of the signal energy is persevered, whereas Higgins et al. demonstrated in [32] that up to 30% PRD is tolerable with EEG compression for applications of automated seizure detection.

Table 1 expatiates on the comparative results between several works presented in the literature on the integration of CS in context of EEG monitoring.

3.2. Joint Channel Reconstruction. In a multichannel acquisition scenario, the joint sparsity of the signal ensemble is often smaller than the aggregate over individual signals sparsities [4]. Therefore, if the signal of interest is characterized by a weak sparsity, such as EEG signals, it is possible to explore the joint sparsity of the ensemble. In addition, exploiting joint sparsity would result in a remarkable reduction in the number of measurements required to achieve an acceptable recovery.

The corresponding CS model CS-based EEG compression system for the J channels can be written as follows:

$$\mathbf{Y} = \mathbf{\Phi}\mathbf{X}, \tag{3}$$

where $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_J] \in \mathbb{R}^{N\times J}$ denotes the original EEG raw data and N is the number of samples in each channel, whereas $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_J] \in \mathbb{R}^{M\times J}$ denotes the compressed form of the EEG data.

The proposed method represents the multichannel signal data by stacking, column-wise, the measurement vectors of the different channels into a single vector. This approach allows the recovery of a multiple channel recording simultaneously by exploring their joint sparsity in order to reduce the number of the required measurements for each channel as well as achieve a better reconstruction quality than the conventional CS recovery which recovers each of the channel measurements individually.

The joint-measurements acquisition model can be expressed as

$$[\mathbf{Y}]_j = \mathbf{\Phi}\,[\mathbf{X}]_j \quad j = 1, 2, \ldots, l, \tag{4}$$

where l is an integer number such that $l = J/\Lambda$ and Λ denotes the number of channels to be recovered simultaneously. The vector $[\mathbf{X}]_j \in \mathbb{R}^{\Lambda\times N}$ is the column-wise stacking of the elements of $\mathbf{X}$ such that $[\mathbf{X}]_j = [\mathbf{x}^T_{(j-1)\Lambda+1}, \mathbf{x}^T_{(j-1)\Lambda+2}, \ldots, \mathbf{x}^T_{(j-1)\Lambda+\Lambda}]^T$.

For EEG data recovery, each $[\mathbf{X}]_j$ is reconstructed individually by applying the SP algorithm. The pseudocode for SP algorithm is presented in Algorithm 1.

Table 1: Summary of CS-based EEG monitoring approaches presented in the literature.

References	Sparsifying basis	Reconstruction algorithm	CR	PRD	MSE
Abdulghani et al. [11]	B-Spline	BP	0.33	18.61	—
Casson and Rodriguez-Villegas [33]	DWT	BP	0.4	25	—
Zhang et al. [15]	DWT	BSBL	0.5	—	0.078
Liu et al. [16]	Second-order difference matrix	$\ell 0$	0.5	—	0.045

Input:
$[\mathbf{Y}]_j$: measurement vector
$\mathbf{\Phi}$: sensing matrix
ϵ: stopping criterion
k: sparsity estimate
Initialization:
$\Omega = [\,]$
$\mathbf{r}^{[0]} = [\mathbf{Y}]_j$
current iteration $i = 0$
Procedure: **While** $\|\mathbf{r}^{[i]} - \mathbf{r}^{[i-1]}\|_2 \geq \epsilon$ **and** $\|\mathbf{r}^{[i]}\|_2 \leq \|\mathbf{r}^{[i-1]}\|_2$
(1) $i = i + 1$
(2) $\mathbf{G}^{[i-1]} = \mathbf{\Phi}^* \mathbf{r}^{[i]}$
(3) $\Omega^{[i]} = \{k$ indices corresponding to the largest absolute value of $\mathbf{G}^{[i]}\}$
(4) $\Omega = \Omega \cup \Omega^{[i]}$
(5) $[\mathbf{X}]^{[i]}_{j\Omega} = \mathbf{\Phi}^{\dagger}_{\Omega}[\mathbf{Y}]_j$
(6) $\mathbf{r}^{[i]} = [\mathbf{Y}]_j - \mathbf{\Phi}_{\Omega}[\mathbf{X}]_j{}^{[i]}$
Output:
$\widehat{[\mathbf{X}]_j}$: reconstructed EEG signal that satisfies $\widehat{[\mathbf{X}]}_{j_{\{1,\ldots,N\}-\Omega}} = 0$ and $[\mathbf{X}]_{j\Omega}^{[i]} = \mathbf{\Phi}^{\dagger}_{\Omega}[\mathbf{Y}]_j$

Algorithm 1: Subspace pursuit algorithm.

3.3. Proposed DSSP Algorithm. Unlike SAMP, this paper proposes the DSSP algorithm, which follows a straightforward modification to the SP algorithm by exploring the key idea of CS. That is, if the signal is highly compressible, then only a small number of the coefficients capture most of the signal's energy. Subsequently, by locating these coefficients at each iteration, the true support of the signal (the locations of the largest nonzero coefficients) can be identified.

Moreover, as the signal structure often changes with time, its sparsity changes as well. Thus, to cope with this variation in the signal structure, the algorithm should exhibit a dynamic selection approach; that is, the number of selected coefficients at each iteration should be updated depending on the changes in the signal structure. Hence, the proposed DSSP algorithm follows an energy-based selection approach that updates the number of selected coefficients depending on the acquired signal at any given time.

Therefore, in the DSSP algorithm, rather than selecting a fixed number of coefficients at each iteration, the algorithm locates the indices of the minimum number of coefficients that capture the most of the signal's energy. This approach expands the true support of the signal stage by stage. In addition, DSSP refines the estimated signal at each iteration, where the columns of the sensing matrix corresponding to the locations of the selected coefficients are used to solve a least square problem. This process is repeated until an acceptable quality of reconstruction is achieved based on error minimization condition.

In order to select the appropriate number of coefficients that capture most of the signal proxy energy at each iteration, the parameter $0 < \lambda < 1$ is defined as the rate between the energy of the selected coefficient and the energy of the signal proxy at each iteration. Therefore, in order to select the appropriate set of coefficients that holds most of the signal proxy energy, the parameter λ can be selected as high as possible as long as it allows the selection of at least one coefficient at each iteration.

The pseudocode of the DSSP is listed in Algorithm 2.

4. Results and Discussion

In order to quantify the performance of CS-based scheme for EEG compression, intensive experiments have been carried out using MATLAB computing software. The EEG signals were taken from the database of the EEGLab [34]. The EEG database considered contains 80 channel recordings, each with 384 samples.

In order to set up the simulations, Symlets wavelet-based transform [35] is used as sparsifying basis for the EEG signal. Symlets have been selected after several empirical simulations to determine the best wavelet class that provides the best reconstruction quality.

Input:
$[\mathbf{Y}]_j$: measurement vector
$\mathbf{\Phi}$: sensing matrix
ϵ: stopping criterion
λ: Energy parameter, $0 < \lambda < 1$
$\Omega = [\,]$
$\mathbf{r}^{[0]} = [\mathbf{Y}]_j$
current iteration $i = 0$
Procedure: **While** $\|\mathbf{r}^{[i]} - \mathbf{r}^{[i-1]}\|_2 \geq \epsilon$
and $\|\mathbf{r}^{[i]}\|_2 \leq \|\mathbf{r}^{[i-1]}\|_2$
(1) $i = i + 1$
(2) Calculate the signal proxy $\mathbf{G}[n]^{[i-1]} = \mathbf{\Phi}^* \mathbf{r}^{[i]}$
(3) Calculate the energy of the signal proxy

$$E = \frac{1}{N}\sum_{n=1}^{N}|\mathbf{G}[n]|^2$$

(4) Sort $\mathbf{G}[n]^{[i-1]}$ in an ascending order
(5) $\Omega^{[i]}$ = minimum k indices corresponding to the largest absolute that satisfies

$$\mathbf{P}_{|\Omega} = \frac{1}{k}\sum_{n=1}^{k}|\mathbf{G}[n]|^2 \geq \lambda \mathbf{E}$$

(6) $\Omega = \Omega \cup \Omega^{[i]}$
(7) $[\mathbf{X}]_{j\Omega}^{[i]} = \mathbf{\Phi}_{\Omega}^{\dagger}[\mathbf{Y}]_j$
(8) $\mathbf{r}^{[i]} = [\mathbf{Y}]_j - \mathbf{\Phi}_{\Omega}[\mathbf{X}]_j^{[i]}$
Output:
$\widehat{[\mathbf{X}]_j}$: reconstructed EEG signal that satisfies $\widehat{[\mathbf{X}]}_{j\{1,\ldots,N\}-\Omega} = 0$ and $[\mathbf{X}]_{j\Omega}^{[i]} = \mathbf{\Phi}_{\Omega}^{\dagger}[\mathbf{Y}]_j$

Algorithm 2: Dynamic selection subspace pursuit (DSSP) algorithm.

In the design of the sensing matrix, two approaches have been adopted. In the first one, the entries of the sensing matrix are drawn from a normal distribution such that $\mathbf{\Phi} \sim (0, 1/M)$. Whereas, in the second one, the entries are drawn form a Bernoulli distribution with entries $\{1, -1\}$. The experiments results are averaged on 100 trials.

The signal reconstruction quality is evaluated in terms of the normalized mean square error (NMSE). NMSE calculates the 2-norm of the difference between the original ECG signal $[\mathbf{X}]_j$ and the reconstructed one $\widehat{[\mathbf{X}]_j}$ as follows:

$$\text{NMSE} = \frac{\left\|[\mathbf{X}]_j - \widehat{[\mathbf{X}]_j}\right\|_2^2}{\left\|[\mathbf{X}]_j\right\|_2^2}. \tag{5}$$

Furthermore, the term compression ratio (CR) is defined as the ratio between M, the number of samples in the compressed signal, and N, the number of samples in the original ECG signal:

$$\text{CR} = \frac{M}{N}. \tag{6}$$

First, the adopted algorithm for EEG signal reconstruction is SP. As shown in Algorithm 1, the SP algorithm requires the sparsity estimate as an input; however, there is no established rule to select the best sparsity parameter k. Thus, in order to determine the optimum value of k, intensive empirical simulations have been performed. The obtained results indicate that the optimal value of k for channel-per-channel reconstruction (EEG signal with 384 samples) is $k = 55 \pm 3$. Subsequently, for joint channel reconstruction the best sparsity estimate value is $k = \Lambda \times 55$.

Table 2: Averaged reconstruction performance.

	[15]	$\Lambda = 1$	$\Lambda = 2$	$\Lambda = 4$	$\Lambda = 8$
Random matrix	N/A	0.156	0.14	0.1056	0.092
Bernoulli matrix	0.078	0.157	0.1423	0.1051	0.0844

Table 2 presents performance comparison in terms of NMSE between the reconstruction quality of the proposed approach and the result presented in [15] where the same EEG data has been used. To this end, the number of samples to be transmitted $M = N/2$ is equal to the number of samples used in [15]. Moreover, in order to quantify the performance of our proposed approach, different values of Λ have been selected, such that $\Lambda = \{1, 2, 4, 8\}$. $\Lambda = 1$ represents the case where each channel is recovered individually and no joint sparsity is explored. The results show clearly that exploiting the joint sparsity improves the reconstruction quality, where the latter increases with the increase of the number of channels recovered simultaneously. In addition, the choice of the sensing matrix does not have a great impact on the reconstruction quality except for the case where CR = 0.5, in which

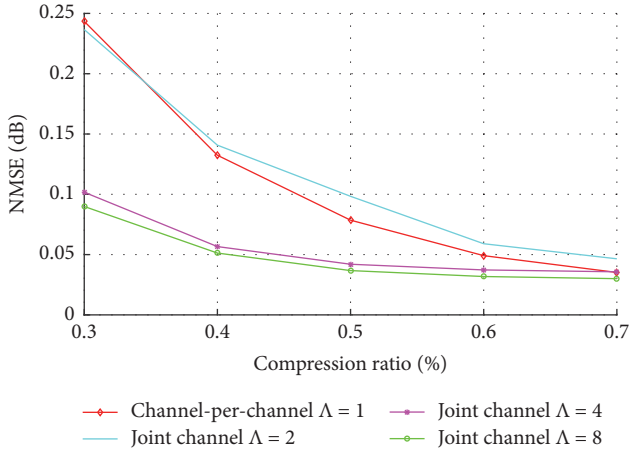


Figure 1: Recovery quality in terms of NMSE for different values of compression ratio (CR) using a random matrix.

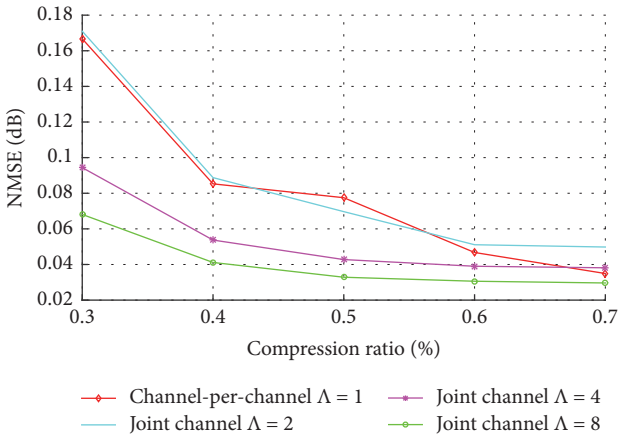


Figure 2: Recovery quality in terms of NMSE for different values of compression ratio (CR) using a binary matrix.

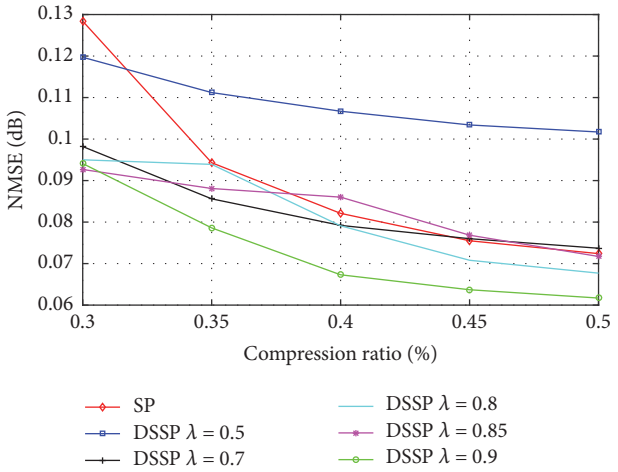


Figure 3: Comparison between reconstruction quality of DSSP with SP for channel-per-channel recovery.

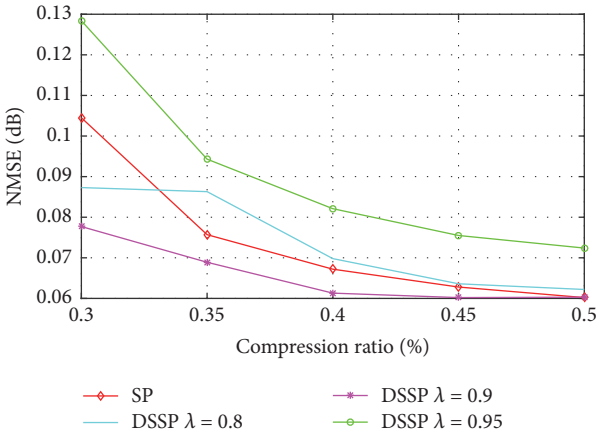


Figure 4: Comparison between reconstruction quality of DSSP with SP for joint-recovery with $\Lambda = 2$.

a better performance is obtained using Bernoulli matrix. Nevertheless, the reconstruction quality obtained in [15] still outperforms the quality achieved by the proposed approach.

The low quality of reconstruction obtained can be explained by the fact that the wavelet basis does not provide a good sparse representation to EEG data. Therefore, by using the concept of dictionary concatenation [36] which has been investigated for ECG signals in [8], a sparsifying basis for EEG has been proposed. The basis is constructed by selecting random elements from DCT and wavelet families, namely, Daubechies and Symlets.

Figures 1 and 2 present the reconstruction quality in terms of NMSE for different values of CR using random and binary sensing matrix, respectively. The obtained results consolidate the previous results where exploiting joint-recovery with $\Lambda = 4, 8$ enhances remarkably the reconstruction quality. Moreover, using the proposed dictionary renders a better reconstruction than using a DWT as sparsifying basis. Furthermore, the same reconstruction quality obtained in [15] by taking CR = 0.5 can be obtained by exploring joint-recovery combined with the proposed basis using only a CR = 0.3 for the case where $\Lambda = 8$. On the other hand, with $\Lambda = 2$, an inferior reconstruction quality is achieved compared to channel-per-channel recovery for CR value higher than 0.4. One reason for such inferior performance is the poor selection of the sparsity estimate for the SP algorithm; hence, SP is not robust to sparsity change within the different signal blocks.

Secondly, in order to quantify the performance of the proposed algorithm (DSSP), intensive numerical simulations have been carried out in order to determine the best approach to select the parameter λ. Figures 3 and 4 quantify the performance of the DDSP algorithm for different values of λ. Figure 3 presents the reconstruction quality in terms of NMSE for channel-per-channel recovery method. λ values have been selected such that $\lambda = \{0.5, 0.7, 0.8, 0.85, 0.9, 0.95\}$. The obtained results show that the NMSE decrease constantly with increasing the value of CR, which leads to a better reconstruction quality. In addition, the results show that, with value of $\lambda = 0.9$, the signal true support is very well located at each iteration which provides a better reconstructing quality.

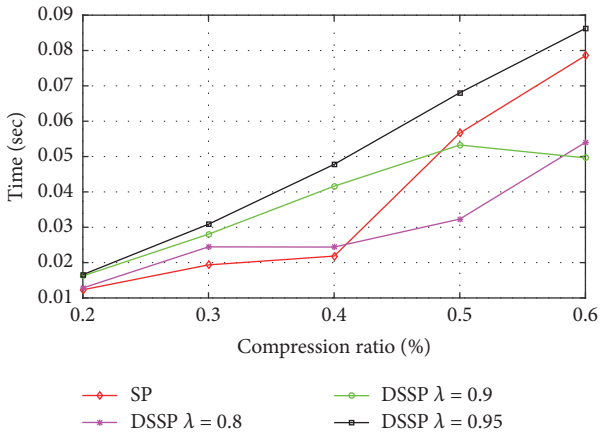


Figure 5: Averaged Reconstruction Time (sec) for different CR values.

Figure 4 illustrates the performance for joint channel recovery with $\Lambda = 2$, and the obtained results consolidate with the results presented in Figure 3 which shows the superiority of DDSP using $\lambda = 0.9$ over the SP algorithm. However, when the value of λ approaches 1, the DDSP selects more coefficients at each iteration; subsequently, the coefficients which do not belong to the true support of the signal are more likely to be selected, which can degrade the quality of the reconstruction such as the case where $\lambda = 0.95$.

Figure 5 presents the execution time for the SP with DSSP using different values of λ. The reported results are averaged over the reconstruction time for 80 channels. As the reconstruction time of the implemented algorithms is controlled by the speed of the processor and the number of tasks handled by the processor, all the algorithms have been implemented on the same program in a single PC to reduce the effect of this dependency. It is worth mentioning that the results are obtained using a PC equipped with an Intel Core i7-3770 @ 3.4 Ghz CPU and a RAM of 16.0 GB. The obtained execution time illustrates that, for CR values less than 0.4, SP converges faster compared to the proposed algorithm. However, it is worth mentioning that, for such low CR, SP does not guarantee an acceptable recovery quality. On the other hand, for CR higher than 0.4, although the proposed algorithm includes more steps in order to determine the best sparsity estimate at each iteration, the obtained results show that the proposed algorithm outperforms the SP algorithm in terms of the reconstruction time in case of $\lambda = 0.8$. This can be explained by the fact that the proposed selection approach is more likely to determine the true support of the signal which will lead to a lower number of iterations to converge to the desired reconstruction quality. However, as DSSP achieves its best performance with $\lambda = 0.9$ for a CR = 0.45 as shown previously in Figure 4, increasing the number of samples transmitted would result in an additional information about the signal which renders a faster recovery as illustrated by the decrease of the execution time from CR = 0.5 to CR = 0.6.

5. Conclusion

Biosignal compression represents a hot topic to be addressed in order to enable IoT-based connected health platforms that offer low-power consumption, simple system design, and efficient performance.

Efficient EEG compression schemes using CS have not achieved the desired performance yet. This is mainly due to the nonsparse nature of EEG signals. This paper investigates the concept of joint-recovery to improve the reconstruction quality of EEG signals by exploiting the collaboration between the multichannel measurements. Moreover, the paper explores the idea of basis concatenation to tackle the issue of EEG nonsparsity. The experimental results have been compared to the state-of-the-art BSBL algorithm using the same database as well as similar system setup. Thus, by adopting SP-based joint channel recovery, the achieved reconstruction quality outperforms the one obtained BSBL algorithm. In addition, the paper quantifies the use of both Bernoulli and random matrices as sensing matrix; the obtained results reveal that both classes provide a similar performance in terms of quality of reconstruction. However, Bernoulli matrices have proved in the literature to be hardware friendly and offer a low system complexity.

Furthermore, an adaptive reconstruction algorithm has been proposed to tackle the problem of the variation of the signal sparsity over time. The most prominent advantage of the proposed DSSP is the ability to provide a good reconstruction quality without prior knowledge of the signal sparsity which can be very useful in several applications.

In conclusion, the obtained results are very promising and their hardware implementation for remote monitoring in IoT applications can be further investigated as future work.

Disclosure

The statements made herein are solely the responsibility of the authors.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This paper was made possible by National Priorities Research Program (NPRP) grant from the Qatar National Research Fund (a member of Qatar Foundation) (Grant no. 9-114-2-055).

References

[1] C. Karakus, A. C. Gurbuz, and B. Tavli, "Analysis of energy efficiency of compressive sensing in wireless sensor networks," *IEEE Sensors Journal*, vol. 13, no. 5, pp. 1999–2008, 2013.

[2] D. L. Donoho, "Compressed sensing," *Institute of Electrical and Electronics Engineers Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[3] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete

frequency information," *Institute of Electrical and Electronics Engineers Transactions on Information Theory*, vol. 52, no. 2, pp. 489–509, 2006.

[4] D. Baron, M. F. Duarte, M. B. Wakin, S. Sarvotham, and R. G. Baraniuk, "Distributed compressive sensing," 2009, https://arxiv.org/abs/0901.3403.

[5] A. M. R. Dixon, E. G. Allstot, D. Gangopadhyay, and D. J. Allstot, "Compressed sensing system considerations for ECG and EMG wireless biosensors," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 6, no. 2, pp. 156–166, 2012.

[6] D. Bortolotti, M. Mangia, A. Bartolini, R. Rovatti, G. Setti, and L. Benini, "Energy-aware bio-signal compressed sensing reconstruction on the wbsn-gateway," *IEEE Transactions on Emerging Topics in Computing*, vol. PP, no. 99, pp. 1-1, 2016.

[7] A. M. R. Dixon, E. G. Allstot, A. Y. Chen, D. Gangopadhyay, and D. J. Allstot, "Compressed sensing reconstruction: Comparative study with applications to ECG bio-signals," in *Proceedings of the 2011 IEEE International Symposium of Circuits and Systems, ISCAS 2011*, pp. 805–808, Rio de Janeiro, Brazil, May 2011.

[8] O. Kerdjidj, K. Ghanem, A. Amira, F. Harizi, and F. Chouireb, "Concatenation of dictionaries for recovery of ECG signals using compressed sensing techniques," in *Proceedings of the 26th International Conference on Microelectronics (ICM)*, pp. 112–115, IEEE, Doha, Qatar, December 2014.

[9] L. F. Polania, R. E. Carrillo, M. Blanco-Velasco, and K. E. Barner, "Exploiting prior knowledge in compressed sensing wireless ECG systems," *IEEE Journal of Biomedical and Health Informatics*, vol. 19, no. 2, pp. 508–519, 2015.

[10] Y. Liu, M. De Vos, I. Gligorijevic, V. Matic, Y. Li, and S. Van Huffel, "Multi-structural signal recovery for biomedical compressive sensing," *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 10, pp. 2794–2805, 2013.

[11] A. M. Abdulghani, A. J. Casson, and E. Rodriguez-Villegas, "Compressive sensing scalp eeg signals: implementations and practical performance," *Medical & biological engineering & computing*, vol. 50, no. 11, pp. 1137–1145, 2012.

[12] S. Senay, L. F. Chaparro, M. Sun, and R. J. Sclabassi, "Compressive sensing and random filtering of eeg signals using slepian basis," in *Proceedings of the 16th European Signal Processing Conference*, pp. 1–5, IEEE, Lausanne, Switzerland, August 2008.

[13] S. Aviyente, "Compressed sensing framework for EEG compression," in *Proceedings of the IEEE/SP 14th WorkShoP on Statistical Signal Processing (SSP '07)*, pp. 181–184, August 2007.

[14] D. Gangopadhyay, E. G. Allstot, A. M. R. Dixon, and D. J. Allstot, "System considerations for the compressive sampling of EEG and ECoG bio-signals," in *Proceedings of the 2011 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, pp. 129–132, IEEE, San Diego, CA, USA, November 2011.

[15] Z. Zhang, T. P. Jung, S. Makeig, and B. D. Rao, "Compressed sensing of EEG for wireless telemonitoring with low energy consumption and inexpensive hardware," *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 1, pp. 221–224, 2013.

[16] Y. P. Liu, M. De Vos, and S. Van Huffel, "Compressed sensing of multichannel EEG signals: the simultaneous cosparsity and low-rank optimization," *IEEE Transactions on Biomedical Engineering*, vol. 62, no. 8, pp. 2055–2061, 2015.

[17] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *Institute of Electrical and Electronics Engineers Transactions on Information Theory*, vol. 53, no. 12, pp. 4655–4666, 2007.

[18] D. L. Donoho, Y. Tsaig, I. Drori, and J.-L. Starck, "Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit," *Institute of Electrical and Electronics Engineers Transactions on Information Theory*, vol. 58, no. 2, pp. 1094–1121, 2012.

[19] D. Needell and J. A. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," *Applied and Computational Harmonic Analysis* , vol. 26, no. 3, pp. 301–321, 2009.

[20] W. Dai and O. Milenkovic, "Subspace pursuit for compressive sensing signal reconstruction," *Institute of Electrical and Electronics Engineers Transactions on Information Theory*, vol. 55, no. 5, pp. 2230–2249, 2009.

[21] E. J. Candes and M. B. Wakin, "An introduction to compressive sampling: A sensing/sampling paradigm that goes against the common knowledge in data acquisition," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, 2008.

[22] T. T. Do, L. Gan, N. Nguyen, and T. D. Tran, "Sparsity adaptive matching pursuit algorithm for practical compressed sensing," in *Proceedings of the 42nd Asilomar Conference on Signals, Systems and Computers (ASILOMAR '08)*, pp. 581–587, Pacific Grove, Calif, USA, October 2008.

[23] G. Sun, Y. Zhou, Z. Wang, W. Dang, and Z. Li, "Sparsity adaptive compressive sampling matching pursuit algorithm based on compressive sensing," *Journal of Computational Information Systems*, vol. 7, no. 4, pp. 2883–2890, 2012.

[24] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 33–61, 1998.

[25] E. J. Candès, "The restricted isometry property and its implications for compressed sensing," *Comptes Rendus Mathematique*, vol. 346, no. 9-10, pp. 589–592, 2008.

[26] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, 1993.

[27] A. M. Abdulghani, A. J. Casson, and E. Rodriguez-Villegas, "Quantifying the feasibility of compressive sensing in portable electroencephalography systems," in *International Conference on Foundations of Augmented Cognition*, vol. 5638 of *Lecture Notes in Computer Science*, pp. 319–328, Springer, Berlin, Germany, 2009.

[28] J. Haboba, M. Mangia, R. Rovatti, and G. Setti, "An architecture for 1-bit localized compressive sensing with applications to EEG," in *Proceedings of the 2011 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, pp. 137–140, IEEE, San Diego, CA, USA, November 2011.

[29] Z. Zhang, T.-P. Jung, S. Makeig, and B. D. Rao, "Compressed sensing for energy-efficient wireless telemonitoring of non-invasive fetal ECG via block sparse bayesian learning," *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 2, pp. 300–309, 2013.

[30] G. Higgins, S. Faul, R. P. McEvoy et al., "EEG compression using JPEG2000 how much loss is too much?" in *Proceedings of the 2010 32nd 2010 Annual International Conference of the IEEE Engineering in Medicine and Biology (EMBC)*, pp. 614–617, IEEE, Buenos Aires, Argentina, September 2010.

[31] J. L. Cárdenas-Barrera, J. V. Lorenzo-Ginori, and E. Rodríguez-Valdivia, "A wavelet-packets based algorithm for EEG signal compression," *Medical Informatics and the Internet in Medicine*, vol. 29, no. 1, pp. 15–27, 2004.

[32] G. Higgins, B. McGinley, N. Walsh, M. Glavin, and E. Jones, "Lossy compression of EEG signals using SPIHT," *IEEE Electronics Letters*, vol. 47, no. 18, pp. 1017-1018, 2011.

[33] A. J. Casson and E. Rodriguez-Villegas, "Signal agnostic compressive sensing for body area networks: Comparison of signal reconstructions," in *Proceedings of the Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 4497–4500, IEEE, 2012.

[34] A. Delorme and S. Makeig, "EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis," *Journal of Neuroscience Methods*, vol. 134, no. 1, pp. 9–21, 2004.

[35] A. Mishra, F. Thakkar, C. Modi, and R. Kher, "Comparative analysis of wavelet basis functions for ecg signal compression through compressive sensing," *International Journal of Computer Science and Telecommunications*, vol. 3, no. 5, pp. 23–31, 2012.

[36] E. J. Candès, Y. C. Eldar, D. Needell, and P. Randall, "Compressed sensing with coherent and redundant dictionaries," *Applied and Computational Harmonic Analysis* , vol. 31, no. 1, pp. 59–73, 2011.

Hindawi
Wireless Communications and Mobile Computing
Volume 2017, Article ID 9321463, 9 pages
https://doi.org/10.1155/2017/9321463

Research Article

Using Emotions in Intelligent Virtual Environments: The EJaCalIVE Framework

Jaime A. Rincon,[1] Angelo Costa,[2] Paulo Novais,[2] Vicente Julian,[1] and Carlos Carrascosa[1]

[1]*Departamento de Sistemas Informáticos y Computación, Universitat Politècnica de València, València, Spain*
[2]*Centro ALGORITMI, Escola de Engenharia, Universidade do Minho, Braga, Portugal*

Correspondence should be addressed to Vicente Julian; vinglada@dsic.upv.es

Received 29 June 2017; Revised 15 September 2017; Accepted 9 October 2017; Published 13 November 2017

Academic Editor: Javier Prieto

Nowadays, there is a need to provide new applications which allow the definition and implementation of safe environments that attends to the user needs and increases their wellbeing. In this sense, this paper introduces the EJaCalIVE framework which allows the creation of emotional virtual environments that incorporate agents, eHealth related devices, human actors, and emotions projecting them virtually and managing the interaction between all the elements. In this way, the proposed framework allows the design and programming of intelligent virtual environments, as well as the simulation and detection of human emotions which can be used for the improvement of the decision-making processes of the developed entities. The paper also shows a case study that enforces the need of this framework in common environments like nursing homes or assisted living facilities. Concretely, the case study proposes the simulation of a residence for the elderly. The main goal is to have an emotion-based simulation to train an assistance robot avoiding the complexity involved in working with the real elders. The main advantage of the proposed framework is to provide a safe environment, that is, an environment where users are able to interact safely with the system.

1. Introduction

Currently there is a lot of efforts employed in the areas of ubiquitous computing, social robotics, and wearable mobile devices. Advances on new hardware devices such as Wi-Fi and Bluetooth have provided the means to implement embedded systems in common households. Furthermore, with the rise of Internet of Things (IoT) every device can be connected to the Internet and transmit data [1]. With an array of these devices (that at their core they are sensors or actuators) the ability of creating a sensor platform that is able to capture several information from the users increases. With this information, the environment can be manipulated to attend to the liking of the people with the system, for instance, changing the temperature and the lighting. Furthermore, it is possible to improve energy consumption by attending to factors that the human-beings (users of these platforms) do not actively consider (like preemptively close windows blinds).

These features are accomplished through the use of machine learning algorithms. The algorithms give the ability of learning the home users' likes and preferences. Typically in these systems the information about the users is introduced early on and through the capture of the users' interaction by the sensors the system changes those values to correspond to personal preferences [2]. The user's information is generated by performing computational operations on sensor data. These operations could be as simple as web service calls or as complex as mathematical functions run over sensed data. There is an issue with this approach, that is, the transformation of these systems into reactive ones, not being able to correctly foresee what the user really likes. Most of the system works almost randomly, where it does not matter if the user is happy or sad; it reacts the same way for both of the emotions, although they correlate with different needs.

The detection of human emotions is preponderant to give to these platforms the ability of evolving in time and having fuzzy logic states [3]. The detection and simulation of emotions can be considered as a new type of interaction that allows the system to know the user's emotional state and/or to simulate and express an emotion. The result of this type of interaction is the active response of the environment to the emotional states.

The main goal is to provide tools which allow the definition and implementation of safe environments that attends to the user needs and increases their wellbeing [4]. Studies show that emotions have a direct impact on mental and physical health [5, 6]. The environment changes may give the sense of detachment and that its decisions are not in favor of the users. This problem is mainly produced by the different semantic interpretation of message meanings sent and received between devices and/or human users. In order to overcome this problem we propose the employment of social robots in these environments. A social robot is an autonomous or semiautonomous robot that interacts and communicates with humans by following the behavioral norms expected by the people with whom the robot is intended to interact [7].

Communication and interaction with humans are a critical point in this definition. Social robots increase the level of human-computer interaction and have social abilities like initiating a conversation and controlling our homes (BIG-i social home robot) (https://www.nxrobo.com/), or learning about the personality of the users (Jibo) (https://www.jibo.com/) or the user's health condition and current treatments (Catalia Health) (http://www.cataliahealth.com/). These robots' aim is to provide a human-like feeling to their interactions with the users. They should be able to respond appropriately to human affective and social cues in order to effectively engage in bidirectional communications [8].

There are relevant developments in the social robots area, but there are technological issues that are still unsolved such as the centralization of operations [9]. The robots are unable to perceive correctly the environment due to the lack of sensors and processing abilities. As a consequence, they typically serve only as a gateway to the rest of the platform. Furthermore, the system scaling is typically a problem. This is because these systems are designed with only one robot in one home, and the introduction of another robot is not considered and, consequently, the results of possible interactions are unexplored.

Addressing these issues is critical; thus, it is necessary to have tools that manage the information in a distributed way. Our solution proposes the use of intelligent entities (agents) that are autonomous and decentralized. This way, homes or offices can be seen as autonomous entities working together in order to provide comfort and security to the inhabitants. The system must be able to cope with the dynamic introduction of new entities.

The proposed tool, called EJaCalIVE, is based on the multiagent system paradigm, which gives the users the ability of designing and simulating emotional intelligent virtual environments (EIVE). Moreover, it incorporates elements of perception and action (machine learning algorithms, artificial vision, speech recognition, and communication with social robots and wearable devices), allowing the design and construction of EIVE capable of interacting with human beings in a natural way. The main advantage is to obtain emotion-based simulations to try different configurations (like training an assistance robot) avoiding the complexity involved in working with real people.

To achieve this, the designed tool has two levels: a user level and a developer level. The developer level gives allowance to design and simulate the EIVE, while the user level gives allowance to use and interact with the IVE constructed by the developer.

The rest of the paper is structured as follows: Section 2 analyses previous works; Section 3 shows the proposed EJaCalIVE framework while Section 4 describes a case study based on that framework; finally, Section 5 explains some conclusions and future work.

2. Related Work

Ubiquitous computing and ambient intelligence (AmI) [10, 11] changed the concept of the Smart Homes to focus on the users' quality of life. From these paradigms platforms that present devices and software that learn and adapt to the inhabitants' tastes emerge. There are already some developments that address the most common concerns that most Smart Homes users present, for example, energy consumption tracking [12] and safer environments for elderly [13, 14]. To address these concerns AmI and IoT solutions may be used.

AmI and IoT projects establish environments where the users are surrounded by different kinds of technology elements [15] that help them on their daily tasks, being transparent to them [16]. A recent area that emerged from the AmI and IoT is the Ambient Assisted Living (AAL), being its goal to provide assistive environments for elderly and disabled people. Due to the elderly and the disabled people medical condition, a regular home environment is very challenging to them. They need special assistance and devices and services that help them to perform activities of daily living (ADL).

In terms of related projects, there are recent developments on the AAL area as well as in intelligent virtual environments. From our observations, the common issue that they present is the lack of interoperability features and the oversight over managing emotions. Next some relevant projects of the referred areas are presented.

The NACODEAL project [17] goal is to provide an augmented reality device that shows information about the activity that they should perform or are performing. This tool is very useful to people with cognition problems or that need assistance when performing a complex task, as the device shows step-by-step instructions about those tasks. An implementation is presented in [18], where a system that uses inexpensive devices (projectors, cameras, and speakers) to guide a user through a home environment is described, showing physical directions and warnings about the surroundings. Internally, the system resorts to virtual environments to forecast the possible outcomes of the user actions. This project changes the paradigm of the Smart Homes by incorporating sensors in the devices that the user uses, thus having a mobile sensor system and requiring fewer home devices. Although this project presents a complex and interesting sensor system and an innovative visual interface, it lacks the ability to detect emotions and really interpret the intentions of the users. Furthermore, the users cannot

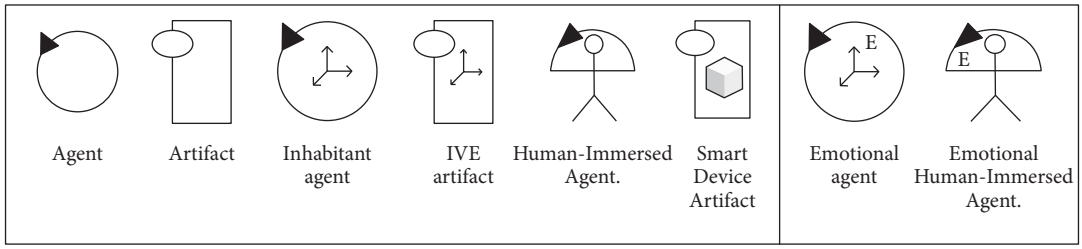


Figure 1: Entities employed in *EJaCalIVE*.

input their preferences or interact outside the predetermined actions.

The PersonAAL project [19] aims to provide a tool for elderly people that reduces their dependence on caregiving services. The tool will be implemented in each user house and constantly monitors them and their activities. The main idea is to use smartphones and wall displays to provide a virtual environment that has useful information about the activity that they are performing, adjusting the visual interfaces to each user according to their taste and medical condition. The issue with this tool is the limitation of the information available and the features that they provide (only information without any acting through actuators or robotic assistants).

Active@Work project [20] addresses an often forgotten area that is the workplace. The main goal is to help elderly workers on their workplace using a virtual assistant. This assistant receives the information of biosensors and identifies possible risk situations through the comparison to normal sensor values. Furthermore, the project aims to develop a visual interface that sensibly communicates possible problems to the users and to caregivers, having a virtual environment for collaborative processes that load-balances the work and sends some of the elderly tasks to younger employees. This project showcases the use of sensor systems to detect and address health issues, as well as emotional and stress levels. The issue is that it proposes limited action when critical levels are achieved; thus it does not work in the best interest of the people that are monitored.

Finally, the Pepper robot [21] was designed to identify the emotions of the users and select the best behavior appropriate to the situation, having a humanoid aspect. This robot takes into account the voice, facial expression, body movements, and oral expression, interpreting the emotion presented and offering the appropriate content. The robot is able to identify joy, sadness, anger, or surprise and responds to the mood of the moment, expressing himself through the color of his eyes, his tablet, or his tone of voice. Furthermore, it uses gamification procedures to engage with the users and keep them interested and follows the users interacting with them as they move. Through its arms it is able to shake hands and express physical emotions. One issue with this robot is that at its core is a companion (as in conversational) robot, not being able to carry or assist in any tasks, and although it is connected to the Internet, the quantity of information that is able to display (visual or auditory) is reduced to simple responses to vocal commands. Also, it is unable to anticipate actions or movements; thus it is not prepared to prevent critical situations and is only able to directly respond to the immediate emotions displayed by the users.

The EJaCalIVE is designed to address the issues that these projects overlook, providing a platform and robot that is able to attend to the users issues and needs.

3. EJaCalIVE (Emotional Jason Cartago Implemented Intelligent Virtual Environment)

This section focuses on the presentation of the *EJaCalIVE* framework. This framework allows the design and programming of intelligent virtual environments, as well as the simulation and detection of human emotions for the creation of *IoT* and *UC* applications.

EJaCalIVE is a tool that allows the design and programming of these new human-agent societies while incorporating the detection and simulation of emotions. *EJaCalIVE* is divided into two parts. The first one focuses on the design and programming of the intelligent virtual environment (IVE) and the second one on the detection and simulation of emotional states. For the design of *IVEs*, *EJaCalIVE* uses the *MAM5* [22] metamodel based on agent and artifact *(A&A)* [23, 24] one. The *A&A* metamodel determines that, within an environment, there are two types of entities, the intelligent ones (Agents) and the objects (Artifacts). Based on this idea, MAM5 metamodel goes a step further to design an IVE, in terms of distinguishing between the entities (agents and artifacts) that have a physical representation in the virtual environment from the ones that have not. *EJaCalIVE* introduces different specializations of agents and artifacts, which give the designer a wide range of entities to realize their design. Figure 1 shows the different entities that can be represented in *EJaCalIVE*.

Detailing a little more *EJaCalIVE* structure, inherited of MAM5 model, which divides the IVE into two workspaces, a workspace where entities that do not have virtual representation exist and an *IVE_Workspace* inhabited by entities with a virtual representation. In turn, within each of these workspaces, the entities are divided into two main classes, corresponding to agents and artifacts. In this way, in the first kind of workspace, those that do not have a virtual representation (no 3D representation), there are agents and artifacts just as defined in the A&A metamodel. On the other hand,

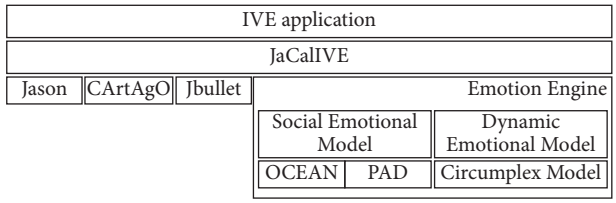


Figure 2: Main structure of *EJaCalIVE* framework.

in the *IVE_Workspace*, there are those that have a virtual representation, the *Inhabitant Agents* and the *IVE_Artifacts*. Moreover, this last kind of workspace allows more specific subclasses: *Smart Device Artifacts* (SDA) [25] and *Human-Immersed Agents*. The SDA are specialized *IVE_Artifacts*, allowing the developer to make a connection with the real world. This connection gives agents and *Inhabitant Agents* the ability to interact with the real world acquiring information through sensors or acting in the real world using actuators. The *Human-Immersed Agent* is an *Inhabitant Agent* that serves to immerse a human in the system, so for the rest of the system is an *Inhabitant Agent*, but it is also a communication bridge between the virtual world and the real world.

However, the introduction of emotions into the IVE design, as a new form of communication and interaction between entities, along with the above commented possibilities of accessing not only the virtual environment but also the real world (in an augmented reality kind of applications) open the door for the creation of new IoT, UC, and complex simulations. This new emotional component gives the different entities the ability to simulate and/or detect human emotions. For this, it is necessary to introduce different emotional models that are commonly used in psychology and are widely used in computer science. Within these models, we can find the big-five model (OCEAN) which is a personality model, the emotional model PAD [26], and the Circumplex Model [27]. *EJaCalIVE* uses different communication channels (image processing through cameras, text, voice, body gestures, or biosignals) for the detection of emotions. This input data dealt with machine learning algorithms such as SVM or neural networks for the detection and classification of emotions from it.

Figure 2 shows the main structure of *EJaCalIVE* framework along with the different modules it is based on. As can be observed in such figure, *EJaCalIVE* is supported by four engines: cognitive, artifact, physical, and emotion. Each of these engines allows the developer to design and program an *Emotional Intelligent Virtual Environment (EIVE)*. The *Cognitive Engine* is supported in turn by Jason who is the agent platform. Jason allows scheduling of each of the behaviors of the agents. The *Artifact Engine* is supported by *CArtAgo* which allows you to create the various objects that are inside the *EIVE*. *EJaCalIVE* has a *Physical Engine*, which is supported by Jbullet. Jbullet allows to introduce physical restrictions (gravity, *IVE-Artifact* position, speed, and acceleration among others) which will be governed by the *IVE Workspace*. The *Emotion Engine* is responsible for simulating and classifying human emotions as well as for the calculation of social emotion and the emotional dynamics of human-agent society.

Each one of the engines is defined by the developer through an XML file, which will later be interpreted by *EJaCalIVE* creating the different templates for agents, artifacts, and workspace data.

The following subsection describes the *Emotion Engine* included in *EJaCalIVE* (the other engines are described in more detail in [28, 29]).

3.1. EJaCalIVE Emotion Engine. The *Emotion Engine* plays an important role within the framework, as it is responsible for incorporating the different human emotions. This engine can be used by all agents, to detect, process, and simulate emotions. This engine also allows the developer to use all simulated or detected emotions to calculate a social emotion.

The processes of detection and simulation of emotions are described below.

(1) Emotion detection: *EJaCalIVE's Emotion Engine* detects the emotions using the artifacts designed to perform this task. However, *EJaCalIVE* allows the developer to connect any other hardware that can perform such detection. By default *EJaCalIVE* incorporates two ways of performing the detection, the first is through image processing and the second is using an *Emotional Wristband*.

In the detection of emotions through images processing, the system uses the Face Landmark Algorithm, which extracts the characteristic points of the face from the images [30]. An example of image detection can be found in [31].

To characterize a face image, a vector of characteristics is created storing not only the characteristic points but also the Euclidean distances between those points. This vector of characteristics serves as the input for a neural network that gives as output the corresponding emotion expressed by such face. This network has been trained previously with similar vectors calculated from a database of face images representing different emotions. Although *EJaCalIVE* uses neural networks to perform the classification, it allows the developer to use their own classifying scripts. This is done through a dedicated artifact for this task (scripts can be made in Python or Node.js).

To perform the detection of emotions through the *Emotional Wristband*, we employ a design described in more detail in [32]. This band is used by a human; an agent is embedded inside the band perceiving the variations of the resistance of the skin. These variations are preprocessed in order to extract a feature vector, which is used by a neural network classifying the emotion (in a similar way as in the emotion detection by images processing).

(2) Emotion simulation: if we want to simulate the emotions of humans, we need to obtain previously the personality values of the involved humans. These values are obtained by making these humans perform

Table 1: Adjectives describing personalities.

$O+$	Curious, alert, informed, perceptive
$O-$	Simple, narrow, ignorant
$C+$	Persistent, orderly, predictable, dependable, prompt
$C-$	Messy, careless, rude, changeable
$E+$	Social, active, assertive, dominant, energetic
$E-$	Distant, unsocial, lethargic, vigorless, shy
$A+$	Cooperative, tolerant, patient, kind
$A-$	Bossy, negative, contrary, stubborn, harsh
$N+$	Oversensitive, fearful, dependent, submissive, unconfident
$N-$	Calm, independent, confident

the *big-five* personality test (https://personality-testing.info/printable/big-five-personality-test.pdf). This test models the personality of an individual through five factors: factor O (openness to new experiences), factor C (conscientiousness or responsibility), factor E (extraversion or extroversion), factor A (agreeableness or kindness), and factor N (neuroticism or emotional instability). If the agent is simulated, we need to determine its personality. To do this it is necessary to vary the values of the OCEAN in each component as shown in Table 1 [33].

However, it is possible to use these personality values to calculate the emotion that our agent would have at the moment of initiating the simulation. For this we use the following equation [34] that allows us to determine the first emotion in terms of *PAD*.

$$\begin{aligned} P0 &= (0.59 * A + 0.19 * N + 0.21 * E), \\ A0 &= (-0.59 * N + 0.30 * A + 0.15 * O), \\ D0 &= (0.60 * E - 0.32 * A + 0.25 * O + 0.17 * C). \end{aligned} \tag{1}$$

Once these first values of *PAD* have been calculated, during the simulation process emotions can vary. This variation is produced through the perceptions of the agent and will depend on the scenario to be simulated (e.g., using music to modify moods of the people in a pub, the agents perceive the music, and their emotions evolve according to their current emotion and the kind of music they like [31]).

As commented above, the *Emotion Engine* allows the developer to use all simulated or detected emotions in order to calculate a social emotion. The use of the concept of social emotion allows the developer to know the emotional state of the group, composed of humans and agents. This social emotion is represented as a triplet composed of: the *Central Emotion* (average of the emotions of each individual in the group), the maximum distance between the Central Emotion and the emotions of the individuals and the dispersion of the emotions around the Central Emotion (for more details [29]).

Working with social emotions allows comparing two groups of individuals and comparing their social emotions, in fact, the distance between such social emotions, and so to know how close or far they are from an emotional perspective or even compare the social emotion of a group with a goal emotion.

It is possible to modify the individual emotions of each person, causing the social emotion to change and thus to make that distance between social emotions increase or decrease. However, the emotions are dynamic, as well as the interactions between agents and humans. Moreover, emotions can be spread among the individuals who are being simulated. In order for this emotional contagion to take place, it is necessary to take into account not only personality traits as the empathy commented above, but also the affinity between the individuals. In this sense, the proposed *EJaCalIVE* incorporates a dynamical model which allows the designer to model emotional contagion between individuals.

4. Case Study

In this section, we present the case study in order to use *EJaCalIVE* in the simulation of a residence for the elderly. Due to the complexity involved in working with these people, an emotion-based simulation is proposed to train an assistance robot. The robot will interact with the agents detecting their emotions and communicating with the caregiver if there is any variation. This emotional variation allows the caregiver to decide if it is needed to change the activities to modify the people emotions.

The simulation presented in this section was divided in a virtual component and a real component. The virtual component is responsible for simulating the elderly; to do this, we use three emotional agents. These agents have different personalities and a list of characteristics that make them different (features such as affinity, empathy, and activity tastes). These different characteristics make each agent's emotion be affected in a different way by environmental stimulus. The real component is performed with the robot; this robot can detect the emotional states and change its behaviour depending of the emotion detected. In this way, this simulation trains not only the mechanism used by the robot to detect (and act to try to modify) the human emotions, but also an environment where the robot could interact with the elements of an IVE was developed. For this reason, other elements have been taken into account such as furniture (chairs, sofa, table, and floor) to be simulated as IVE_Artifacts. These elements are located as virtual obstacles to be avoided by the robot (in the real world). We have integrated a human in our simulation. This human is modeled as an *Emotional_Human_Inmersed_Agent*. This agent is a virtual representation of a caregiver and allows us to test the mechanism of emotion detection using the camera and the interaction through different actions. Figure 3 shows in the upper left corner the metamodel design of the case study system, formed by 4 IVE_Artifacts (modelling the table, chair, sofa, and TV), 4 Emotional Inhabitant Agents (modelling the assistant robot and the 3-elderly people in the residence), and one Emotional Human-Immersed Agent (modelling the caregiver). *EJaCalIVE* allows compiling this metamodel and build automatically the skeleton files for such agents, artifacts, and the IVE_Workspace (that is seen in the upper right corner of Figure 3). The last part of this figure shows the view of the simulation done.

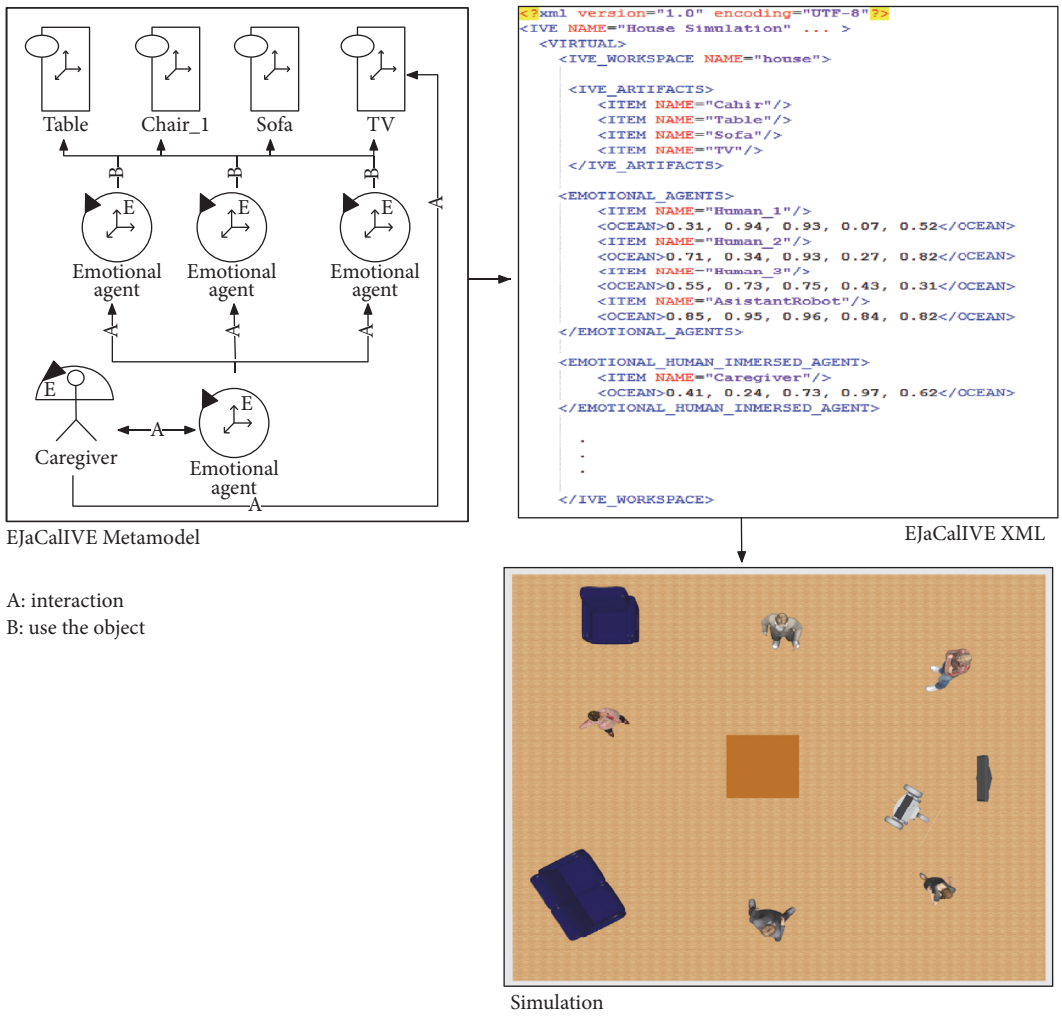


Figure 3: Visualization of the design environment.

4.1. Robot Description. The robot is responsible for interacting with the elderly; this interaction is performed through emotion detection and the proposal of activities. As the robot moves in a real world, it is important to provide it with a series of sensors that help it to navigate, to perceive the human emotions, and to interact with the people. For this reason, the robot has been divided into two levels.

The first level provides the robot with the capability of controlling the motors and giving access to different sensors that allow perceiving the environment. We employ sensors such as ultrasound, magnetometers and gyroscope. This control has been developed using an Arduino Mega (https://www.arduino.cc/en/Main/arduinoBoardMega). The data acquired by these sensors and processed by the Arduino is sent to the agent. This information is the knowledge of the environment (angle inclination, long-distance obstacle detection). However, this low-level control has a reactive behaviour that allows reacting to external events without having to be reasoned by the agent (see Figure 4).

The cognitive level was developed with a Raspberry Pi 3 (https://www.raspberrypi.org/). This level is responsible for recognizing faces and detecting emotions through image processing using a camera. At the same time, this level is in charge of controlling the robot movement. The robot includes a LCD touchscreen where users can interact, as well as speakers and microphones. All the described processes have a very high consumption of resources. This is the main reason why the robot has three raspberry pi connected as a cluster (see Figure 5).

The cluster configuration allows distributing the information and the different processes to be done. In the robot, each node is an agent with a specific task and different resources needs:

(i) The "node_1" controls the LCD touchscreen. It executes the person identification and the emotion classification behaviors. At the same time, this node is in charge of visualizing the corresponding emoticons according to the detected emotions.

(ii) The "node_2" incorporates the robot's behaviors in charge of acquiring the temperature, CO_2 level, and relative humidity sensors. This information is used by

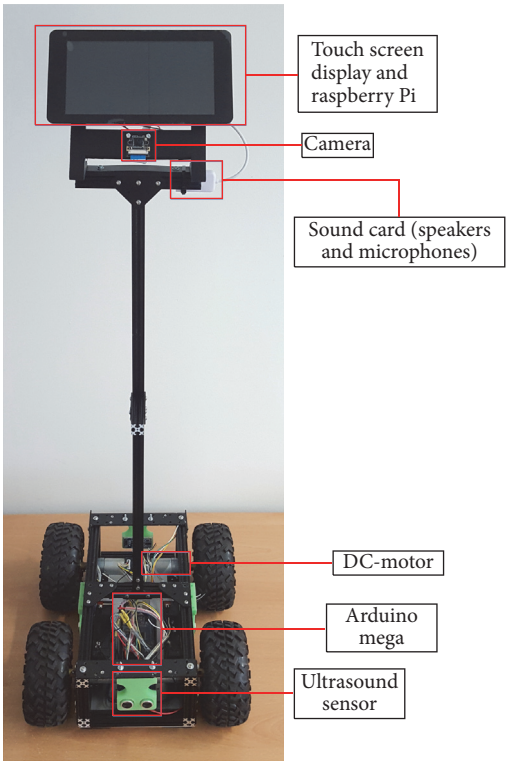


Figure 4: Image of the developed prototype of the assistant robot.

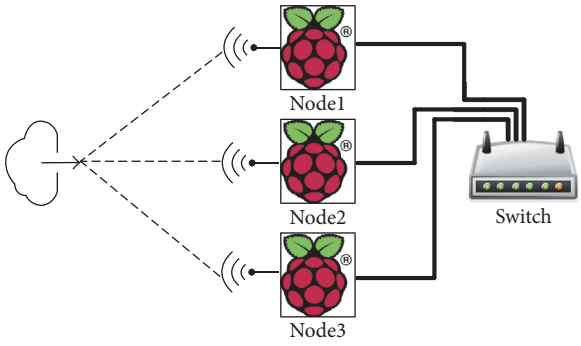


Figure 5: Cluster agent distribution.

the robot to determine if the environmental conditions where people are located are adequate, that is, if the temperature is adequate, or the humidity level is right, and if the CO_2 level are suitable. These values are acquired using a hat sensor of the raspberry pi. At the same time, this agent is in charge of carrying out the speech recognition and converting the text to speech.

(iii) Finally, the "node_3" incorporates the behaviors in charge of communicating with the low-level control. For instance, it sends the different values that make the robot move within the environment and receive the information sent by the sensors located in the low-level control, that is, velocities values, angle rotations, ultrasonic distance, motor position, and so forth.

All the agents are interconnected with each other through the SPADE platform (http://spade.gti-ia.dsic.upv.es). Each agent located in each node uses the switch for communication through messages, distributing the information between them.

The personality of the robot was defined using the OCEAN values, as defined in Table 1, taking into account that since the robot has to be at the service of humans, the OCEAN values are high to have a robot with the following characteristics:

(i) Agreeableness, tendency to be compassionate and cooperative towards others

(ii) Conscientiousness, tendency to act in an organized or thoughtful way

(iii) Extraversion, tendency to seek stimulation in the company of others

(iv) Neuroticism, emotions being sensitive to the individual's environment

(v) Openness, tendency to be open to experience a variety of activities.

Since the robot is real and older people are simulated through agents, in this simulation we have defined two restrictions. The first one is that the agents communicate their emotional states through a message since there is no emotional representation through avatars using screens. The second one is that the robot knows the emotion of all people. This allows us to calculate the social emotion and also determine the emotional dynamics of the group. With the use of the emotional dynamics, the robot can determine which person influences more about others. This way the robot can determine which person is the one that causes the emotion of the group to fall. This information is used by the caregiver, in order to design actions focused on the people who make the group's emotion fall.

5. Conclusions and Future Work

This paper has presented the EJaCalIVE framework which is an intelligent virtual environment that implements the concept of emotions and also allows an integral interaction with human beings. The main goal of this project is to build a robust virtual environment that is able to capture and reproduce atomic emotions to its agents and artefacts.

The social aspect of this project is to provide assistance to an elderly community by reading each person's emotional status and interacting through environmental changes. The robot developed (although it is still an initial build) is used as an humanoid to ease the interaction with the people on its environment. Furthermore, its mobility helps in terms of screen and sensors displacement, meaning that instead of requiring several sensor systems and that the users have to displace themselves to locations that have interaction interfaces we are able to bring it to them. We believe that this will have a lesser visual impact and disturb less than the robot. Previous studies show that robots are well accepted by elderly people.

The combination between the robot and agent projections of the elderly community has provided interesting preliminary results. Unfortunately, due to the fragility of the elderly people and the alpha version of the robot, tests on real environment and with real people were not possible. But that restriction has given way to the enhancement of the EJaCalIVE [25], with the introduction of extended emotion representation, artifacts (and its enhancements like smart devices), and social emotion representation. In fact, these developments are now the part of the core ecosystem operation; that is, the EJaCalIVE will be used even when real users and robot are interacting with each other. In terms of future work, we aim to build a user-safe version of the robot and deploy it in an environment with real users and capture the interaction with them, which may result in a validation scenario. Furthermore, we aim to develop robust quality of information methods that are able to assert an quantify scenarios with lack of information, correspondent with most that happen in real life events.

Finally, the trained robot will be tested by elderly of a daycare center in the northern area of Portugal, *the Centro Social Irmandade de S. Torcato*. After the tests, a validation will be performed through a questionnaire that will be done to the caregivers (registered nurses and medical personnel), trying to identify the obtained results and detected problems.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is partially supported by the MINECO/FEDER TIN2015-65515-C4-1-R and the FPI Grant AP2013-01276 awarded to Jaime-Andres Rincon. This work is also supported by COMPETE: POCI-01-0145-FEDER-007043 and Fundacao para a Ciencia e Tecnologia (FCT) within the projects UID/CEC/00319/2013 and Post-Doc scholarship SFRH/BPD/102696/2014 (A. Costa).

References

[1] L. Atzori, A. Iera, and G. Morabito, "The internet of things: a survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[2] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context aware computing for the internet of things: a survey," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 414–454, 2014.

[3] G. J. Klir and B. Yuan, *Fuzzy Sets and Fuzzy Logic: Theory and Applications*, vol. 4, Prentice Hall, Upper Saddle River, NJ, USA, 1995.

[4] C. Björkskog, *Human Computer Interaction in Smart Homes*, 2007.

[5] D. N. Kiosses, J. J. Gross, S. Banerjee, P. R. Duberstein, D. Putrino, and G. S. Alexopoulos, "Negative emotions and suicidal ideation during psychosocial treatments in older adults with major depression and cognitive impairment," *The American Journal of Geriatric Psychiatry*, vol. 25, no. 6, pp. 620–629, 2017.

[6] J. Ciarrochi, F. P. Deane, and S. Anderson, "Emotional intelligence moderates the relationship between stress and mental health," *Personality and Individual Differences*, vol. 32, no. 2, pp. 197–209, 2002.

[7] C. Bartneck and J. Forlizzi, "A design-centred framework for social human-robot interaction," in *Proceedings of the 13th IEEE International Workshop on Robot and Human Interactive Communication (RO-MAN '04)*, pp. 591–594, Kurashiki, Japan, September 2004.

[8] T. Fong, I. Nourbakhsh, and K. Dautenhahn, "A survey of socially interactive robots," *Robotics and Autonomous Systems*, vol. 42, no. 3-4, pp. 143–166, 2003.

[9] M. Rambow, F. Rohrmüller, O. Kourakos et al., "A framework for information distribution, task execution and decision making in multi-robot systems," *IEICE Transaction on Information and Systems*, vol. E93-D, no. 6, pp. 1352–1360, 2010.

[10] M. Satyanarayanan, "A catalyst for mobile and ubiquitous computing," *IEEE Pervasive Computing*, vol. 1, no. 1, pp. 2–5, 2002.

[11] E. Mangina, J. Carbo, and J. M. Molina, *Agent-Based Ubiquitous Computing*, Atlantis Press, Amsterdam, the Netherlands, 2009.

[12] D.-M. Han and J.-H. Lim, "Smart home energy management system using IEEE 802.15.4 and zigbee," *IEEE Transactions on Consumer Electronics*, vol. 56, no. 3, pp. 1403–1410, 2010.

[13] S. S. Intille, "Designing a home of the future," *IEEE Pervasive Computing*, vol. 1, no. 2, pp. 76–82, 2002.

[14] M. Satyanarayanan, "Pervasive computing: vision and challenges," *IEEE Personal Communications*, vol. 8, no. 4, pp. 10–17, 2001.

[15] J. C. Augusto, "Ambient intelligence: the confluence of ubiquitous/pervasive computing and artificial intelligence," in *Intelligent Computing Everywhere*, pp. 213–234, Springer, London, UK, 2007.

[16] K. Ducatel, M. Bogdanowicz, F. Scapolo, J. Leijten, and J.-C. Burgelman, *Scenarios for Ambient Intelligence in 2010*, Office for official publications of the European Communities, 2001.

[17] NACODEAL, 2017, http://www.nacodeal.eu.

[18] R. F. Saracchini and C. A. Catalina, "An augmented reality platform for wearable assisted living systems," *Journal of Theoretical and Applied Computer Science*, vol. 9, no. 1, pp. 56–79, 2015.

[19] G. Ghiani, M. Manca, F. Paternò, and C. Santoro, "End-user personalization of context-dependent applications in AAL scenarios," in *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '16)*, pp. 1081–1084, Association for Computing Machinery, Florence, Italy, September 2016.

[20] K. Stendal, D. Thapa, and A. Lanamaki, "Analyzing the Concept of Affordances in Information Systems," in *Proceedings of the 49th Hawaii International Conference on System Sciences (HICSS '16)*, pp. 5270–5277, Koloa, Hawaii, USA, January 2016.

[21] SoftBank Robotics, Peper, 2017, https://www.ald.softbankrobotics.com/en/robots/pepper.

[22] A. Barella, A. Ricci, O. Boissier, and C. Carrascosa, "MAM5: multi-agent model for intelligent virtual environments," in *Proceedings of the 10th European Workshop on Multi-Agent Systems (EUMAS 2012)*, pp. 16–30, Dublin, Ireland, December 2012.

[23] A. Omicini, A. Ricci, and M. Viroli, "Artifacts in the A&A meta-model for multi-agent systems," *Autonomous Agentsand Multi-Agent Systems*, vol. 17, no. 3, pp. 432–456, 2008.

[24] A. Ricci, M. Viroli, and A. Omicini, "CArtA gO: A framework for prototyping artifact-based environments in MAS," in *Environments for Multi-Agent Systems III*, vol. 4389 of *Lecture Notes in Computer Science*, pp. 67–86, Springer, Berlin, Germany, 2007.

[25] J. A. Rincon, J.-L. Poza-Lujan, V. Julian, J.-L. Posadas-Yagüe, and C. Carrascosa, "Extending MAM5 meta-model and JaCalIV E framework to integrate smart devices from real environments," *PLoS ONE*, vol. 11, no. 2, Article ID e0149665, pp. 1–27, 2016.

[26] A. Mehrabian, "Analysis of affiliation-related traits in terms of the PAD Temperament model," *The Journal of Psychology*, vol. 131, no. 1, pp. 101–117, 1997.

[27] J. A. Russell, "A circumplex model of affect," *Journal of Personality and Social Psychology*, vol. 39, no. 6, pp. 1161–1178, 1980.

[28] J. A. Rincon, A. Costa, P. Novais, V. Julian, and C. Carrascosa, "A dynamic emotional model for agent societies," in *Advances in Practical Applications of Scalable Multi-agent Systems. The PAAMS Collection*, vol. 9662 of *Lecture Notes in Computer Science*, pp. 169–182, Springer International Publishing, Cham, Switzerland, 2016.

[29] J. A. Rincon, V. Julian, and C. Carrascosa, "Social emotional model," in *Advances in Practical Applications of Agents, Multi-Agent Systems, and Sustainability: The PAAMS Collection*, vol. 9086 of *Lecture Notes in Computer Science*, pp. 199–210, Springer International Publishing, Cham, Switzerland, 2015.

[30] Y. Huang, H. Yao, S. Zhao, and Y. Zhang, "Towards more efficient and flexible face image deblurring using robust salient face landmark detection," *Multimedia Tools and Applications*, vol. 76, no. 1, pp. 123–142, 2017.

[31] J. A. Rincon, J. Bajo, A. Fernandez, V. Julian, and C. Carrascosa, "Using emotions for the development of human-agent societies," *Frontiers of Information Technology & Electronic Engineering*, vol. 17, no. 4, pp. 325–337, 2016.

[32] J. A. Rincon, Â. Costa, P. Novais, V. Julian, and C. Carrascosa, "Using non-invasive wearables for detecting emotions with intelligent agents," in *International Joint Conference SOCO'16-CISIS'16-ICEUTE'16*, vol. 527 of *Advances in Intelligent Systems and Computing*, pp. 73–84, Springer International Publishing, Cham, Switzerland, 2017.

[33] F. Durupinar, J. Allbeck, N. Pelechano, and N. Badler, "Creating crowd variation with the OCEAN personality model," in *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS '08)*, vol. 3, pp. 1217–1220, Estoril, Portugal, May 2008.

[34] A. Mehrabian, "Analysis of the big-five personality factors in terms of the PAD temperament model," *Australian Journal of Psychology*, vol. 48, no. 2, pp. 86–92, 1996.