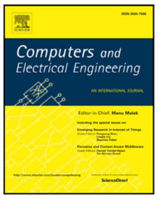
Computers and Electrical Engineering 58 (2017) 385–396

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng



Collaborative building of behavioural models based on internet of things

José Francisco Colom, Higinio Mora*, David Gil, María Teresa Signes-Pont

Specialized Processors Architecture Laboratory, Department of Computer Science Technology and Computation, University of Alicante, Alicante, Spain

A R T I C L E   I N F O

Article history:
Received 28 February 2016
Revised 22 August 2016
Accepted 23 August 2016
Available online 31 August 2016

Keywords:
Social internet of things
Big data
Embedded systems
Healthcare
Distributed system framework

A B S T R A C T

This paper proposes a new framework that takes advantage of the computing capabilities provided by the Internet of Thing (IoT) paradigm in order to support collaborative applications. It looks at the requirements needed to run a wide range of computing tasks on a set of devices in the user environment with limited computing resources. This approach contributes to building the social dimension of the IoT by enabling the addition of computing resources accessible to the user without harming the other activities for which the IoT devices are intended. The framework mainly includes a model of the computing load, a scheduling mechanism and a handover procedure for transferring tasks between available devices. The experiments show the feasibility of the approach and compare different implementation alternatives.

1. Introduction

The proliferation of Radio Frequency IDentification (RFID), Wireless Sensor Network (WSN) [1], and smart mobile technologies, among others, in a communicating-actuating network creates the Internet of Things (IoT), wherein sensors and actuators blend seamlessly into the environment around us, and the information is shared across networks [2]. The increase in the computing capabilities of these devices turns them into intelligent objects that can cooperate with each other to build distributed and autonomous ecosystems. This evolution creates cyber-physical systems that are beyond the interconnection of individual things and can provide joint and collaborative services.

The number of small computers in a user environment is increasingly large in the IoT paradigm. In a typical IoT scenario, user computers and devices usually fall into two categories, sometimes diffuse: (1) general purpose computers, designed for a wide diversity of tasks (i.e. laptops and desktop computers), and (2) specific purpose devices (i.e. wearables and smartphones). Computers falling into category (1) often have a variable computing resource availability depending on the dynamic load imposed by user activities. On the other hand, devices in (2) are intended to be used basically for specific applications and data capture. However, advanced applications arise, targeting to those devices (2), which require ever greater computing resources. Here we have, for example, the new virtual reality, gaming and health applications. These complex new applications can exceed the computing resources of the devices and produce delays and malfunctions. In these cases, the user devices quickly become obsolete and users must acquire more powerful devices in order to run the applications properly.

* Corresponding author.
E-mail addresses: jfcolom@dtic.ua.es (J.F. Colom), hmora@ua.es (H. Mora), dgil@ua.es (D. Gil), teresa@dtic.ua.es (M.T. Signes-Pont).

http://dx.doi.org/10.1016/j.compeleceng.2016.08.019
0045-7906/© 2016 Elsevier Ltd. All rights reserved.

This situation identifies the main challenge for the research carried out here. The proposal defines a collaborative processing framework to overcome this issue by sharing the application tasks among the available devices in the user environment. This approach is especially useful for critical applications such as health applications. In those domains, the systems do not provide high availability features but more than one unit can be used to improve the probability of accurately capturing a given variable and, therefore, a set of available devices can be operational in the users surroundings.

The main technical contribution of this work consists of the method described which allows for interaction and the distribution of the running of the applications among the IoT objects. As scientific contribution, this work proposes a formal framework wherein the performance and computing resource aspects of distributing the computation among a set of computers are specified. Moreover, the scheduling criteria is described on the basis of these specifications. The novelty of this work lies in considering the set of devices around the user as a cyber-physical system for collaborating in the computation of tasks. In addition, the orientation of the model to the area of healthcare creates a realistic working environment in which to implement the ideas for a social collaborative IoT.

The rest of the paper is organized as follows: First, we present a brief review of related research areas and their relevant architectural solutions. Then, we approach our framework by formally defining the problem and providing a conceptual view of the solution. After that, we address the experimental design and testing, where we verify the feasibility of the key framework elements. Finally, we draw the relevant conclusions and identify future directions for this research.

2. Related work

The following two subsections discuss the state-of-the-art of using the IoT paradigm for e-Health applications and data gathering from biosensors and other devices to perform computations. There are many research groups focussing their work on some of the topics covered in this work, which shows its significance within the scientific community and society as a whole. Only those most recent, relevant and representative results of the progress made are discussed. A final subsection is added which summarizes the main findings of the section and the contribution of this research to previous work.

2.1. E-health internet of things

IoT has become one of the most popular terms in recent years in the technology industry [3]. The IoT basically consists of embedding computing and communication capabilities into sensors and other *things* in the general environment. This evolution transforms these devices, making them smarter to fulfil their functions more effectively and perform other applications in order to improve the quality of life. There are numerous examples such as smart cities [4,5], sustainability [6], energy management, domotics and automation and other business models. The large increase in address space supplied by IPv6 is an important factor in the development of the IoT due to the greater ability it provides to address machines and connect them to the Internet.

The introduction of these new technologies in patient monitoring and diagnosis processes is increasingly important. In recent years we have seen a growing range of devices and sensors able to capture biomedical signals [7]. This includes specific-design sensors (a), new wearable devices (b) and even mobile smartphones (c).

(a) Sensors specifically designed for biological sensing, or biosensors, have led to a range of measurement techniques [8]. These elements not only provide measurement data, but are also appearing in analytical devices with signal processing capabilities to obtain advanced information on patients. They have traditionally been part of the medical equipment of hospitals and have been designed for use by medical staff.

(b) Wearable devices are a kind of electronic machine that interacts directly with the user [9]. These devices can be worn by people in the normal course of their activities, such as work, sport or resting. The first generation of wearables was basically used to alert the user (notifications of incoming emails, alerts, time, weather, etc.), but these devices are fast becoming a very popular technology. The acceptance of wearables by people has resulted in the creation of an industry of user-centric applications around these and the development of new features, especially biomedical sensing for health monitoring (respiration, electrocardiogram, oxygen blood saturation, skin temperature, etc.). Today, wearable health technology is also present in professional scenarios such as medical healthcare and professional sport.

(c) Mobile smartphones are mini portable computers held by users. These devices have a wide functionality and can run many applications, including, of course, talking on the phone. In addition, new smartphones have a lot of sensors of physical magnitudes such as the Global Positioning System (GPS), accelerometer, gravimeter and gyroscope which can be used to monitor human activity and peoples level of inactivity. Moreover, it is a fact that nowadays there are many *Apps* that can show the degree of physical activity in peoples lifestyles. For example, variables such as the speed of walking, running, climbing stairs and duration of physical activity can be measured by new healthy lifestyle Apps. The proliferation of smartphones and the high propagation rate of mobile devices connected among the population create a window of opportunity for the development of advanced health applications [10].

2.2. Collaborative processing

Collaborative processing is the way in which the computing of tasks can be shared among several connected devices to achieve a common goal. Some applications are incapable of being processed by the devices used to interact with them.

At present, the most popular way to outsource the processing costs is the Cloud Computing paradigm. This method allows computing resources to be managed in an efficient way since the processing can be carried out by remote servers on the Cloud.

Cloud Computing can deliver powerful applications to the users because the computing infrastructure is located in remote servers on the cloud and not in the local devices and sensors. There are currently proposals based on this principle which provide computing support for advanced healthcare applications. These proposals involve healthcare system architectures and service in IoT environments [11,12]. The main drawbacks of the cloud computing paradigm are the costs of communicating with remote servers and the lack of security when transmitting data from the devices through the network. This latter weakness is especially relevant in the case of patient biomedical data.

What connects mobile devices and cloud computing is the Mobile Cloud Computing (MCC) paradigm [13]. This method creates a computation model to address cases where the computing needs go beyond the capabilities of the mobile device. The applications workload is divided between devices and a central element located in the cloud. Thus, devices can move processing needs to the cloud (computation offloading) where they will run as services on Cloud Computing servers [14]. Distributed MCC applications are designed taking into account that additional computing resources can be provided from the Cloud in application runtime[15]. The work presented by [16] is the first thorough survey that analyses the state-of-the-art in Cloud-based Mobile Application Execution Frameworks (CMAEFs) taking into account the inherent limitations of the wireless channel that impedes the achievement of seamless execution in MCC ecosystems.

In terms of collaborative environments for connecting devices or sensors [17], several configuration scenarios can be established depending on the application requirements and hardware deployment. On the one hand, sensors can offload the computing to a central node in the network to centralize and process the data collected and, on the other hand, another way to increase, enhance and optimize computing capabilities is to share processing among them. This latter scenario can produce collaborative processing configurations between heterogeneous devices such as sensors on wearables and sensors on smartphones. Other configurations can be created using other connected devices with computing resources such as personal computers, workstations, IoT devices and even the Cloud. For this purpose, Cognitive Radio (CR) has emerged as an intelligent technology to meet the need to use the unoccupied spectrum band. This is very useful in the IoT and makes the matching and meeting of both CR and IoT possible [18].

E-health applications usually need intensive sensing and computing resources, so this is a good candidate for applications with collaborative processing. The integration of mobile devices and e-health applications has defined the *mHealth* paradigm where technology allows for interaction and information sharing to achieve a common goal. Recently, different approaches have been developed for communication between biomedical sensors and smartphones. For example, the system architecture developed by Renard et al. [19] exhibits two communication methods in relation to Type 1 Diabetes: the first is communication between the Continuous Glucose Monitor (GCM) sensor and the smartphone, and the second is communication between the smartphone and the insulin pump. Moreover, in the area of cholesterol measurements, Oncescu et al. [20] present a system that can be used to measure and track cholesterol levels directly on a smartphone through optimal image acquisition and processing. In addition, Yi et al[21]. demonstrate a communication architecture between the smartphone and different sensors such as a temperature sensor and electrocardiography. Another approach to communication between the smartphone and biomedical sensors is that presented by Fortino and Giampà [22] which includes Continuous Blood Pressure Measurement.

Finally, when there are a range of *things* offering services such as computing or any other services, one important issue to be addressed is service discovery. There are functionalities provided by devices that are designed to be real-world services that interact with other things. One innovative approach proposes interaction between objects following the Social Internet of Things (SIoT) paradigm based on a social network of intelligent objects and the notion of social relationships between things [23].

2.3. Findings

After reviewing the previous work, some findings can be drawn that justify our contributions to the state-of-the-art:

- Sensors have evolved from simple physical transductors to complex devices able to capture a lot of data. Consequently, computing power is required to transform the data and produce processed information. IoT devices are moving towards becoming smart things with communication and ever greater computing resources in order to run advanced applications.
- The Internet and Cloud platforms can offer promising computing solutions for smarter E-health applications. The Mobile Cloud Computing paradigm allows us to design methods for distributing the processing between devices and the Cloud. In this way, devices can offload some of the tasks to be executed to remote servers. These computing strategies require ongoing connectivity with the Internet to be able to maintain performance.
- The standard E-health application context is a collaborative environment made up of sensors and other embedded systems. In addition, other devices in the hands of users such as tablets, mobile phones, wearables, pads, etc. also have high computing capabilities. However, most of the time they are idle, which leads to an underuse of resources.

Against this background, this work is focused on creating collaborative processing strategies to overcome these issues by sharing application tasks among the available devices and Cloud platforms. The proposal described in the next section

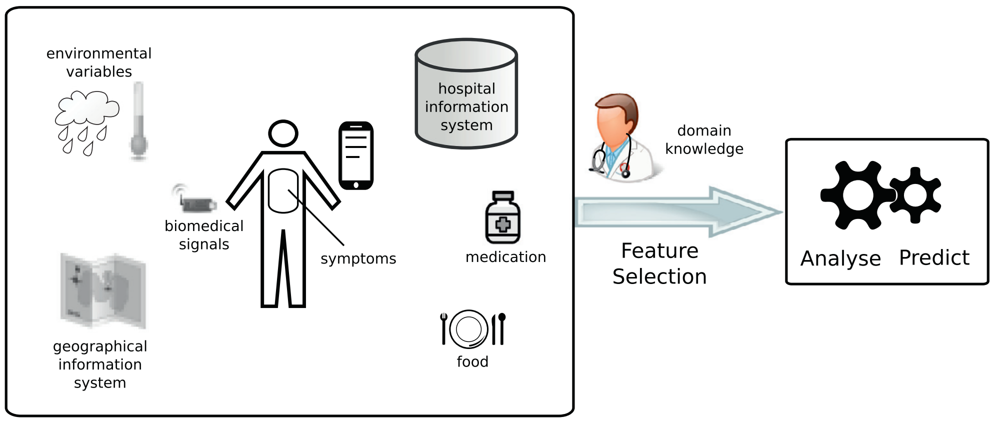


Fig. 1. Application example in immunological research domain.

introduces a formal framework for generalizing the use of computing platforms at several layers to perform the computations involved. This framework not only covers the collection of data in a distributed way, but also allows for sharing of the computation of the data at different layers (including the Cloud if necessary). This approach takes into account the ecosystem of devices around the user (not only smartphones) and allows us to define collaborative models depending on the requirements of the applications and the constraints for the available platforms. In this way, the framework facilitates interaction between elements, promoting the Social Internet of Things.

This approach should be especially useful in critical scenarios such as health applications since many powerful devices are available in the users surroundings. Other scenarios such as 'smart cities'can also benefit from this. In this kind of environment there are a lot of connected devices in the users surroundings which can interact and execute advanced applications in a collaborative way. Thus, this framework can be used to define cyber-physical systems and build more real environment models.

3. Collaborative behavioural model

In this section the proposed collaborative model is introduced. First, it is described the characteristic of the family of applications on which this research is focused; next, the formal framework and relevant elements taken into account in the proposal are defined; and finally, the method that allows a set of networked devices to share their resources in a collaborative way is set out.

3.1. Context application

The key features that define the context of this research are described below.

(a) Applications process big datasets coming from local sensor devices or services available over the Internet. Here, the word *big* must be understood on the basis of current *big data* thinking, taking into account not only the size of databases but other issues such as *variety* and *velocity*.
(b) Much of the data to be processed comes from sensors that collect information about their environment. In the field of healthcare, these sensors could be biomedical sensors or wearable devices.
(c) The application can be decomposed into a number of independent elements that we will call *services*. A service obtains input data by retrieving the results of other services and, at the same time, provides results for other services (service composition). Service processing could require reasonable computing power. For example, this could include advanced feature selection techniques and machine learning algorithms.
(d) Generally, there will be more than one device in the IoT user environment that can be used to run a given service. In fact, a number of general or specific-purpose computers may be active around the user, and these could dedicate part of their computing resources to the objectives of the application. The suitability of a device when it comes to running a given service or set of services will depend on its ability to satisfy service requirements in addition to the requirements of its corresponding dedicated tasks. Several variables must be taken into account here, such as processor load, free memory and free bandwidth, among others.

These features are of a general nature in order to add versatility to the framework. However, the research is mainly focused on the healthcare area because this application context is especially appropriate for the proposed collaborative framework. Fig. 1 shows a common scenario for this kind of application. Other scenarios may exist in well-connected cyber-physical systems such as smart cities, industrial environments, self-driving, etc. In these examples, there are a lot of con-

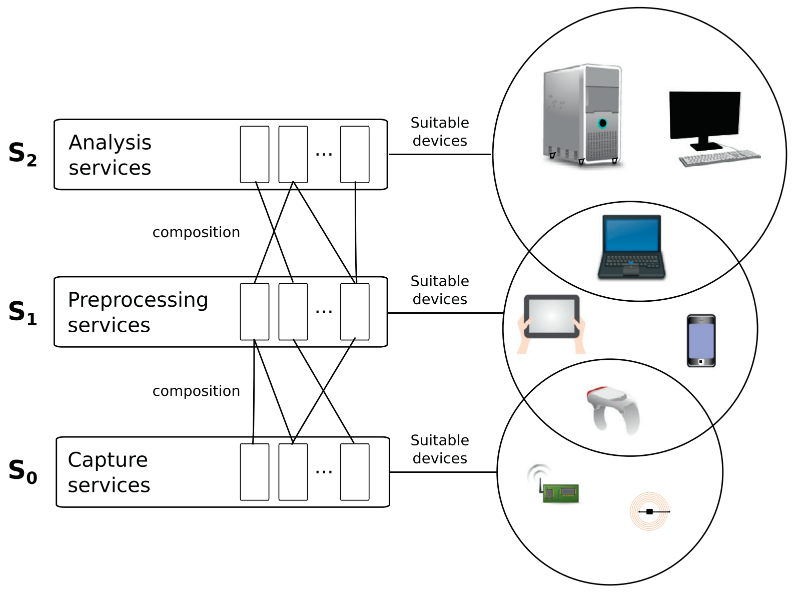


Fig. 2. Example of service decomposition in layers, and suitable devices.

nected devices in the users surroundings which can interact with them and run advanced applications in a collaborative way.

The working hypothesis of this research is that the design of a framework to perform the scheduling and sharing of the processing load can provide computing resources where necessary and take advantage of the network of connected devices in an IoT context. Other personal devices owned by the user, such as a smartphone, tablet or laptop, can also form part of this collaborative processing network. For example, in the case depicted in Fig. 1, the objective is to identify complex patterns in the patient environment that trigger allergic reactions or intolerance. In this application context, the IoT elements capture environmental and biomedical data, providing data flows. Other important information can be provided manually by the patient in the absence of suitable sensor devices. This information includes patient perceived symptoms such as itching, pain, wheezing, swelling, etc. Different processes involving feature selection, data mining techniques and machine learning, among others, are in place in order to build an analytic or predictive behavioural model, identifying complex patterns.

In terms of privacy and security issues, the proposed model is especially useful for maintaining privacy since the data can be computed inside the collaborative environment without having to leave the near application context. In addition, the computation can be done on elements owned by the user, by a trusted group of users, or at least on those controlled directly by a trusted institution inside a private domain. Therefore, sensitive information does not have to pass through a public network to a remote server to be processed. Nevertheless, standard distributed computing methods can be implemented to maintain security and privacy. In this way, a module can be included in the framework to manage the reliability, privacy and security of all data exchanged [24]. Furthermore, it may be appropriate to monitor the behaviour of both the users and the elements by establishing trust boundaries and using accounting methods.

3.2. Framework specification

In our proposed model, user devices and computers collaborate, providing their resources to help achieve the objectives of the application. In order to simplify the structure for communication between services, we organize services into different layers: services obtain input data for their operation only from the services in lower layers.

Fig. 2 shows a possible breakdown of services, and a number of suitable devices for each level. This example shows an organization of activities that could be used to obtain valuable knowledge extraction from raw data coming from sensors.

Generalizing Fig. 2, the application is broken down into n layers,

$$\{S_0, S_1, \cdots, S_{n-1}\}$$

where each layer $(S_i)_{i=0}^{n-1}$ is a set of services. Services in S_0 will perform data capture. Each service in the remaining layers, $(S_i)_{i=1}^{n-1}$ will process data coming from some service in S_{i-1} (composition).

To distribute resources in an efficient way, the computing load must be properly characterized. For this reason, we define L as a vector domain of relevant features modelling the computing load. The individual features are expressed using normalized scalar values. For example, L could be defined as

$$L = B \times P \times M \times E$$

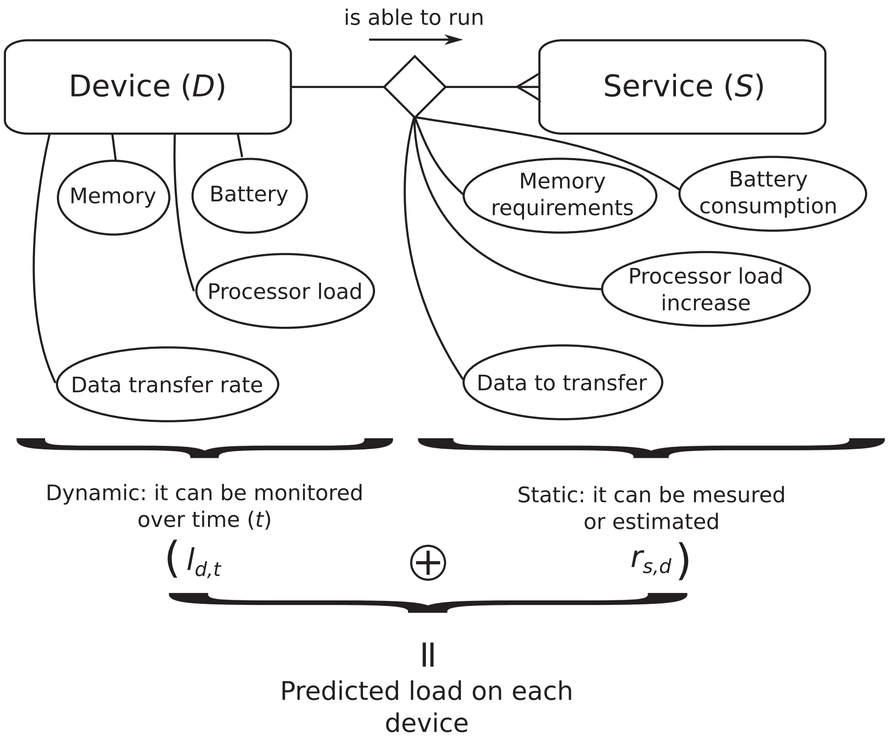


Fig. 3. Predicted load from the current monitored status and the estimated impact of running each service.

where B is the transfer rate (fraction of the total bandwidth used in a defined time interval), P is the processor load (average number of processes sharing the processor in a defined time interval), M is the memory in use (fraction) and E is the battery consumption (fraction of the total battery consumed in a defined time interval).

In addition, the application is going to be run using a set D of m devices. Each device $d \in D$ shows a state, which is a vector $l_{d,\,t} \in L$ describing its computing load, and therefore its ability to run other processes. Obviously, these values change over time t, depending on the different activities with which the device is involved.

For each device and service, we need to estimate the computing load that would be added to the device if it were to run the service. This increase can be represented by $r_{s,\,d} \in L$ (requirements of service $s \in S_i$ in device $d \in D$). This vector can be estimated empirically by testing each service on each device.

The values in $r_{s,\,d}$ model the suitability of a device for a given service. For example, a laptop computer with a powerful Graphic Processing Unit (GPU) will generally show a low r for services requiring heavy image processing; in other words, this quantifies the degree to which the computer with GPU is suitable for image processing services.

Next, we define an internal binary operator in L, and we represent this using the symbol $\oplus$ and $\Sigma^{\oplus}$ for the accumulation. This specifies the concrete procedure for adding the service requirements to the current device load, allowing us to predict the load of a device if a given service is run on it. The model described is depicted in Fig. 3.

Finally, for the set of m devices in D we define a feasibility function

$$f : L^m \longrightarrow \{0, 1\}$$

modelling the real possibility of working with a given load. For example, a service cannot be assigned to a device if there is not enough free memory for it.

The scheduling problem can then be stated by finding a correspondence for each layer $(S_i)_{i=1}^{n-1}$

$$schedule : S \longrightarrow D$$

satisfying the expression 1 (for a given time t):

$$f\left(\forall d \in D\left(l_{d,t} \oplus \sum_{\forall s|schedule(s)=d}^{\oplus} r_{s,d}\right)\right) = 1 \tag{1}$$

In addition, we can set a more ambitious goal, by defining an objective function

$$g : L^m \longrightarrow [0, 1]$$

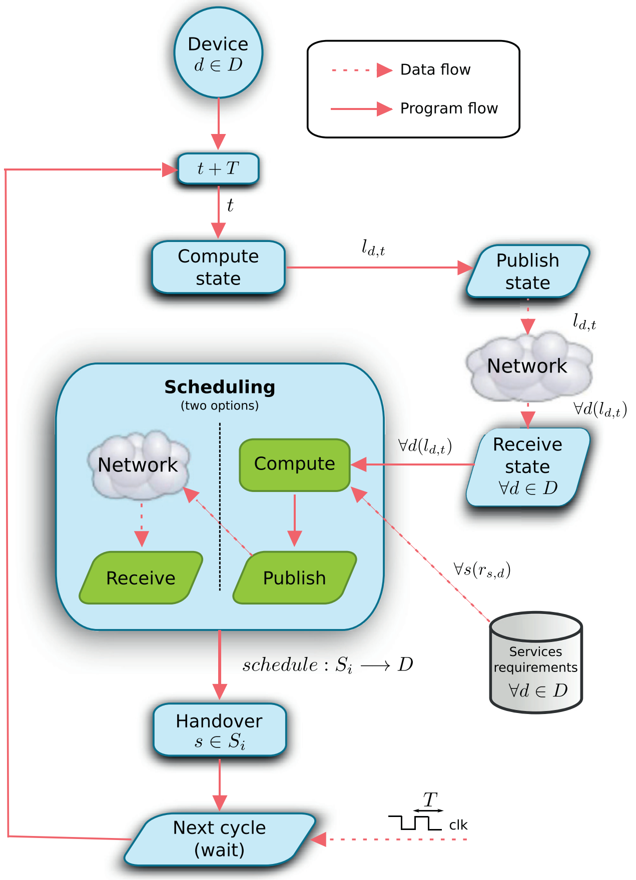


Fig. 4. Flow diagram for the proposed framework.

instead of just a feasibility function, and then finding the correspondence that maximizes the value of the expression 2.

$$g\left(\forall d \in D\left(l_{d,t} \oplus \sum_{\forall s|schedule(s)=d}^{\oplus} r_{s,d}\right)\right) \tag{2}$$

3.3. Scheduling method

The key aspect of the framework is how the available computing layers can collaborate to meet the application requirements taking into account the computing load features. This subsection presents the scheduling method to implement the framework. An explanation of the key steps and different design options is provided. Fig. 4 shows a flow diagram describing the procedure run by the proposed framework.

3.3.1. Compute state

Each device must compute its own computing load modelled according to *L*. Some features can be computed with the help of operating system routines, if there is one available. For example, operating system kernels can generally provide the number of free memory blocks or the average processor load. These data must be properly normalized to create a good model of the current computational load. For example, memory resources may be computed as

$$memory\ use = \frac{busy\ memory}{total\ memory}$$

and processor use as

$$processor\ use = \frac{average\ processor\ load}{maximum\ average\ load\ allowed}$$

Other parameters could require some specific techniques. For example, the current data transfer rate produced by the device can be accurately determined. In addition, the data transfer rate due to other communications taking place in the shared medium can be obtained by enabling the monitor mode in wireless interfaces [25]. However, the available free bandwidth is a difficult point in wireless and shared medium networks, and this information is relevant to model the suitability of a device to accept bandwidth-hungry tasks. One easy solution is to estimate some average values and take these into account. Other more elaborated approaches involve periodic measurements with specific tools.

3.3.2. Publish state and receive state

The status of each device involved must be shared with the remaining devices. For network communication, broadcast or multicast transfer will make the most of the shared medium. By working this way, all devices will have the information required for computing the schedule. In addition, the *receive* operation can be used as a synchronization primitive, so all devices start computing the schedule at the same time.

3.3.3. Scheduling

At this point the problem specified in expression 1 or the more elaborate expression 2 in Section 3.2 must be solved. Computer science has historically provided a wide range of algorithms for this type of problem, ranging from resource-hungry ones reaching optimal solutions, to more efficient ones achieving suboptimal results.

Two strategies are considered here: (1) all devices compute the schedule vector, avoiding communication latencies, or (2) only the fastest devices compute the schedule and the result is shared across the network with the remaining devices. The best alternative will depend on the complexity of the algorithm, the network latency, and the size of the problem to solve (number of services and number of devices). The Section 4 will focus on this issue.

3.3.4. Handover

As a result of the scheduling process, the device must offload some services and perhaps run new ones. In simple cases, this just means kill and start-up processes. However, for some complex services, information about the state of the service must be exchanged between devices. This exchange is specified in the handover procedure.

3.3.5. Next cycle

The overall process is repeated every period T. A synchronization mechanism is needed at this point, and several solutions are possible. One approach is to implement a synchronization barrier using network primitives. Another solution is to use a common clock, such as the one provided by the Network Time Protocol (NTP).

4. Experimental design and results

The experiment conducted aims to show that the proposed framework can successfully run on embedded devices and provide valuable results: the flexible distribution of processing among the overall set of user devices. To accomplish this objective we select the most computationally complex section of the proposed method, scheduling. We test and compare two well-known approaches to the scheduling problem which fit the framework provided (see section 3.3.3). In addition, we simulate an application environment to test the feasibility of the overall approach.

4.1. Test environment

For the purpose of the test, we deploy different scheduling algorithms on several sensor devices and a server computer. The sensor device role is played by a low-cost, single-core, credit card-sized computer: Raspberry Pi, [1] with an ARMv6-compatible processor and 512MB of main memory. For the server we use a desktop computer, with an Intel® Core™ i5 CPU and 6GB of main memory. The computers are connected through a standard wireless local area network, performing at 28 Mbits/s.

We simulate part of a system for immunological research support, taking into consideration five services in the pre-processing layer (all these services prepare sensor data for upper layers).

- s_0: Get the current temperature and humidity by querying a sensor if available or by using an Internet service.
- s_1: Average the pulse oximetry values obtained from a wearable biosensor.
- s_2: Ask the patient to set the level of perceived itching using the smartphone display.
- s_3: Retrieve the current amount of medication prescribed from the hospital information system.
- s_4: Get the current height above sea level from a GPS device

Those services or their variants can be usually found in common health monitoring applications.

In addition, we assume that the patient monitored is in an environment where they are surrounded by a variable number of computing devices, ranging from 3 (personal devices) to 10 (when other users add their devices to contribute to the healthcare research).

For performance evaluation, we take $L = \textit{Processor load}$ (average number of processes sharing the processor in a defined time interval). Therefore, the proposed method is expected to properly distribute the services among the devices according

[1] Raspberry Pi is a trademark of the Raspberry Pi Foundation.

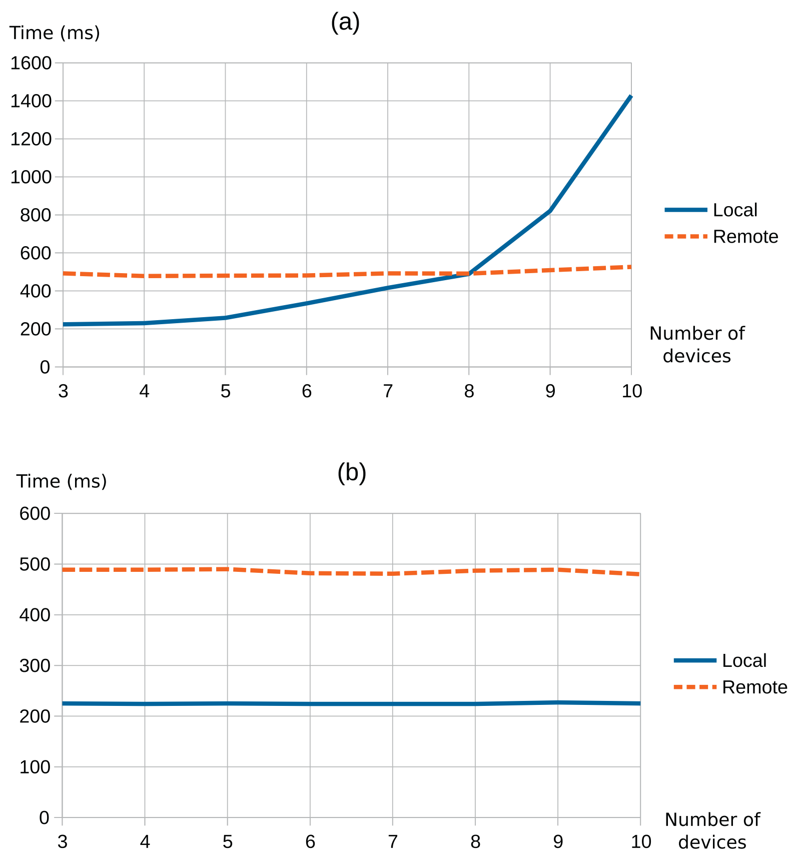


Fig. 5. Response time for the two scheduling algorithms when run locally on a device or remotely on a server: (a) optimal solution; (b) suboptimal solution.

to the defined processor load criteria. The predicted impact on the load of each device for each service ($r_{s,\,d}$) is estimated according to common patterns.

Finally, we simulate some additional loads on user devices coming from user activities by running audio reproduction tasks on the devices.

4.2. Scheduling algorithms

We implement two solutions for the problem stated by the expression 2 in Section 3.2, using the following objective function:

$$g\left(l_{d_1}, l_{d_2}, \cdots, l_{d_m}\right) = \frac{1}{\sigma_{i=1\ldots m}(norm(l_{d_i}))}$$

where $l_{d_i} \in L$ is the predicted processor load, $norm : L \longrightarrow [0, 1]$ is a weighted average of the components in L (in this test, it is just a quantification of the processor load), and σ is the standard deviation. This objective function aims to distribute services equitably among available devices.

First, we test a state space search with a time complexity $O(m^s)$, where m is the number of devices and s is the number of services. This procedure always obtains an optimal solution according to the selected objective function. Second, we run a greedy algorithm, directly assigning each service to the device on which the estimated impact will be the lowest. This procedure generally obtains a good suboptimal solution with better time complexity $O(sm\log m)$ (it still can be improved to $O(sm)$ by pre-sorting the estimated impact on each device for each service, $r_{s,\,d}$)

4.3. Results

Fig. 5 shows the response time of the two scheduling algorithms. For each algorithm, we compare the execution time when running the scheduling algorithm on the sensor device, with the one obtained when retrieving the solution from the server.

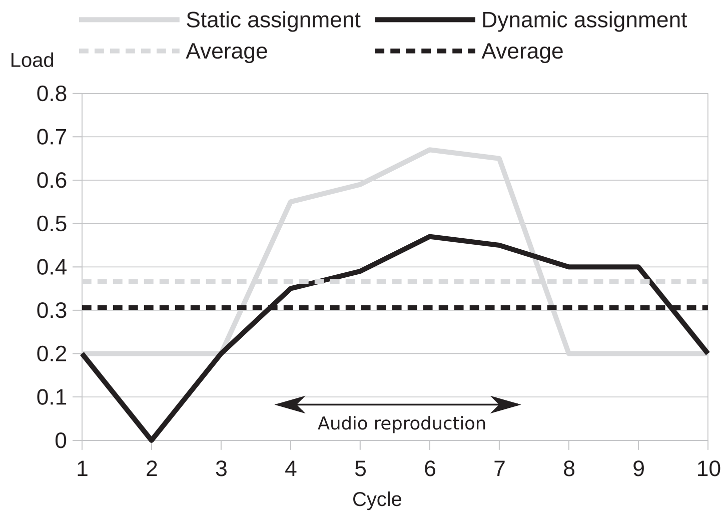


Fig. 6. Evolution of the processor load of a particular user device in *D* during 10 scheduling cycles. The user device also plays an audio stream from cycle 4 to 7. The dynamic assignment is done by the proposed architecture using the local suboptimal scheduling.

As can be seen, when searching for the optimal solution (section **a** of Fig. 5), running the scheduling algorithm locally in the device is recommended if the number of devices is fewer than 9. Otherwise, retrieving the solution from network resources is clearly preferred. In contrast, when just computing a suboptimal solution (section **b** of Fig. 5) we find that local execution is the more efficient approach. The results clearly show the feasibility of the polynomial scheduling solution; the exponential solution is not feasible in all cases, only in situations with a small number of devices.

Next, Fig. 6 shows the result of the overall simulation from the point of view of a particular device (the local execution of the polynomial scheduling solution has been chosen for this test) from a set of six available devices. The normalized processor load monitored in each cycle and the corresponding average is shown for two tests: dynamic assignment, using the proposed method; and static assignment, where the device is determined to be in charge of running a particular service. Moreover, an additional task (audio reproduction) is performed during the experiment to identify the corresponding changes in the scheduling decisions.

The results demonstrate the viability of the approach, showing how an application can take advantage of the surrounding collaborative devices to reduce the processor load. The overload introduced by the logic of the proposed method is more than compensated for by the benefits from dynamic assignment and the adaptive behaviour.

5. Conclusion and future work

We have designed a novel framework that enables collaboration among devices in IoT environments, allowing efficient resource sharing to achieve the objectives of a set of applications. The proposed solution is especially suitable for applications supporting research in the healthcare area, where small resource-constrained computers controlled by patients must accomplish certain results in a privacy-aware way.

Our experiments show the feasibility of the approach, and also provide insights into some future areas of work. Here, we can expand the framework to include a flexible scheduling method, allowing devices to switch the mode (local or remote) under which the scheduling algorithm is run according to network conditions and device load.

Another interesting area of future work is system resilience. The distributed nature of the proposed solution offers some degree of system resilience as a secondary effect. However, specific design considerations can be put in place in order to guarantee that all services are run properly with enough system resources in the event of a system failure.

Acknowledgements

This work has been partially funded by the Spanish Ministry of Economy and Competitiveness (MINECO/FEDER) under the granted Project SEQUOIA-UA (Management requirements and methodology for Big Data analytics) TIN2015-63502-C3-3-R, by the University of Alicante, within the program of support for research, under project GRE14-10, and by the Conselleria de Educación, Investigación, Cultura y Deporte, Comunidad Valenciana, Spain, within the program of support for research, under project GV/2016/087.

References

[1] Rehmani MH, Pathan A-SK. Emerging communication technologies based on wireless sensor networks: Current research and Future applications. CRC Press; 2016.

[2] Gubbi J, Buyya R, Marusic S, Palaniswami M. Internet of things (iot): a vision, architectural elements, and future directions. Future Generation Comput Syst Int J Grid Comput Escience 2013;29(7):1645–60.
[3] Borgia E. The internet of things vision: key features, applications and open issues. Comput Commun 2014;54:1–31.
[4] Mora Mora H, Gilart Iglesias V, Gil D, Sirvent Llamas A. A Computational architecture based on rfid sensors for traceability in smart cities. Sensors 2015a;15(6):13591–626.
[5] Lau SP, Merrett GV, Weddell AS, White NM. A Traffic-aware street lighting scheme for smart cities using autonomous networked sensors. Comput Electr Eng 2015;45:192–207.
[6] Gilart Iglesias V, Mora H, Pérez-delHoyo R, García-Mayor C. A Computational method based on radio frequency technologies for the analysis of accessibility of disabled people in sustainable cities. Sustainability 2015;7(11):14935–63.
[7] Santos DFS, Almeida HO, Perkusich A. A Personal connected health system for the internet of things based on the constrained application protocol. Comput Electr Eng 2015;44:122–36.
[8] Kissinger PT. Biosensors a perspective. Biosens Bioelectron 2005;20(12):2512–16.
[9] Lara O, Labrador M. A survey on human activity recognition using wearable sensors. IEEE Commun Surv Tutor 2012;1:1–18.
[10] Lee YS, Cho SB. Activity recognition with android phone using mixture-of-experts co-trained with labeled and unlabeled data. Neurocomputing 2014;126:106–15.
[11] Xiao ZR, Lv BG, Wang X, Zhao YJ. A healthcare service system based on internet of things. Adv Mater Res 2013;774–776:1903–7.
[12] Guo B, Dong B, Zhang X, Yang J, Wang Z. Based on the improved internet of things architecture. Int J Smart Home 2015;9(9):51–62.
[13] Fernando N, Loke SW, Rahayu W. Mobile cloud computing: a survey. Future Generation Comput Syst 2013;29(1):84–106.
[14] Abolfazli S, Sanaei Z, Ahmed E, Gani A, Buyya R. Cloud-based augmentation for mobile devices: motivation, taxonomies, and open challenges. IEEE Commun Surv Tutor 2014;16(1):337–68.
[15] Mora Mora H, Gil D, Colom López JF, Signes Pont MT. Flexible framework for real-time embedded systems based on mobile cloud computing paradigm. Mob Inf Syst 2015b:1–14.
[16] Ahmed E, Gani A, Khan MK, Buyya R, Khan SU. Seamless application execution in mobile cloud computing: Motivation, taxonomy, and open challenges. J Netw Comput Appl 2015a;52:154–72.
[17] Rashid B, Rehmani MH. Applications of wireless sensor networks for urban areas: a survey. Elsevier J Netw Comput Appl 2016;60C:192–219.
[18] Khan A, Rehmani M, Rachedi A. When cognitive radio meets internet of things?. IWCMC; 2016.
[19] Cobelli C, Renard E, Kovatchev B, Keith-Hynes P, Brahim N, Place J, et al. Pilot studies of wearable outpatient artificial pancreas in type 1 diabetes. Diabetes Care-Alexandria 2012;35(9):65–7.
[20] Oncescu V, Mancuso M, Erickson D. Cholesterol testing on a smartphone. Lab Chip 2014;14(4):759–63.
[21] Yi WJ, Jia W, Saniie J. Mobile sensor data collector using android smartphone. In: IEEE international midwest symposium on circuits and systems; 2012. p. 956–9.
[22] Fortino G, Giampà V. Ppg-based methods for non invasive and continuous blood pressure measurement: an overview and development issues in body sensor networks. In: IEEE international workshop on medical measurements and applications; 2010. p. 10–13.
[23] Atzori L, Iera A, Morabito G. Siot: giving a social structure to the internet of things. IEEE Commun Lett 2011;15(11):1193–5.
[24] Atzori L, Iera A, Morabito G. The internet of things: a survey. Comput Netw 2010;54:2787–805.
[25] Ahmed E, Yaqoob I, Gani A, Imran M, Guizani M. Internet of things based smart environments: state-of-the-art, taxonomy, and open research challenges. IEEE Wireless Commun 2015b:Accepted.

José Francisco Colom received a PhD in computer science from the University of Alicante in 2016. He currently works in teaching and research activities in the Department of Computer Science Technology at the same university. His current areas of research interest include computer networks, distributed systems, internet of things, cyber-physical systems and computer security.

Higinio Mora received a PhD in computer science from the University of Alicante in 2003. Since 2002 he has been an associate professor and researcher in the Specialized Processors Architecture Laboratory. His areas of research interest include computer modelling, computer architectures, high performance computing, embedded systems, internet of things and cloud computing paradigm.

David Gil is an assistant professor at the Department of Computing Technology at the University of Alicante. His main research topics include artificial intelligence applications, data mining, open data, big data, medical decision support systems and cognitive sciences. He has participated in numerous national and international projects and agreements with private companies and public organizations related to his research topics.

María Teresa Signes Pont received a PhD in Computer Science from the University of Alicante in 2005. Since 1996 she has been a member of the Computer Science and Technology department of the same university where she is currently an associate professor and researcher. Her areas of research include computer arithmetics, natural systems modelling and cyber-physical systems.