IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 23, 2018, accepted February 24, 2018, date of publication March 5, 2018, date of current version March 28, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2811900

PriorityNet App: A Mobile Application for Establishing Priorities in the Context of 5G Ultra-Dense Networks

FRANKS GONZÁLEZ-LANDERO[1], IVÁN GARCÍA-MAGARIÑO[2,3], RAQUEL LACUESTA[2,3], (Member, IEEE), AND JAIME LLORET[4], (Senior Member, IEEE)

[1]Edison Desarrollos, 44002 Teruel, Spain
[2]Department of Computer Science and Engineering of Systems, University of Zaragoza, 44003 Teruel, Spain
[3]Instituto de Investigación Sanitaria Aragón, 50009 Zaragoza, Spain
[4]Instituto de Investigación para la Gestión Integrado de Zonas Costeras, Universitat Politecnica de Valencia, 46022 Valencia, Spain

Corresponding author: Jaime Lloret (jlloret@dcom.upv.es)

This work was supported in part by the Construcción de un framework para agilizar el desarrollo de aplicaciones móviles en el ámbito de la salud through the University of Zaragoza and the Foundation Ibercaja under Grant JIUZ-2017-TEC-03, in part by the Program—Estancias de movilidad en el extranjero José Castillejo para jóvenes doctores through the Spanish Ministry of Education, Culture and Sport under Grant CAS17/00005, in part by the Universidad de Zaragoza, Fundación Bancaria Ibercaja, and Fundación CAI in the Programa Ibercaja-CAI de Estancias de Investigación under Grant IT24/16 and Grant IT1/18, in part by the Research Project—Desarrollo Colaborativo de Soluciones AAL through the Spanish Ministry of Economy and Competitiveness under Grant TIN2014-57028-R, in part by the Organismo Autónomo Programas Educativos Europeos under Grant 2013-1-CZ1-GRU06-14277, in part by the Fondo Social Europeo and the Departamento de Tecnología y Universidad del Gobierno de Aragón under Grant Ref-T81, and in part by the Ministerio de Economía y Competitividad in the Programa Estatal de Fomento de la Investigación Científica y Técnica de Excelencia, Subprograma Estatal de Generación de Conocimiento within the project under Grant TIN2017-84802-C2-1-P.

ABSTRACT The devices and implementations of 5G networks are continuously improving, and people will probably use them daily in the near future. 5G networks will support ultra-dense networks. In the literature, several works apply 5G networks in smart cities and smart houses. One of the most common features of these works is to use priorities in tasks, such as the management of electrical consumption at houses, waste collection in cities, or pathfinding in self-driving cars. The proper management of priorities facilitates that urgent service requests are rapidly attended. However, to the best of our knowledge, the literature lacks appropriate mechanisms for considering users' priorities in the 5G ultra-dense networks. In this context, we propose a mobile application that allows citizens to request smart city services with different priority levels. The experiments showed the high performance of the app and its scalability when increasing priority list sizes. This app obtained 72.3% of usability in the system usability scale and 82.9% in the ease-of-use dimension of the usefulness, satisfaction, and ease of use questionnaire.

INDEX TERMS Smart city, priority, ultra-dense network, mobile application, app.

I. INTRODUCTION

Nowadays each citizen commonly has a mobile device. The amount of interchanged information usually increases due to the active use of clouds, document repositories and streaming-video servers. It is predicted that the number of connected mobile devices will exceed 11.5 billion in 2019. In addition, 5G ultra-dense networks [1] will probably efficiently support all these connections, especially in smart cities with a high population density.

5G technologies are advancing in several aspects such as (a) the reduction of energy consumption, (b) the improvement of signal reachability in wall trespassing [2], and (c) the management of networks for rapidly transferring big data. The demand of data traffic is increasing steeply for transferring large files such as high-resolution videos, which have up-growing popularity [3]. In this line of work, a software-defined networking (SDN) approach improved the fairness in streaming video so that users get this service with a similar quality level [4]. In addition, an intelligent handover process algorithm guaranteed load balance in 5G networks, and improved the quality of service (QoS) in the upload of videos from mobile cameras in environmental surveillance [5]. 5G networks are also useful in fields that need real-time responses such as e-health monitoring [6]. In general, 5G networks are based in some pillars such as SDN, network function virtualization (NFV) and mobile edge computing (MEC) [7].

Smart cities need to urgently attend some service requests, such as waste collection in some critical locations like hospitals, chemical factories and schools [8].

Our previous work shows that priorities self-reported by citizens can be fairly managed even when there are people trying to take advantage of the priority system [9]. However, in order to allow citizens to easily self-report the priority levels of their requests, an easy-to-use app could be useful for final users. This work addresses this problem by proposing a novel app that allow users to define their prioritized lists of service requests.

The remainder of the paper is organized as follows. The next section introduces the related work, highlighting the gaps of the literature that the current work addresses. Section III presents the current approach introducing the novel PriorityNet app. Section IV describes the experiments about the performance and usability of the app. Finally, section V mentions the conclusions, and depicts some future lines of research.

II. RELATED WORK

The most relevant related works fall into the three categories of (a) smart cities with their communications and smart services discussed in II-A, (b) solutions for supporting ultra-dense networks introduced in section II-B, and (c) applications that manage priorities for tasks or services presented in section III-C.

A. SMART CITIES

Smart cities usually need to communicate large amounts of data. Some of the smart city services are real time, and need short latencies. Hence, an adequate networking performance generally has a positive effect on the citizens quality of life.

To begin with, [8] proposed four dynamic models for collecting waste in smart citizens. In summary, a smart city had a fleet of trucks and several trash bins. Each bin had a device for tracking when it was full. The models distinguished between two types of bins, which were the high priority bins (HPBs) and the other ones. The HPBs were located in places such as hospitals, factories and high schools. The models were the Dedicated Trucks Model (DTM), Detour Model (DM), the Minimum Distance Model (MDM) and the Reassignment Model (RM). They described how trucks collected waste from HPBs in each model. Each model has its mechanism of reassigning routes. They evaluated the models in Saint Petersburg (Russia), and all of them had advantages and disadvantages. A set of simulations showed the utility of their approach and the proper functioning of the route reassignment mechanisms. They mentioned the possibility of expanding their work by taking truck capacities into account. However, this work did not allow users to change the priorities of certain bins. For example, this could be useful for requesting the collection of a really risky material, like an extraordinarily highly toxic waste from a chemical factory.

Moreover, [10] introduced some of the smart city bases such as 5G networks, Internet of Things (IoT), cloud of things and advanced artificial intelligence. Smart cities normally had smart homes. 5G technologies were used for two purposes. The first one was to let people communicate with their smart homes. For example, they could use their mobile devices for remotely scheduling the tasks of IoT appliances. The second purpose was to facilitate the transfer of huge amounts of data with clouds. In the future, it will be necessary to have more bandwidth for performing such operations. Lastly, smart cities were expected to collect information from citizens in order to take better decisions for improving their quality of life. For instance, a smart city could analyze electrical consumption in each house, and provide customized recommendations for saving money. More concretely, similar approaches can be adopted to specific buildings. For example, [11] presented a case study about products for improving security in smart buildings. In smart homes, this work used surveillance devices such as motion detector, sensors for opening/closing doors, and presence sensors. All of these were connected to Internet so that users could check their homes with their mobile devices. Nevertheless, none of these works proposed an app for establishing priorities for most urgent matters. In the case of surveillance, the transmission of some cameras may need a higher priority in some situations such as robberies.

Furthermore, [12] presented a theoretical model about self-driving connected cars. This model was based on the theory of multi-agent system. Each vehicle was represented as an individual agent in simulations. They followed a Belief-Desire-Intention (BDI) approach [13], in which each agent was aimed at satisfying its desires by completing some specific goals. The connected cars were an innovative solution for reducing collision risk, avoiding economic cost of crashes and saving humans lives. Furthermore, disabled people, elders or people without valid driver's license could safely travel long distances. Moreover, vehicle-to-vehicle (V2V) communication were necessary for supporting this self-driving model. Thanks to V2V communications, cars could potentially exchange information about their positions, speeds, routes, plans for changing speed or lane, turning, stopping and so on. This model was simulated with Qt Framework 5.4, and it worked properly. However, this model would need to consider more aspects to be applied in real world, such as the possibility that large amounts of messages could overload V2V networks and the limitations of the city infrastructures. Hence, 5G networks could be useful for efficiently transferring these amounts of data avoiding losses. Nonetheless, this work did not mention the possibility of establishing priorities for assuring the safe travel in self-driving cars. A proper mechanism of prioritization could overcome this barrier. For example, the communications of some cars could be prioritized when transporting patients in critical health situations.

Therefore, in most of these works, an app could be useful for allowing users to manually change the priorities in some services in real time.

B. ULTRA-DENSE NETWORKS

In the current society, the use of mobile device connections is increasing rapidly. In this context, the field of ultra-dense networks is aimed at supporting dense sets of connection points. Several works have discussed new technological advances in ultra-dense networks. For instance, [2] introduced the variety of 5G technologies with cognitive radio (CR). CR is a mechanism for dynamically selecting the best wireless channels according to different criteria for avoiding congestions and interferences. They addressed the challenge of saving energy in smart home networks (SHNs), considering the spread of WiFi signal. [14] mentioned that devices needed a high amount of energy to emit signals for communications through obstacles. CR was planned to be implemented with 5G technologies. They made a simulation of a SHN with and without CR, and the energy consumption was reduced in 26%.

Several works about ultra-dense networks focus on transferring files with large amounts of data. For example, [15] proposed a new framework that was able to support huge data traffics requested by users. This framework was called Big Data Driven. It focused on providing a high quality of experience to users with their mobile devices and wearable sensors. Basically this model reallocated resources to manage data traffic based on the user locations. Moreover, this framework also reduced some costs by analyzing data from users. In addition, [3] presented a new paradigm called Information-centric-networking (ICN). This paradigm was aimed at retrieving videos from in-network caches. ICN reduced network traffic and video retrieval delay significantly. However, these kinds of applications may have a low priority when sharing the network with other more critical services (e.g. critical health situations and network failures).

Reference [16] proposed a mechanism for improving communications among neighbor cells. It was based on a reward mechanism for promoting collaboration. Their approach considered real-time priority levels and QoS.

In conclusion, ultra-dense networks usually need intensive transfer of data. Some services may need urgent responses. These urgencies can dynamically vary depending many factors. An easy-to-use app could be useful to allow users to indicate these changing priorities in smart cities.

C. PRIORITY APPLICATIONS

The priorities of services and tasks have been widely used in many fields. For instance, [17] proposed a cultivation priority planning based on the needs of food and the locations provided by a geographical information system (GIS). Their approach considered several aspects such as soil depth, climate, pH, and the existence of certain minerals in the land. They applied a fuzzy approach and an analytical hierarchy process to assign the cultivation type to the different land areas.

Moreover, in the field of parallel computing, [18] proposed to use user-assigned priorities for job scheduling. Their approach was based on their ReShape framework that supported resizing parallel applications. The priorities were used to take informed decisions about changing the number of processors assigned to each job. They applied their framework in three case studies, and they obtained improvements of the execution time and higher utilization of the existing resources.

In the public transportation area, [19] used several flexible priority rules for assigning passengers to trains and seats. Their approach had low computational times. They did not significantly increase the level of unattended requests or travel delays. However, they increased the variability of passengers. In this manner railway operators were able to test different policies about passenger priorities.

The Priority-based Application-Specific Congestion Control Clustering protocol (PASCCC) [20] used priorities for managing network communications. PASCCC used a clustering approach for detecting congestions in the network and avoiding these areas. Some of the communications needed to take larger paths in networks for avoiding congestions. The priorities were used to select which communications used the shorter paths and which ones the larger ones. In addition, [21] proposed to use priorities for managing services through sensor networks in general.

Nevertheless, none of these works explicitly provided an easy-to-use mobile application for allowing final users to establish their own priorities. The next section introduces a novel app that covers this gap of the literature.

III. PRIORITYNET: AN APP FOR SETTING PRIORITIES IN ULTRA-DENSE NETWORKS

PriorityNet app is a tool that allows users to easily assign priorities to certain service requests. This tool was developed considering usability principles.

Figure 1 depicts the functionality of the app and some of its possible utilities. Users can determine priority lists for themselves of companies. The app connects with the system and uploads these priority lists. In this way, a user could set the priorities in several service requests of a smart home, such as the ones related with security cameras (e.g. when something suspicious had been observed), turning on/off electrical appliances, or download video streaming in real time with a high quality. In smart cities, the priorities could be used for reassigning routes in self-driving cars or trucks that collect dangerous waste.

A. USER INTERFACE

Figure 2 presents the main screen of the app, which allows users to define priority lists. This screen has two vertical lists. The list in the right side contains certain services, including the ones from the smart city or the use of the network for certain mobile applications. Priorities can be assigned for all

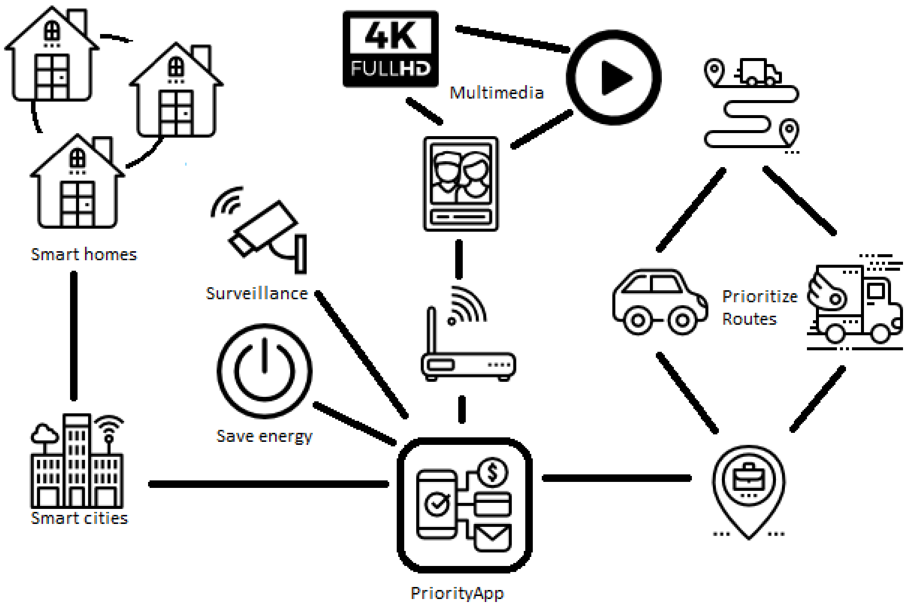

FIGURE 1. Overview of the priority system of PriorityNet app.

FIGURE 2. Main screen of PriorityNet App.

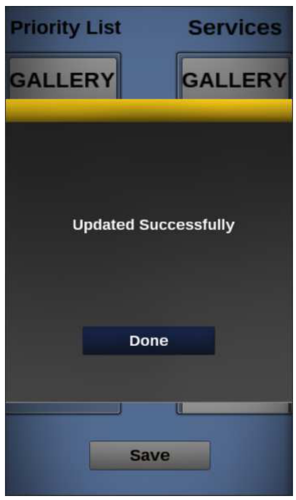

FIGURE 3. Confirmation message in PriorityNet app.

these services and mobile applications. The priority order is established in the left list by dragging and dropping elements from the right-side list. The services with more priority are placed at the top of the list and the ones with less priority are placed at its bottom.

The use of this application is summarized in three steps. In the first step, the user decides the service to which they assign higher priorities. In the second step, the user drags and drops an item from the right list to the left one. In the last step, the user can re-sort the the priorities of the list of services, if they want to.

When a user has finished assigning priorities to services, they can touch the "Save" button, and the priority list is transferred to the system. Figure 3 shows an example of a confirmation message.

B. DESIGN AND IMPLEMENTATION

As one can observe in the sequence diagram of Figure 4, users can see all available services. In this step, they must drag one service and drop it in a priority list, so that this priority list is created. The sequence diagram has a squared yellow background in order to indicate that this operation part continues as long as the user wants. Hence, it is possible to rearrange priorities, and adding/deleting services to/from the list.

When the user touches the 'Save' button, the system receives input from the priority list, converts it into a JavaScript Object Notation (JSON) file, and sends it to a database management system (DBMS). JSON is a file format for sending information to DBMSs. The DBMS can be running either in a cloud or a specific server. The server has

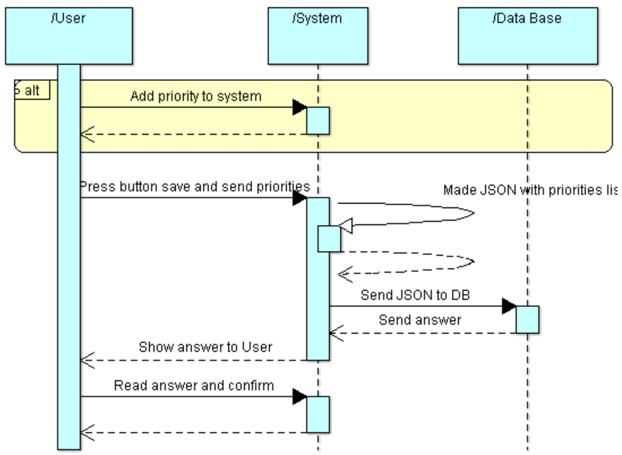

FIGURE 4. Sequence diagram for creating and saving a priority list.

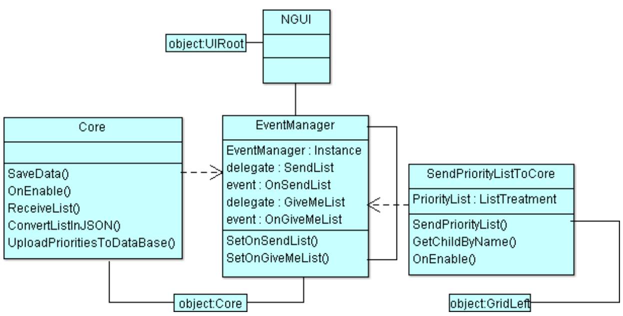

FIGURE 5. Class diagram excerpt of PriorityNet app.

several scripts in PHP programming language. One of these receives information from a JSON file.

PriorityNet app sends information to DBMS through a web request, specifically a POST request. The request includes the JSON file content, and the URI/URL of the receptor script for managing the information. When a PHP file receives a request, it might update, add or delete information depending on the content of the JSON file. In case of managing a change of priority list in the network, the PHP file updates the database with a new priority list for a specific user. Finally, the PHP file returns another JSON file to the mobile device. This JSON file contains information about the result of the query in the DBMS, which can be either a confirmation of success or an error. In this way, users can read this message, and could repeat their actions later if necessary.

The class diagram of Figure 5 specifies the structure of two classes. On the one hand, the "Core" class receives input from a priority list, turns it into a JSON file, and uploads the converted file to the DBMS. On the other hand, the "EventManager" class receives input from a list of events and delegates, and was implemented with a singleton pattern. A delegate is a type that represents references to methods with a particular parameter list and a return type. Events are triggered for notifying some observers when something of interest occurs. The methods of some classes are associated with these events.

The "SendPriorityListToCore" class has a method that collects all the items from the priority list, and sends the names of these items as strings to the Core class. After executing this method, the system collects this priority list, and adds certain information for conforming a JSON file. This file includes information such as the user ID. This ID allows the system to track users and to manage priorities fairly.

PriorityNet App was developed with Unity 3D engine and the library Next Graphical User Interface (NGUI) [22]. Unity 3D has been widely used in both industry and research communities, like in the simulation of distributed sensor networks [23]. One of its main advantage is its multi-platform nature, allowing the deployment of apps in the main mobile operative systems such as Android and iOS. We used NGUI for performing tasks associated with the user interface. The NGUI library was created by Tasharen Entertainment and released in 2011. The main aim of NGUI was to ease the creation of user interfaces. In the presented app, NGUI was useful for adding the drag and drop functionality. NGUI also supported the implementation of lists with draggable elements. These functionalities were relevant for achieving an easy-to-use and intuitive app.

C. PRIORITY MECHANISM

The priority mechanism of the current approach is aimed at satisfying the following rule in which the differences of waiting times should be as large as possible:

$$p(r_x) > p(t_y) \Rightarrow w(t_x) \geq w(t_y) \tag{1}$$

where r_x and r_y are requests of services, $p(r)$ determines the priority of r request, and $w(r)$ determines its waiting time.

Figure 6 shows the block diagram of the mechanism for managing and attending requests. On the one hand, the system manages the reception of requests of smart city services, as shown in the left side of the diagram. The system receives a request with a certain priority. Then, the system normalizes this priority into the range of priorities of the system, e.g. [0, 2] being zero the highest priority and two the lowest one. In this step, the system can optionally use a mechanism of normalizing the priorities to avoid selfish citizens. The system has several queues of requests, and each queue is associated with the requests of each priority level. In order to attend the high-priority requests first, the requests are not only added to the queue of its level but also to the ones with lower priority levels. In this manner, it is guaranteed that each request is attended before any other later request of its level or lower ones.

On the other hand, this priority mechanism attends the service requests in each iteration, as one can observe in the right side of Figure 6. It starts with the highest-priority queue. In order to regulate that normally high priorities are attended much faster, the number of attended requests per iteration is different for each queue regarding its priority level. For example, in each iteration, the system can attend three requests of the highest priority level, two requests of the medium priority level and one priority of the lowest iteration level. Once a

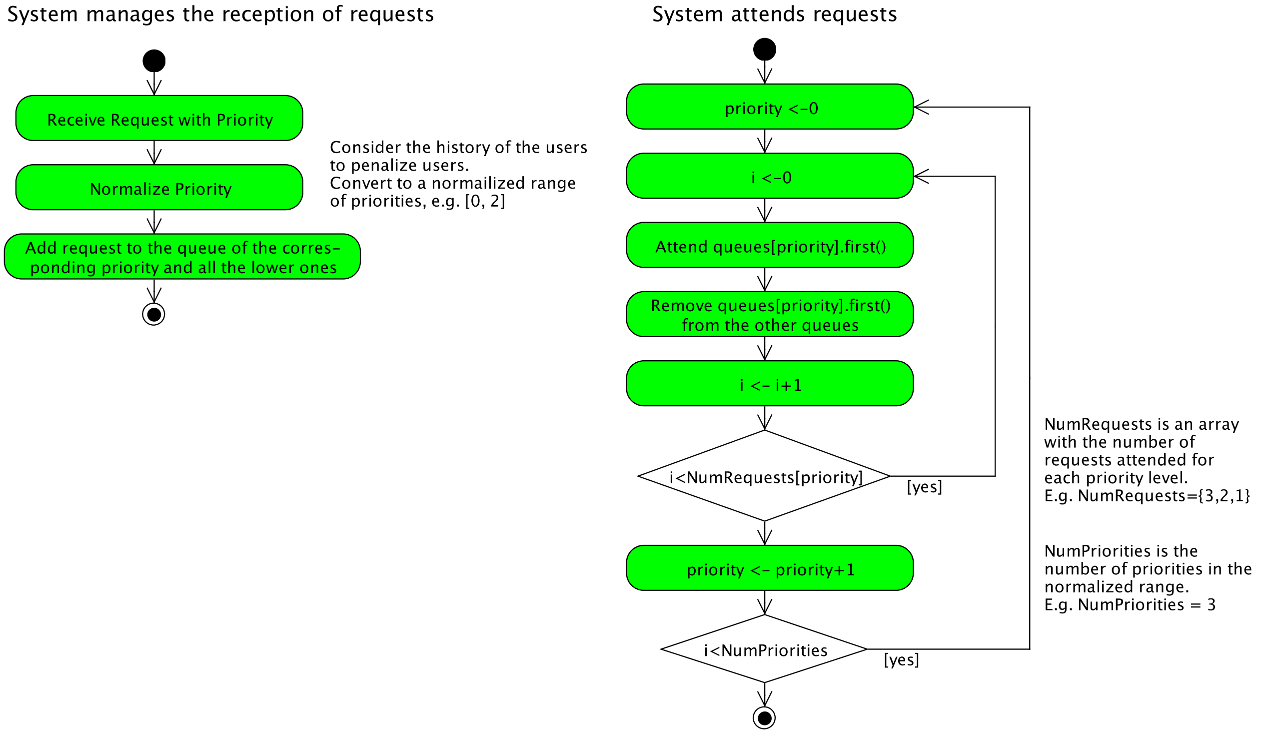

FIGURE 6. Block diagram of the priority mechanism.

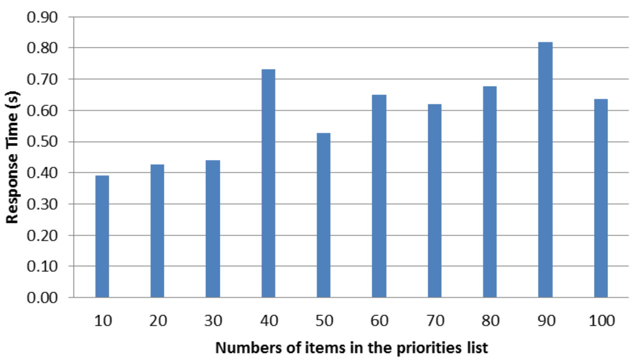

FIGURE 7. Response time of PriorityNet App.

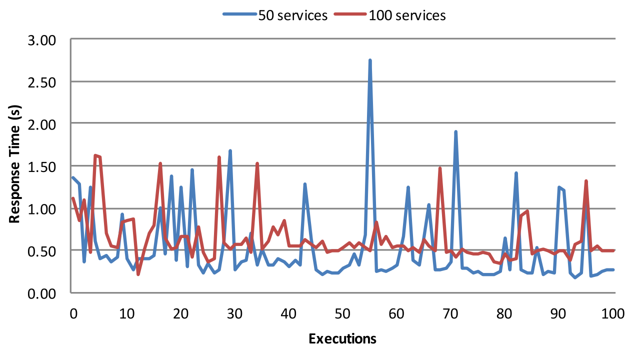

FIGURE 8. Response times for uploading priority lists of respectively 50 and 100 services.

request is attended, it is removed from the corresponding queue and all the others.

IV. EXPERIMENTATION

A. EVALUATION OF RESPONSE TIME

We measured the response time of PriorityNet when uploading a priority list to the system. This task includes (a) converting a priority list into a JSON file, (b) sending a POST request with a JSON file, (c) receiving an answer from the server, and (d) processing the server's answer. We measured all the time elapsed between touching the Save button and the presentation of the confirmation message to the user.

These experiments used lists of sizes from 10 to 100 with intervals of ten. We performed 100 executions for each list size, and Figure 7 presents the average results.

As one can appreciate, the average time never exceeded one second in any case. We consider these results as satisfactory, because the user did not wait an excessive time even when using priority lists of 100 services.

The experimentation with PriorityNet showed its scalability. When increasing the size of the priority list, the response time only slightly increased. There were some exceptions in the results collected from priority lists of sizes 40 and 60.

Moreover, Figure 8 shows all the response times for uploading priority lists of respectively 50 and 100 services. We executed 100 times the upload process for each of these two list sizes. Both priority list sizes had some peaks that are high in comparison to their common values. These peeks may be due external factors regarding the network, such as congestions or general overloads.

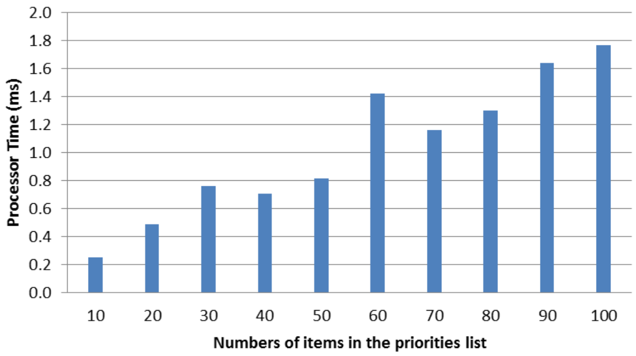

FIGURE 9. Processing time of PriorityNet app.

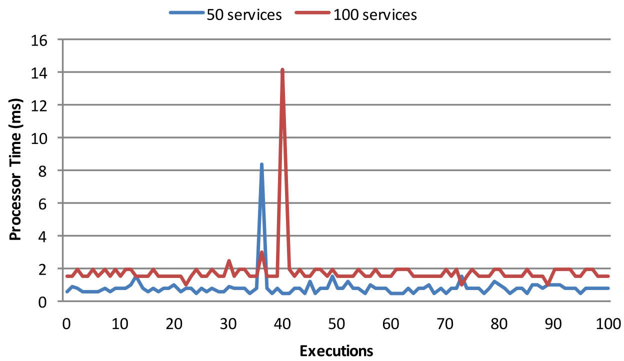

FIGURE 10. Processing times for lists of respectively 50 and 100 services.

B. EVALUATION OF PROCESSING TIME

We measured the processing time of PriorityNet app when defining and saving priority lists. More concretely, we measured the elapsed time from pressing the Save button until the data were prepared to be transferred. This includes (a) triggering an event to request the priority list, (b) receiving a priority list by means of a delegate, and (c) converting the visual list into a serializable format.

This analysis executed 100 times each of the same ten priority list sizes as before. Figure 9 shows the average results of the processing time for these list sizes. It is worth noting that the absolute values of processing time were low. For example, the average processing time for priority lists of 100 services was only 1.76 ms.

Figure 10 shows the processing times of 100 executions for priority list sizes of respectively 50 and 100 services. Most values of each priority list size were near its corresponding average, which was 0.81 ms for lists of 50 services and 1.76 ms for lists of 100 services. There was a high peek value for each priority list size, but we consider that these values are outlier, since they only occurred once in each analyzed set of 100 simulations. The standard deviation values of 0.78 ms (for the list of 50 services) and 1.28 ms (for the list of 100 services) revealed that the variation of processing times was low considering their absolute values.

C. USER STUDY

We conducted a user study to measure some features of the users' experience when using PriorityNet app.

TABLE 1. Priority lists for the tests of the user study.

Test 1	Test 2	Test 3
Music	Send Message	Maps
Maps	Calendar	Clock
Whatsapp	Music	Radio
Gallery	E-mail	Send Message
Radio		GPS
Send Message		

1) SAMPLE OF PARTICIPANTS

We recruited 21 people for participating in this user study. They were 21.1 years old in average (SD=4.17). Seven participants were male (33.3%) and the others were female (66.6%). Only one participant worked or studied in computer science field (0.047%).

2) PROCEDURE

The app was briefly introduced to each participant. Then, the experimenter briefly showed how to use the app to each participant. More concretely, he explained how to set up priorities in the system. Participants saw how to drag and drop items from the right list to left one, and they only saw this operation. The experimenter did not explain how to alter the elements of the priority list if they had made a mistake, either including the wrong element or assigning it to the wrong priority. We did not explain this on purpose, because if the app was sufficiently easy to use, users should be able to deduce it by themselves.

The experimenter asked each participant to sequentially define and save the priority lists presented in table 1. The first priority list was longer than the others to let them get used to the app. The second list had items that the users were not able to see without scrolling down. In this way, participants needed to scroll down for completing the task. The last test was similar to the second one. The main difference was the way to do it. When a participant had finished a test, this participant was told to change the order of the items of the list.

Lastly, the participants were asked to answer a questionnaire about this application. This questionnaire was composed from the validated scales mentioned in next section.

3) MEASURES

The questionnaire about the app was composed of several validated scales. The first scale was the System Usability Scale (SUS) [24]. SUS is composed of 10 items with alternatively direct and inverted items. SUS uses a 5-point Likert scale

(1 = "strongly disagree"; 5 = "strongly agree"). This scale has been validated, and is widely used. The items were scored in two different ways regarding whether these questions were direct or inverted:

- Direct items: $score = response - 1$
- Inverted items: $score = 5 - response$

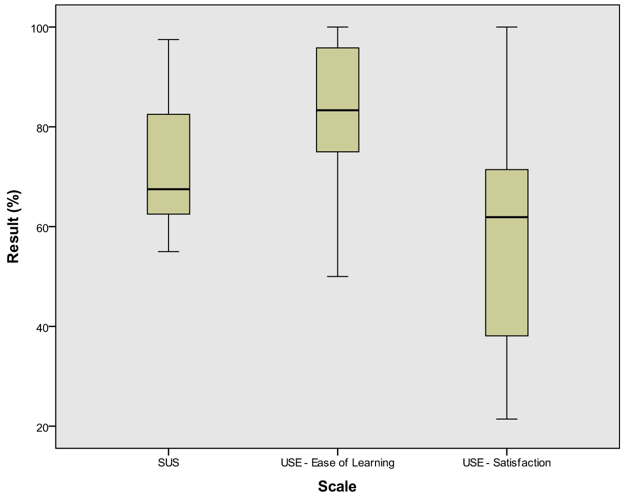

FIGURE 11. Boxplot of the questionnaire results from the user study.

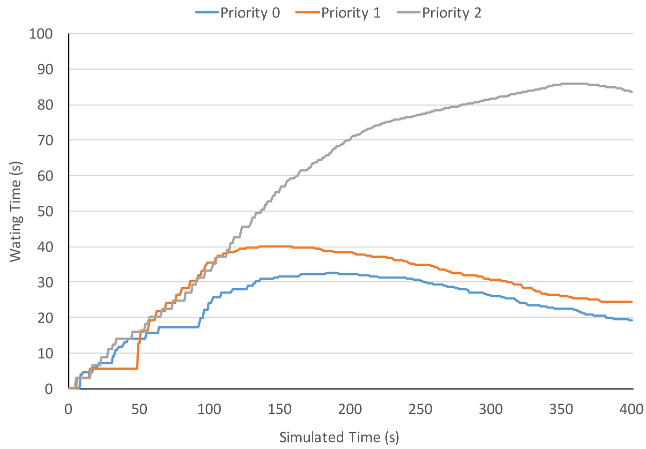

FIGURE 12. Waiting times for 35 services/second without penalizing selfish agents.

The sum of the scores of all the items was in the 0 - 40 range for each replied questionnaire. In order to obtain a score in the 0 - 100, the original score was multiplied by 2.5.

The other scale was Usefulness, Satisfaction, and Ease of use questionnaire (USE) [25]. This scale has four independently validated dimensions, and we only used the dimensions of (a) ease of learning and (b) satisfaction. All the items were directly related with the corresponding feature, without using inverted questions. The questions were answered with a 7-point Likert scale. Each item was assessed with a value in the 0 - 6 range. Each dimension was converted to a 0-100 range by multiplying the sum of all the responses by $100/maxScore$ where $maxScore$ was the maximum summed score in the corresponding dimension.

4) RESULTS

All the participants successfully completed all the tasks. It is worth mentioning that in the rearrangement task of the the third test, the participants faced to something new, because they had never learned how to reorder items in the priority list. The participants used two ways to address this task. The first way was to drag and drop elements from one position to another of the same list. The other way was to remove elements by dragging them outside the priority list and then to add them again in the priority list in the corresponding order. Regardless of the methods used by participants, all of them were able to successfully finish this task of the third test.

Figure 11 presents the results of the validated scales with a boxplot. The highest-ranked feature of the app was the ease of learning dimension from the USE scale, with a mean value of 82.94%. Thus, PriorityNet app was easy to learn according to the validated scale. This is also confirmed by the fact that all users figured out how to alter an existing priority list.

The second ranked feature was the usability measured with the SUS scale with an average value 72.26%. This reflects the high usability of the app, showing that the user interface of the app was probably properly designed. The last feature obtained a 57.48% average value in the satisfaction dimension from the USE scale. This dimension was probably the least ranked since participants did not observe the whole utility of the app in the short tasks of the user study. For example, they did not experience the fast communications and services for real important matters in their lives after configuring their priority preferences.

We conducted paired t-tests [26] for assessing the significances of the differences in all the possible pairs among the analyzed usability dimensions 2. One can observe that ease of learning had significant differences from the other two dimensions with a significance level of 0.001. The difference between usability and satisfaction was also significant, but with a higher significance level of 0.002.

D. COMPARISON OF WAITING TIMES FOR SERVICES CONSIDERING PRIORITIES

In order to further assess the current approach, we performed simulations with the ABS-SmartPriority application. We executed 100 agents simulating users that requested services with different priorities. These agents took nondeterministic decisions based on their goals following TABSAOND (a technique for developing agent-based simulation apps and online tools with nondeterministic decisions) [27]. The simulator simulated a duration of 400 s in each execution. Figure 12 shows an example in a smart city that was able to deliver 35 services per second for these 100 agents. This simulation did not penalize selfish users that overused high priorities. One can observe that the system was able to properly attend high-priority requests faster.

In addition, we executed a simulation in which the smart city was only able to attend 15 services per second, without using the penalization mechanism. Figure 13 shows the results. In this case, the overload of requests made the system not to be able to attend urgent requests faster.

Furthermore, the current approach was simulated with a priority mechanism in which some users could be penalized by lowering the priorities of the requests in case they had

TABLE 2. Paired t-test results for comparing the results about the different features of the app.

		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference		t	df	Sig. (2-tailed)
					Lower	Upper			
Pair 1	SUS - USE Ease of Learning	-10.67429	12.97284	2.83091	-16.57945	-4.76912	-3.771	20	.001
Pair 2	SUS - USE Satisfaction	14.77810	18.51249	4.03976	6.35131	23.20488	3.658	20	.002
Pair 3	USE Ease of Learning - USE Satisfaction	25.45238	25.2834	5.50964	13.95948	36.94529	4.620	20	.000

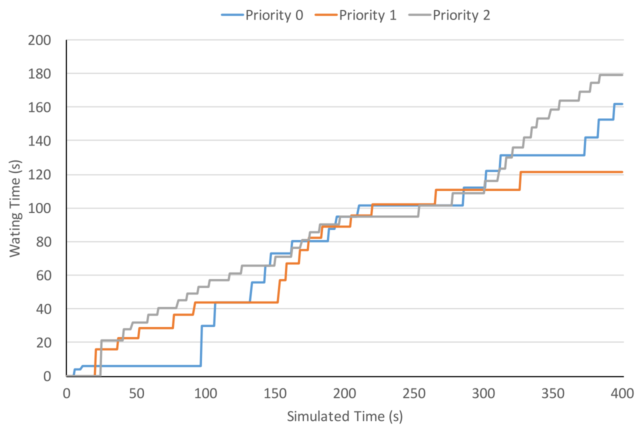

FIGURE 13. Waiting times for 15 services/second without penalizing selfish agents.

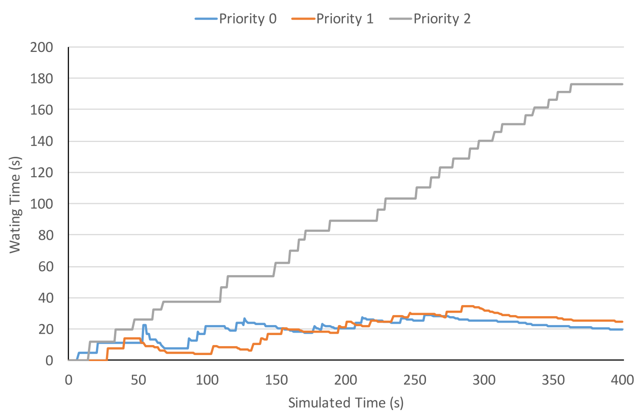

FIGURE 14. Waiting times for 15 services/second penalizing selfish agents.

overused high-priority requests. The rate of attended services was the same as in the previous case. Figure 14 presents the results. This penalization mechanism allowed the system to attend faster the high-priority requests.

V. CONCLUSIONS AND FUTURE WORK

This work has presented a novel app that allows users to dynamically establish priorities. This is useful for prioritizing different services and fastening urgent services in smart cities with ultra-dense networks. This opens a new channel of communication between smart cities and citizens. The app has showed a high level of usability, especially in ease of learning. This app has also shown its proper scalability when increasing the list priority size in terms of response time for uploading preferences. Some agent-based simulations showed that the proposed approach allows attending urgent services faster.

The current work is planned to be extended by allowing users to activate an automatic mode. In this mode, the app would automatically select the prioritized list of services based on the particular history of the corresponding user. For instance, if the user normally establishes a service as urgent in the weekdays and removes it from the list in weekends, the app would follow a similar pattern in automatic mode.

REFERENCES

[1] C. Galiotto, N. K. Pratas, L. Doyle, and N. Marchetti, "Effect of LOS/NLOS propagation on 5G ultra-dense networks," *Comput. Netw.*, vol. 120, pp. 126–140, Jun. 2017.

[2] P. Lynggaard and K. E. Skouby, "Deploying 5G-technologies in smart city and smart home wireless sensor networks with interferences," *Wireless Pers. Commun.*, vol. 81, no. 4, pp. 1399–1413, Apr. 2015. [Online]. Available: https://doi.org/10.1007/s11277-015-2480-5

[3] Z. Zhang, C.-H. Lung, I. Lambadaris, and M. St-Hilaire, "When 5G meets ICN: An ICN-based caching approach for mobile video in 5G networks," *Comput. Commun.*, vol. 118, pp. 81–92, Mar. 2017.

[4] M. Taha, L. Garcia, J. M. Jimenez, and J. Lloret, "SDN-based throughput allocation in wireless networks for heterogeneous adaptive video streaming applications," in *Proc. 13th Int. IEEE Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Jun. 2017, pp. 963–968.

[5] M. Taha, L. Parra, L. Garcia, and J. Lloret, "An Intelligent handover process algorithm in 5G networks: The use case of mobile cameras for environmental surveillance," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, May 2017, pp. 840–844.

[6] J. Lloret, L. Parra, M. Taha, and J. Tomás, "An architecture and protocol for smart continuous eHealth monitoring using 5G," *Comput. Netw.*, vol. 129, pp. 340–351, Dec. 2017.

[7] B. Blanco *et al.*, "Technology pillars in the architecture of future 5G mobile networks: NFV, MEC and SDN," *Comput. Standards Interfaces*, vol. 54, pp. 216–228, Nov. 2017.

[8] T. Anagnostopoulos, K. Kolomvatsos, C. Anagnostopoulos, A. Zaslavsky, and S. Hadjiefthymiades, "Assessing dynamic models for high priority waste collection in smart cities," *J. Syst. Softw.*, vol. 110, pp. 178–192, Dec. 2015.

[9] I. García-Magariño and R. Lacuesta, "ABS-SmartPriority: An agent-based simulator of strategies for managing self-reported priorities in smart cities," *Wireless Commun. Mobile Comput.*, vol. 2017, 2017, Art. no. 7254181. [Online]. Available: https://doi.org/10.1155/2017/7254181

[10] K. E. Skouby and P. Lynggaard, "Smart home and smart city solutions enabled by 5G, IoT, AAI and CoT services," in *Proc. Int. Conf. IEEE Contemp. Comput. Informat. (IC3I)*, Nov. 2014, pp. 874–878.

[11] I. Chatzigiannakis, "Apps for smart buildings: A case study on building security," in *Start-Up Creation: The Smart Eco-Efficient Built Environment*. Sawston, U.K.: Elsevier, 2016, pp. 465–479.
[12] P. Gora and I. Rüb, "Traffic models for self-driving connected cars," *Transp. Res. Procedia*, vol. 14, pp. 2207–2216, Apr. 2016.
[13] A. S. Rao and M. P. Georgeff, "Modeling rational agents within a BDI-architecture," in *Proc. KR*, 1991, pp. 473–484.
[14] I. Ucar, C. Donato, P. Serrano, A. Garcia-Saavedra, A. Azcorra, and A. Banchs, "On the energy efficiency of rate and transmission power control in 802.11," *Comput. Commun.*, vol. 117, pp. 164–174, Feb. 2018. [Online]. Available: https://doi.org/10.1016/j.comcom.2017.07.002
[15] K. Zheng, Z. Yang, K. Zhang, P. Chatzimisios, K. Yang, and W. Xiang, "Big data-driven optimization for mobile networks toward 5G," *IEEE Netw.*, vol. 30, no. 1, pp. 44–51, Jan./Feb. 2016.
[16] B.-J. Chang and S.-H. Liou, "Adaptive cooperative communication for maximizing reliability and reward in ultra-dense small cells LTE-A toward 5G cellular networking," *Comput. Netw.*, vol. 115, pp. 16–28, Mar. 2017.
[17] J. Seyedmohammadi, F. Sarmadian, A. A. Jafarzadeh, M. A. Ghorbani, and F. Shahbazi, "Application of SAW, TOPSIS and fuzzy TOPSIS models in cultivation priority planning for maize, rapeseed and soybean crops," *Geoderma*, vol. 310, pp. 178–190, Jan. 2018.
[18] R. Sudarsan and C. J. Ribbens, "Combining performance and priority for scheduling resizable parallel applications," *J. Parallel Distrib. Comput.*, vol. 87, pp. 55–66, Jan. 2016.
[19] S. Binder, Y. Maknoon, and M. Bierlaire, "Exogenous priority rules for the capacitated passenger assignment problem," *Transp. Res. B, Methodol.*, vol. 105, pp. 19–42, Nov. 2017.
[20] M. A. Jan, P. Nanda, X. He, and R. P. Liu, "PASCCC: Priority-based application-specific congestion control clustering protocol," *Comput. Netw.*, vol. 74, pp. 92–102, Dec. 2014.
[21] E.-J. Kim, M. Kim, S.-K. Youm, S. Choi, and C.-H. Kang, "Priority-based service differentiation scheme for IEEE 802.15.4 sensor networks," *AEU-Int. J. Electron. Commun.*, vol. 61, no. 2, pp. 69–81, 2007.
[22] C. Bernardoff, *NGUI for Unity*. Birmingham, U.K.: Packt, 2014.
[23] I. García-Magariño, R. Lacuesta, and J. Lloret, "ABS-FishCount: An agent-based simulator of underwater sensors for measuring the amount of fish," *Sensors*, vol. 17, no. 11, p. 2606, 2017.
[24] J. Brooke *et al.*, "SUS—A quick and dirty usability scale," *Usability Eval. Ind.*, vol. 189, no. 194, pp. 4–7, 1996.
[25] A. M. Lund, "Measuring usability with the use questionnaire[12]," *Usability Interface*, vol. 8, no. 2, pp. 3–6, 2001.
[26] H. Hsu and P. A. Lachenbruch, "Paired *t* test," in *Wiley Encyclopedia of Clinical Trials*. Hoboken, NJ, USA: Wiley, 2008.
[27] I. García-Magariño, G. Palacios-Navarro, and R. Lacuesta, "TABSAOND: A technique for developing agent-based simulation apps and online tools with nondeterministic decisions," *Simul. Model. Pract. Theory*, vol. 77, pp. 84–107, Sep. 2017.

FRANKS GONZÁLEZ-LANDERO received the degree in computer science engineering from the University of Zaragoza in 2012, where he is currently pursuing the Ph.D. degree. He did his master's thesis in computer graphics, videogames, and virtual reality at the University Rey Juan Carlos of Madrid in 2016. He adapted his engineering grade to the European University Education System in 2017. He has been a Computer Science Engineer with Edison Desarrollos since 2015. He has specialized in mobile application development with several courses, with topics, such as Android and Corona SDK. He has professional experience in mobile application development with a cross-platform engine and devices, such as Microsoft Kinect. His research interests include mobile application development, usability, and agent-based simulators.

IVÁN GARCÍA-MAGARIÑO received the Ph.D. degree in computer science engineering from the Complutense University of Madrid in 2009. He was a Lecturer with Madrid Open University from 2010 to 2014. He has been a Lecturer with the University of Zaragoza since 2014. He belongs to the EduQTech research group. Among journals, book chapters, conferences, and workshops, he has over 100 publications (36 in journals with ISI Thomson JCR). His most relevant publications belong to international journals with a high impact, such as *Engineering Applications of Artificial Intelligence*, *Expert Systems With Applications*, *Information Sciences*, *Knowledge-Based Systems*, *Information and Software Technology*, *Simulation Modelling Practice and Theory*, *Journal of Systems and Software*, *Personal and Ubiquitous Computing*, *International Journal of Medical Informatics*, *Medical and Biological Engineering and Computing*, *Journal of Biomedical Informatics*, and *Computer Standards and Interfaces*. He was a recipient of the FPI Researcher Scholarship from 2006 to 2010.

RAQUEL LACUESTA (M'17) received the degree in computer science engineering and the Ph.D. degree (Dr.Ing.) in computer science engineering from the Polytechnic University of Valencia, in 1999 and 2008, respectively. She has been a Lecturer of Computer Science with the University of Zaragoza, for over 12 years, where she currently teaches human–computer interaction, security, and databases subjects. She has authored or co-authored over 30 scientific papers published in national and international conferences, over 15 papers about education published in national and international conferences, and several papers published in international journals. Her main topics of research are security and auto-configuration on ad hoc and spontaneous networks, the design and evaluation of routing algorithms, computer–human interaction, and education. She has been involved as an Organizer and Chair for several important program committees of international conferences. She is an Associate Editor and Reviewer of the *International Journal Networks Protocols and Algorithms* and a member of different national research projects.

JAIME LLORET (M'07–SM'10) received the M.Sc. degree in physics in 1997, the M.Sc. degree in electronic engineering in 2003, and the Ph.D. degree in telecommunication engineering (Dr.Ing.) in 2006. He is currently an Associate Professor with the Universitat Politecnica de Valencia. He is the Chair of the Integrated Management Coastal Research Institute and the Head of the Active and Collaborative Techniques and Use of Technologic Resources in the Education Innovation Group. He is the Director of the University Diploma Redes y Comunicaciones de Ordenadores and the University Master Digital Post Production. He leads many national and international projects. He has authored 22 book chapters and over 360 research papers published in national and international conferences and international journals (over 140 with ISI Thomson JCR). He is an IARIA Fellow. He has been the Internet Technical Committee Chair (the IEEE Communications Society and the Internet Society) from 2013 to 2015. He is the IARIA Journals Board Chair for eight journals. He is currently the Chair of the Working Group of the Standard IEEE 1907.1. He has been a general chair (or co-chair) of 39 international workshops and conferences. He has been involved in over 320 program committees of international conferences, and over 130 organization and steering committees. He has been a co-editor of 40 conference proceedings and a guest editor of several international books and journals. He is the Editor-in-Chief of *Ad Hoc and Sensor Wireless Networks* (with ISI Thomson Impact Factor), the *International Journal of Networks Protocols and Algorithms*, and the *International Journal of Multimedia Communications*. He has been an associate editor of 46 international journals (16 with ISI Thomson Impact Factor).