

Research Article

Offloading Positioning onto Network Edge

José Santa , **Pedro J. Fernández** , **Ramon Sanchez-Iborra,**
Jordi Ortiz , and **Antonio F. Skarmeta** 

Department of Information and Communications Engineering, University of Murcia, Murcia 30100, Spain

Correspondence should be addressed to José Santa; josesanta@um.es

Received 10 May 2018; Revised 19 September 2018; Accepted 4 October 2018; Published 23 October 2018

Guest Editor: Marco D. Santambrogio

Copyright © 2018 José Santa et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

While satellite or cellular positioning implies dedicated hardware or network infrastructure functions, indoor navigation or novel IoT positioning techniques include flexible storage and computation requirements that can be fulfilled by both end-devices or cloud back-ends. Hybrid positioning systems support the integration of several algorithms and technologies; however, the common trend of delegating position calculation and storage of local geoinformation to mobile devices or centralized servers causes performance degradation in terms of delay, battery usage, and waste of network resources. The strategy followed in this work is offloading this computation effort onto the network edge, following a Mobile Edge Computing (MEC) approach. MEC nodes in the access network of the mobile device are in charge of receiving navigation data coming from both the smart infrastructure and mobile devices, in order to compute the final position following a hybrid approach. With the aim of supporting mobility and the access to multiple networks, an Information Centric Networking (ICN) solution is used to access generic position information resources. The presented system currently supports WiFi, Bluetooth LE, GPS, cellular and NFC technologies, involving both indoor and outdoor positioning, using fingerprinting and proximity for indoor navigation, and the integration of smart infrastructure data sources such as the door opening system within real smart campus deployment. Evaluations carried out reveal latency improvements of 50%, as compared with a regular configuration where position fixes are computed by mobile devices; at the same time the MEC solution offers extra flexibility features to manage positioning databases and algorithms and move extensive computation from constrained devices to the edge.

1. Introduction

Positioning is nowadays a need for the vast number of location-aware applications that run in mobile devices. Current operating systems for these devices, like Android or iOS, tend to abstract apps from the positioning technology used, but several underlying information sources can be used, such as GPS, cell identification, gyroscope, or WiFi access point (AP) location. When available, GPS is usually taken as the primary technology, since WiFi and cellular alternatives provide estimations of the device position within a geographical area in the range from meters to kilometers. Sensors like gyros and barometers can further improve this location with orientation and altitude.

There are many solutions in the literature about indoor navigation to solve the problem of GPS obstruction [1, 2]. Even cooperative positioning strategies can be used to improve GPS location by exchanging location data among

terminals [3]. However, there are alternative technologies to GPS that have been analyzed for indoor scenarios in the areas of sound, vision, magnetic fields [4], and radio-frequency. The last ones have been especially studied because of flexibility advantages, since currently available network deployment (such as WiFi) can be used for positioning. When using radio-frequency technologies, among the various algorithms proposed to compute a position, trilateration, triangulation, and fingerprinting are the most widespread. The last one has been especially explored [5], given that it can compute quite accurate positions if a proper training has been carried out beforehand. Hybrid solutions further generalize the problem, by combining several technologies, and even algorithms, to provide a better position estimation.

The issue found when using a positioning algorithm, or a combination of them in hybrid approaches, is the performance implications regarding calculation and storage of training data sets. Fingerprinting solutions, for instance,

require the execution of a pattern matching algorithm over a potentially large dataset of previously collected radio signal measurements. Observe that, according to our experience, a basic training dataset of a 20-storey office building with about ten APs per floor, with no other additional technology used, can imply more than 20 MBs. This problem gets worse when more spaces are modeled, since the pattern matching algorithm must perform an exhaustive search among all of these values. Hence, the computing performance issue is clear if mobile devices are in charge of that task, but even cloud solutions can be a bottleneck with a growing number of users, positioning spaces, different algorithms, and more than one technology.

Mobile Edge Computing (MEC) is found as a promising technology to alleviate this computing performance issue in positioning. MEC involves intermediary nodes near end (mobile) nodes that are attached to the access segment of the network, which can perform processing and storage tasks to offload end-nodes or saturated cloud platforms [6]. For that reason, different services have been identified for benefiting from the possibilities empowered by the MEC paradigm: computation/storage offloading, optimized content delivery and caching, or radio access network management, among many others [7]. This implies a location awareness feature that perfectly fits with the requirements of the positioning problem. Considering that each positioning technology, such as WiFi, can provide a hierarchical access network, a set of MEC nodes can execute positioning algorithms and store training datasets. The performance speedup in this case is twofold, since a small network delay is expected from these MEC nodes when accessing a positioning service, as compared with a remote cloud platform, but also a greater computing performance can be offered here, as compared with mobile devices.

Following the previous approach, we go a step further as compared with current research in the area and present a hybrid positioning platform where calculation and data storage are offloaded onto a distributed set of MEC nodes. Each of them offers a positioning service for mobile devices connected to their attached access networks. A hybrid positioning is offered and, as compared with previous works, apart from combining GPS, cellular, WiFi, and Bluetooth LE, information coming from sensory data gathered from smart spaces is integrated in the computed fixes. MEC nodes maintain a dataset of WiFi APs, Bluetooth LE tags, and fingerprinting training only concerning the space covered by them, and the position determined is based on an accuracy factor calculated for all technologies at a particular moment for the mobile device. A great advance is also the mobility support and access to the positioning service, which is powered by the use of an Information Centric Networking (ICN) solution, which is in charge of routing mobile node requests to the proper MEC node in a transparent way. Hence, the next **contributions** are provided in the paper:

- (i) a MEC offloading scheme for position calculation,
- (ii) an embedded hybrid positioning approach integrating wireless technologies and smart infrastructure,

- (iii) an ICN solution to provide a generic access to positioning service,
- (iv) evaluation of the system in a real smart campus deployment.

The paper starts with a review of related solutions in the field in Section 2, and then the proposal is presented in its general form in Section 3. This is followed by the hybrid algorithm used to compute positioning fixes in Section 4 and a description of the resolution of positioning requests in Section 5. Then the real operation of the system is assessed in a real deployment at the University of Murcia in Section 6. Finally, conclusions and next steps are included in Section 7.

2. Related Work

2.1. MEC for Positioning. MEC is a recently created computing paradigm that can speed up and make more scalable applications involving huge volumes of data and, above all, those requiring considerable processing and low response times, such as traffic management [8], monitoring of complex events and processes [9], data-caching accessing Internet resources [10], or contextual adaptation of contents [11]. As can be seen, most of these identified services take advantage of the offloading process of computation, storage, or traffic handling from the end-devices onto the MEC nodes. This operation has been the first target of study regarding the novel MEC-enabled services identified by the research community [12]. For example, state-of-the-art networking techniques such as Software Defined Networks (SDN) [13], IoT-related communication technologies, or 5G are being introduced or extended with the support of MEC nodes by offloading certain networking tasks onto them [14]. In turn, in mobile scenarios, offloading computational-complex tasks from the notably constrained end-devices onto nodes placed at the edge of the radio access network have also been deeply explored [15, 16]. However, an application that has not received particular attention in MEC yet is positioning, which has the distinctive feature of being a service that enables a lot of applications to work.

An interesting advantage of MEC is its location-oriented approach, since MEC nodes are intended to be installed at the access network edge, being possible to provide a context-oriented treatment of data [17]. This is of incredible help for positioning, since location algorithms, especially the indoor ones, require high computing demands that do not easily scale, as discussed above. Novel architectures of MEC emphasize the advantages of its location-aware design [18], but this has not been exploited in positioning, if scientific literature is reviewed.

An initial effort in this line has been the definition of a MEC Location Application Programming Interface (API) [19], which includes the primitives for implementing MEC nodes able to resolve location queries following a Web service model. The work in [20] offers implementation of this API by using the positioning information coming from a 3GPP network, by using the Parlay X Terminal Location Service. The proposal presented in the current paper follows a similar line, by offering a Web service to resolve positioning requests

of both users and external applications, but it is offered in a novel way by using an ICN that supports terminal mobility and network topology independence when asking to solve positioning or location information. In addition, a more advanced positioning technique is used, integrating a hybrid approach where different technologies and algorithms can coexist.

2.2. SDN and ICN. The Software Defined Networking [21] is a network paradigm in which there is a intended separation between the control and data planes. The control plane, which has been historically distributed over the network elements with a limited knowledge of the operation environment, is now centralized into an entity named SDN controller. This node has a complete vision of the network and is able to take optimal network decisions (e.g., routing). SDN is considered a key enabler for new advances such as Network Function Virtualization (NFV) or 5G. Among the developments following the SDN paradigm, OpenFlow [22, 23] can be considered as the reference technology, partly due to its early adoption by market products that attracted the interest of the industry.

SDN can be a useful enabler for the development of Content Delivery Networks (CDN). In CDN, data contents are transparently retrieved from the most convenient network node (usually the nearest) by means of traffic steering techniques. As evolution of CDN, Information Centric Networks [24] do not address content by a location-related identifier, such as the IP address of a web server, but by a content identifier. The ICN paradigm usually implies in-network caching, where the network nodes acquire the capability to route the requests towards the source and store a copy of the delivered data.

In this work an ICN solution is proposed to abstract end-devices from the network resource needed to solve location requests. This scheme inherently provides mobility support, since the system is agnostic from the access network used, and allows the core or edge network to balance the computing load.

2.3. Hybrid Positioning. Attending exclusively to location technologies and algorithms, hybrid location is gaining momentum in the area of positioning, being particularly exploited when indoor spaces are considered. In [25], for instance, the authors presented a solution with a framework approach equivalent to the one presented in the current paper. The system supports the inclusion of new technologies in a modular way, although the implementation is limited and no operation tests are included. The age of the proposal (2012) is remarkable; thus, novel technologies such as Bluetooth LE or NFC are not considered, and the handover between technologies is not discussed. Similar lack is found in the more recent work in [26], where the Android positioning service, mainly based on WiFi proximity and cell ID, is combined with NFC for indoor support. An additional problem here is the need for a significant deployment of NFC tags to compute indoor fixes. The handover issue is addressed in [27], evaluating an approach also followed in this work, which is based on border areas that prepare the

system to decide the new technology to choose. This softens the handover, since technologies such as GPS need some time to start up. Handover capabilities are also included in [28], but attending more to the estimation of the accuracy. For GPS the dilution of precision (DOP) value is used, as in the current work, whereas for the case of WiFi, only the number of available APs is used as quality indicator. In [29] an even more simple approach is used, since the trigger of the handover is the number of GPS satellites visible, which avoids the usage of more than one indoor positioning solution. In our work, when fingerprinting is used for indoors, the accuracy is calculated on the basis of the spatial distribution of the nearest fingerprints after the pattern matching algorithm is executed. According to the work in [30], this is the algorithm that presents a better trade-off between accuracy and computational cost.

There are sophisticated dead reckoning (DR) algorithms in the literature that deals with indoor navigation using multiple information sources. In [31], a pedestrian DR (PDR) algorithm is improved with a landmark detection system to correct the cumulative error. Here, inertial sensors are used by the DR algorithm, but no communication technologies nor other algorithms are hybridized with this approach, limiting the application domain of the solution. The work in [32] suffers from the same constraint, but it proposes the usage of Bluetooth LE and maps to correct DR drift. These PDR proposals offer good solutions without the need of training the algorithm, like in fingerprinting-based systems; however, they are not properly prepared to perform handovers to other positioning technologies. A handover mechanism is used in [33], and the parameters considered to choose among several algorithms are based on an estimation of the accuracy, which is also considered in the present work. However, positioning technologies and algorithms are too coupled in the solution and the DR algorithm used is too dependent on an initial good location estimation. In [34], a PDR system with handover support between indoor and outdoor operational modes is presented. The indoor or outdoor location is detected using the light and magnetic reads obtained from the mobile device sensors. Then, according to the operational settings, the PDR algorithm is used for indoors or regular GPS is activated for outdoors.

The issue with PDR approaches lies in the complexity of calculations and the need for extra positioning technologies to provide an initial location and compensate for cumulative errors. A distinctive factor of the positioning subsystem presented here is the development of an open framework where multiple technologies and algorithms can coexist and, more importantly, where the computations can be performed in a distributed way, alleviating the computing load of mobile devices and/or clouds. Moreover, the generic approach allows the usage of infrastructure-based positioning data coming from smart deployment, such as the card-based authentication used in our smart campus testbed. Extra technical advances include multifloor support, history-based selection of potential fingerprints, high-accuracy positioning using proximity to WiFi APs and Bluetooth LE tags, weighted positioning calculation, and support of different mobile terminals with different radio capabilities, among others.

TABLE 1: Review of related works about hybrid positioning systems with the technologies used (“Parlay” stands for 3GPP Parlay, “Bar.” for Barcode, “BT” for Bluetooth, “Cell” for cellular tower ID, “RFID” for both RFID and NFC, “Mag.” for magnetic, “BT-LE” for Bluetooth Low Energy, “Ine.” for inertial, and “Smart” for smart deployment) and their main features (“HO” stands for handover, “Map” for map-matching, and “Alg.” for positioning algorithm).

Proposal	Positioning technology													Features		
	Parlay	Bar.	BT	Cell	GPS	WiFi	RFID	Mag.	BT-LE	Ine.	Light	Smart	MEC	HO	Map	Alg.
[20]	X												X			
[25]		X	X	X	X	X										Fing/Prox/DR
[26]				X			X									Prox
[27]			X		X	X	X							X		Fing
[33]					X				X	X				X		Trilat/DR
[28]					X	X								X		Fing
[29]								X						X		Kalman
[31]										X						DR/Landmarks
[32]									X	X					X	DR/Kalman/Prox
[34]					X			X		X	X			X		DR/Kalman
Current				X	X	X	X		X			X	X	X		Fing/Prox

2.4. *Progress beyond the State-of-the-Art.* The positioning systems previously cited are summarized in Table 1, together with the current proposal in the last row. As compared with previous works, our solution presents a novel flexible positioning solution based on vitalization advances. The main prominent feature is the offloading of positioning computation from both constrained mobile devices and potentially saturated cloud servers onto a set of MEC nodes, presenting a context-aware solution where location is determined using an adaptive hybrid approach, and positioning-related requests are managed using a novel ICN resolution mechanism powered with SDN. Furthermore, our platform is the one integrating more positioning technologies. Although this is not a great advantage by its own, it reflects the openness of the framework for the integration of other localization sources. In fact, from the solutions analyzed, it is the only one supporting the provision of positioning data coming from external smart infrastructures. Moreover, as it is later explained in detail, the localization algorithm is adjusted to radio sensing capabilities of the target mobile device, and the handover approach supports both the change of technology and the transition between floors.

3. Offloading Solution

The general design of the MEC approach used to offer high-performance and location-aware positioning is shown in Figure 1. Mobile devices are able to acquire positioning or radio measurements from different technologies, such as GPS, cellular ID, WiMAX, NFC, Bluetooth LE, or regular WiFi. When a mobile device requires a fix, data collected from these technologies are affixed to an ICN request to a generic identifier (e.g., *location.um.es/calcPos*), which is routed through the current access network (AN). This request is routed through a set of SDN-managed switches towards a proper MEC node, which replies with the calculated position, an accuracy level, and an expiration time for the fix. Usually, this server is in charge of a particular physical area, containing

positioning training data that will be used to compute the final position. This is particularly relevant when using WiFi fingerprinting, but other technologies not requiring off-line training will also benefit from an enhanced calculation performance.

As described in Figure 1, smart deployment able to detect users through direct authentication (e.g., smart card) or intelligent tracking mechanisms (e.g., mix of initial authentication and tracking) can input this positioning data to the system by sending an ICN request to another identifier, such as *location.um.es/userDetected*. This request is again routed towards the most suitable MEC node, which finally saves the information. By using this ICN approach, users or applications outside the network domains covered by the solution could also send commands for searching for concrete users by using, e.g., the identifier *location.um.es/searchUser*.

In this solution, we have opted for a reactive model, where the position is calculated upon the user request, but a proactive model could be also developed, through a continuous transmission of raw positioning data from the mobile device. However, we assume that a reactive model can save network resources and end-node energy. Regarding the data included as parameters in calculation requests, apart from radio measurements, already calculated positions by the technologies that provide this information, such as GPS or cellular ID, are also included. The hybrid positioning algorithm is then in charge of providing the most accurate position among all the available fixes and the ones computed through the set of implemented algorithms.

Apart from calculating locations and saving training parameters of positioning algorithms, MEC nodes also cache positions in order to reply in a timely way to remote nodes searching for users. It would be also possible for the MEC node to ask the mobile device to provide measurements in order to serve position requests, although this is not included in the current implementation of the system.

As can be seen, the system offers a scalable architecture to accommodate a hybrid positioning algorithm, which

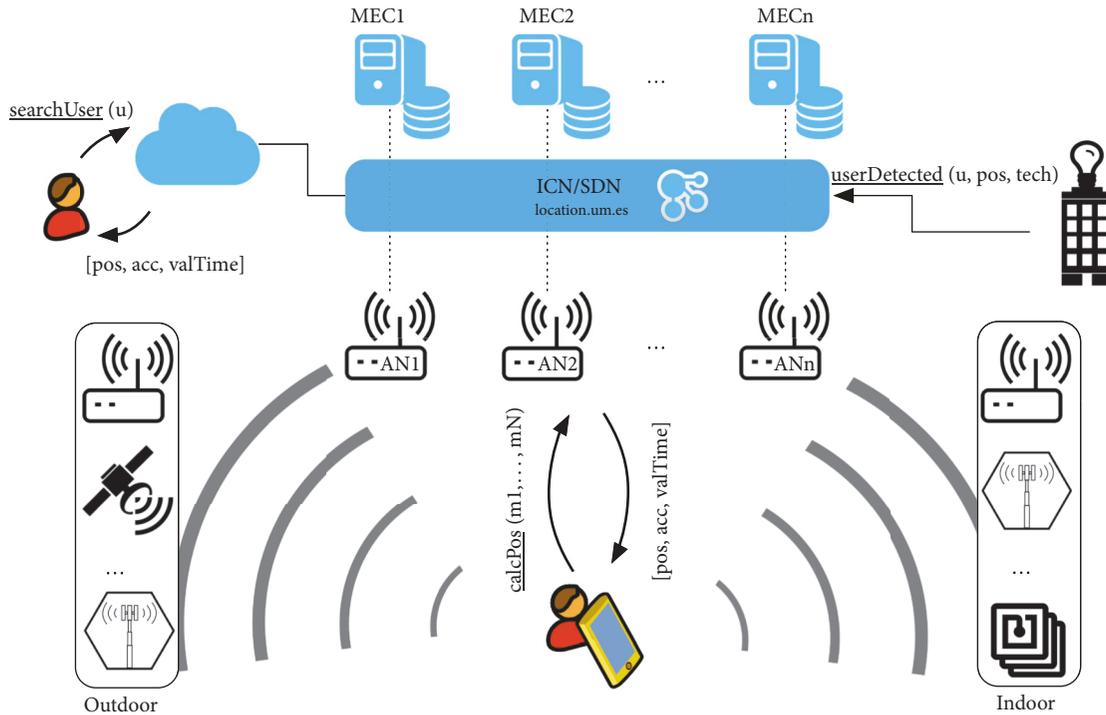


FIGURE 1: Positioning solution using MEC and ICN.

offloads both end-devices and cloud servers. This is done with MEC nodes opportunistically deployed to cover computation requirements at particular geographic areas or building premises. Moreover, the communication with MEC nodes relies on an SDN network, which facilitates the traffic steering needed for load balancing or fault tolerance. This routing could be also performed on the basis of policy or business decisions. More details on this are provided in Section 5.

4. Hybrid Localization Approach

As presented in the general design, the positioning algorithm is executed in MEC nodes. With the aim of solving the deficiencies found in positioning technologies offered by current mobile devices, a hybrid localization approach has been developed on the basis of common positioning sources (GPS, cellular and wireless proximity) and the extra aid offered by indoor positioning algorithms and technologies coming with smart spaces. This way, the most convenient location sources will be chosen, improving the availability and accuracy of the system across different sites.

A flow chart of the main functions of the positioning system deployed in MEC nodes is included in Figure 2. Data collected from smart deployment and radio-frequency positioning technologies coming from the mobile device are depicted on the left, while the main algorithm blocks are on the right.

Regarding measurements from short- and medium-range communication technologies, the signal strength and ID of base stations sensed from the surroundings are the input

for both state-of-the-art fingerprinting and proximity algorithms, over which several improvements have been applied.

The fingerprinting algorithm performs a pattern matching search comparing previously collected fingerprints (in a prior training stage) with current measurements. For the moment, the Euclidean distance is used to obtain the most similar fingerprints. This search is bounded within a configurable threshold radio (default value is 20 meters) around the previous location, in order to offer significant results. The search is also carried out over fingerprints of the same floor when the user is indoors. With the aim of supporting the change between floors, handover areas are modeled with fingerprints of a special type. When the user is detected on these fingerprints, a vertical cylinder-based search is performed using the previously configured radio. Once the list of candidate fingerprints is obtained, a weighted position is calculated on the basis of its Euclidean closeness to the sensed radio environment. Finally, an estimation of the position accuracy is calculated with a mean of all distances between the best fingerprint and each one of the set.

The proximity algorithm detects the closeness to a radio tag or a short/medium-range base station. Depending on the technology and the communication range configured, when the signal strength reaches a certain configurable threshold, which is finally translated to consider five meters by default, the mobile device is said to be positioned at the installation point of the base station or tag. Hence, the database of the MEC server maintains information about the deployed communication devices within its action area. The accuracy parameter used here is the distance sensed to the device.

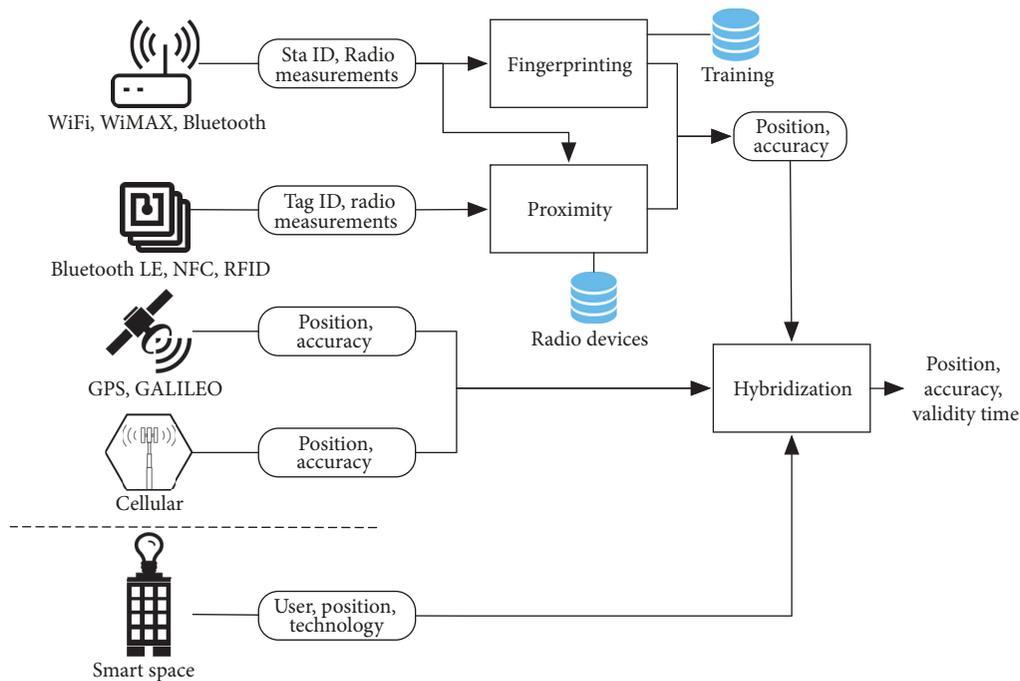


FIGURE 2: Positioning subsystem.

Positioning technologies, such as GPS or cellular, directly input the decision module with the hybridization algorithm. Here the accuracy is usually provided by the technology, such as the position quality indicators of GPS and cellular networks, which is translated by Android in our case to a meter-based value. Regarding position estimations from smart deployment, an identifier of the user detected is needed, together with an identifier of the technology used. This way, it is possible to know the accuracy level and validity time of the position. Door opening systems, for instance, offer a very accurate position, but only useful for a short time.

The hybridization module is finally in charge of deciding the most suitable position to output. Currently, the decision is based on the accuracy level, selecting the location source with less error. This is directly implementable, given that all accuracy parameters are converted to meters. Since a sole technology and algorithm are finally chosen to provide the final position, there are not synchronization issues regarding fusing. In addition, the position generated is complemented with a validity time of the location, which depends on the technology used.

Since the framework has been designed in a completely modular way, the algorithms implemented are treated as complementary positioning sources. This means that new algorithms can be easily implemented and added, such as Kalman-based or particle filter fusing solutions. According to the review presented in Section 2, PDR algorithms using these statistical procedures offer good results, and they do not need prior training and could complement the implementation presented here based on fingerprinting and proximity. For the moment, we have opted for developing practical algorithms that have been shown to work properly in the

literature and in some commercial products. More complex algorithms will further demonstrate the advantage of edge computing to offload end-devices.

The system is also prepared to be used in different mobile devices, since we have detected that radio perception varies with the hardware. Hence, the application used to sense the radio environment is firstly configured with a simple test near known base stations or tags, in order to compute a correcting factor for the signal strength perceived. Moreover, apart from the handover between floors, a second handover mechanism has been designed to support the effective change between technologies, in order to both save battery and prepare the technology beforehand. This is particularly interesting when using GPS, since it can be disabled indoors and activated when the mobile device is detected at especially identified handover fingerprints.

5. Position Request Resolution

As stated before, our solution follows an ICN/SDN approach in order to route the requests to the proper MEC instance. The ICN paradigm is characterized by routing information based on the content and not on the network topology. The advantages of ICN for edge computing services have been presented in works such as [35]. In our case, the MEC positioning scheme introduced in Section 3 leverages in Representational State Transfer (REST) messaging to communicate final devices with the architecture. Therefore, the transport is delegated to HTTP. To that end, our system inspects the URL of HTTP requests, and then an SDN solution is used to redirect traffic and rewrite network and transport level addresses as needed [36]. From the security

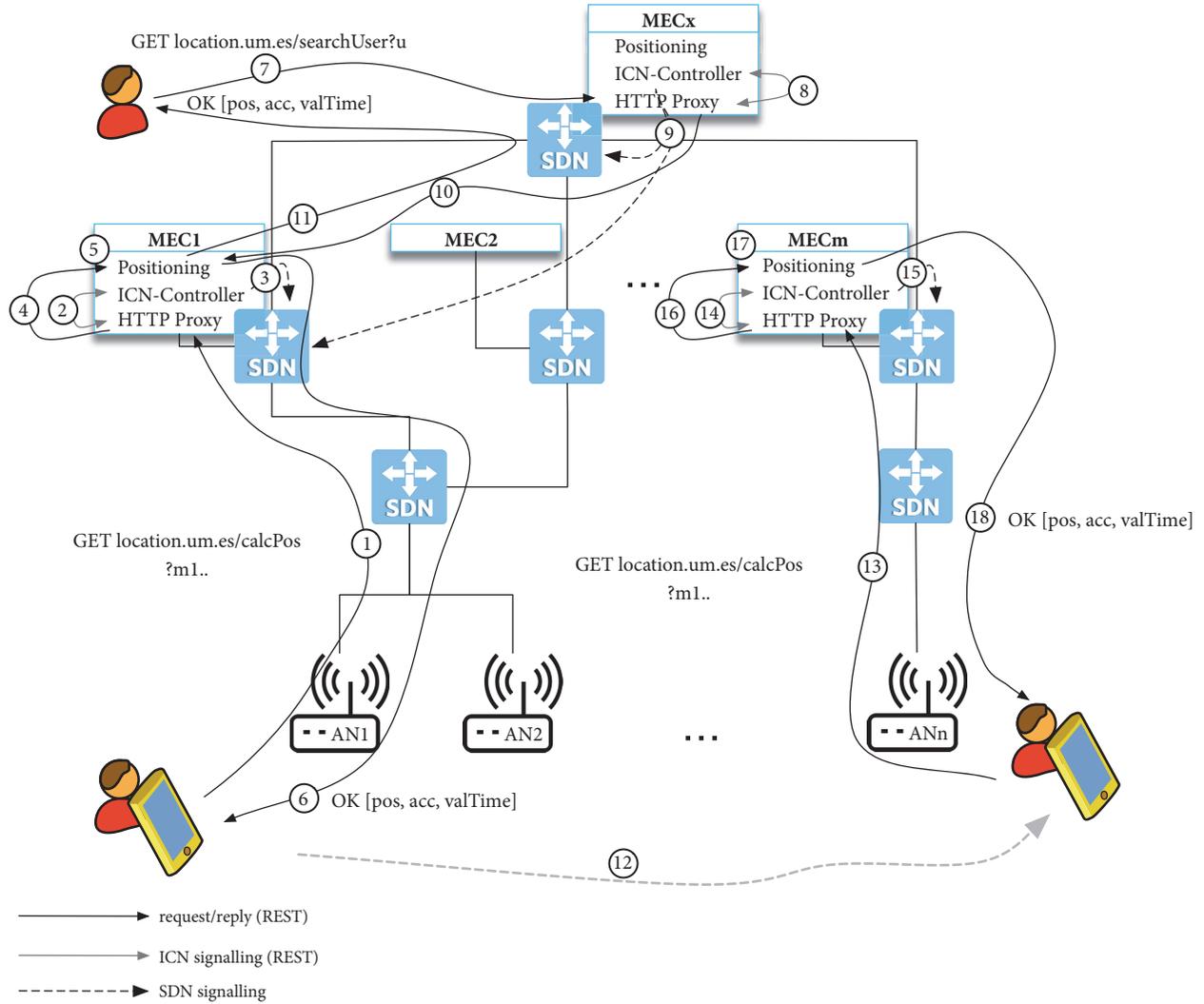


FIGURE 3: Request resolution using the ICN approach based on SDN.

perspective, our solution bets on REST over HTTPS, which offers confidentiality and facilitates the incorporation of application level authentication. From the point of view of the ICN proposal, it only affects the need for deploying certificates on the proxies as it is usually done in CDN.

In order to inspect the URL of the request, a complete TCP 3-way handshake process is needed so that the request is actually transmitted. To unload the SDN/ICN controller from that burden, HTTP proxies are collocated at the MEC nodes, as can be seen in Figure 3, and proactive SDN routing is performed to direct requests to the nearest proxy from the SDN network entry point. The use of proxies to take care of the TCP session management is known as TCP splicing technique. An equivalent approach is presented in [37]. Other ICN proposals like OpenCache [38] could be employed in the proposal but it would require the deployment of specific caching software. The adopted approach relies on already existing HTTP caching software without the need for adapters. Once the proxy receives the URL, it is forwarded through the control plane to the ICN instance, which sits on

top of the SDN controller as an application. We assume a clustered controller deployment in which the SDN controller is deployed on the same MEC instance as the proxy and near to the SDN entry points. The ICN instance decides the positioning unit to which the request will be directed and programs the path to that MEC node (e.g., the nearest one for position calculation and the latest accessed for user location). The path will be closed by the inspection of TCP FIN messages by the controller or by timeout if the former does not happen.

The operation of our architecture is described in detail in three requests in Figure 3:

- (i) In the first one, a positioning request indicated as *location.um.es/calcPos* is issued when the user is near AN1 (step 1). The requests received from any of the ANs are redirected automatically to the nearest HTTP proxy by installing flows in the switches proactively, meaning that the SDN controller is not asked during that part of the communication. The HTTP proxy

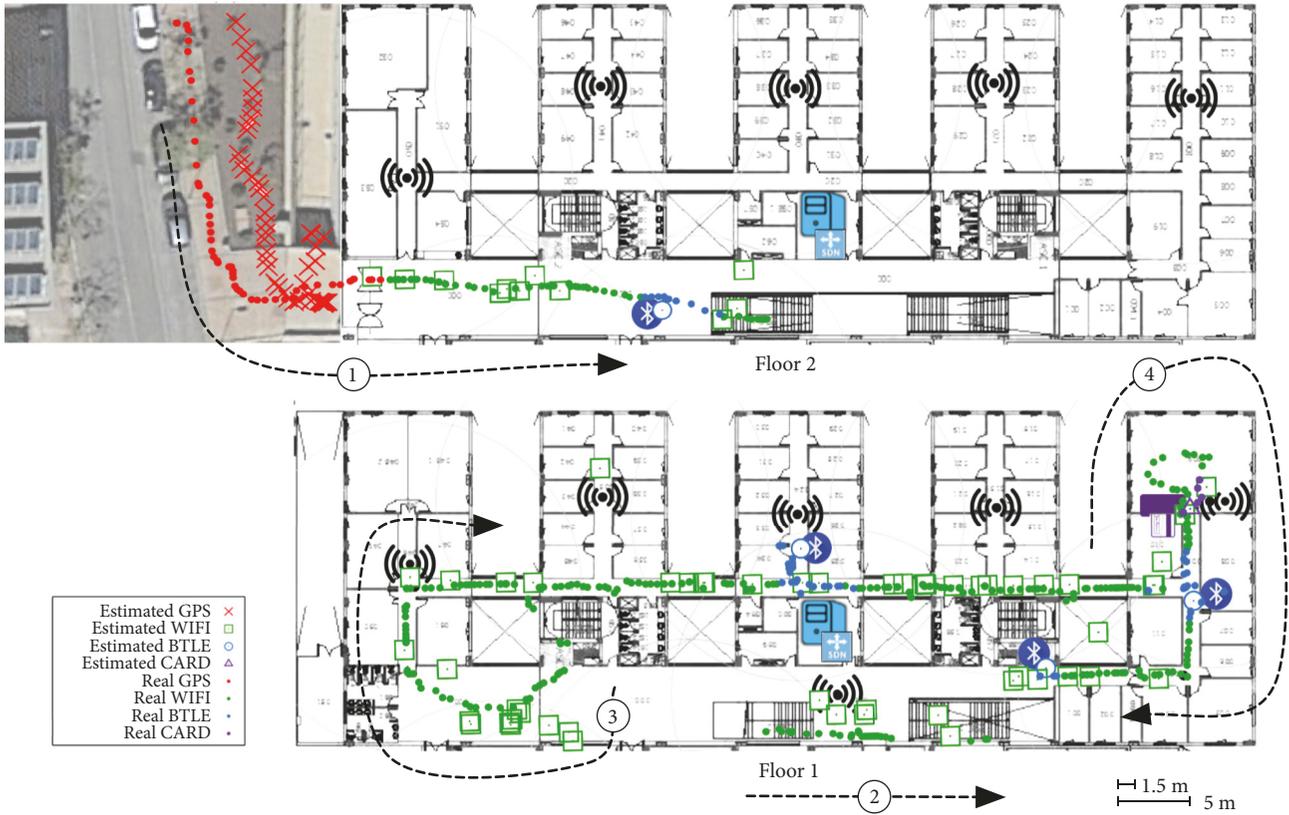


FIGURE 4: Deployment and operation test of the MEC-based positioning solution.

extracts the URL and provides it to the ICN controller sitting on top of the clustered controller instance (step 2). The ICN controller installs the flows (step 3) to redirect the request from the proxy to the proper positioning unit (step 4), which is in the same MEC instance in this case. When the Positioning unit calculates the fix (step 5), the result is returned through the proxy to the user (step 6).

- (ii) The second request is a user searching request, indicated as *location.um.es/searchUser*, which is issued by an external application (step 7). This request arrives directly to the nearest HTTP proxy. Again, the HTTP proxy extracts the URL and provides it to the ICN controller (step 8). In this case, the ICN controller knows the last location of the user (note the *u* parameter in the URL), which was calculated in step 5, so the network is programmed (step 9) to access the positioning instance where the position was calculated in MEC1 (step 10) and finally receive the location (step 11).
- (iii) When the user moves to another AN (step 12), requests are addressed to a different local MEC to compute the location as above (now in steps 13-18).

Notifications of user position from intelligent infrastructures are not included for the sake of simplicity in Figure 3, but they are properly managed. The ICN controller

redirects the requests to the nearest MEC instance's HTTP proxy of the last user's known position. If the user has not ever been previously located, a whole new challenge for electing the positioning unit to which the request should be directed is created, as for cache election in collaborative caching environments. For the moment, our implementation routes the request to the nearest positioning unit to the entry HTTP proxy, with the aim of reducing the response delay.

6. Operation of the Real System

6.1. Deployment. A real testbed has been deployed at the Computer Science Faculty (University of Murcia), where the main network nodes described in the previous sections have been set up. As can be seen in Figure 4, real plans of the building floors are used to show the main architectural components and one of the validation tests. For the sake of clarity, Bluetooth LE tags are depicted with the technology symbol, while WiFi APs are included as points surrounded by waves. The contact-less card reader used in the trials is also shown in purple on the right of the first floor. An SDN-capable switch is used for each floor, to which a MEC server is directly connected using one of the available Ethernet ports. These switches are interconnected with the building switch and then the university backbone. WiFi APs are also wired with the floor switch.

Each MEC node is a Poweredge R430 server running Centos 7. LibVirt/KVM software is used for the positioning subsystem and HTTP proxy virtual machines, and Docker to run the ONOS instances in a cluster. Running a cluster introduces the advantage of distributed storage for the ICN application running on top of it. The SDN switches are HPE Aruba 2920 running firmware WB.16.04.0008 OpenFlow 1.3 instances. A VLAN production network is allocated for the SDN control plane and specific VLANs are employed for each SDN link between switches. The mobile phone used in the trials is a Galaxy S7 with Android Nougat.

The technologies used for positioning are WiFi, GPS, Android *Network* location, which uses WiFi proximity and cellular ID as input, and Bluetooth LE, whose tags are detected when the user approximates at distances of around five meters (about -20 dBm). Additionally, the door opening system of the building is prepared to send positions to the system when the user is identified.

The positioning subsystem has been implemented as a Java application with a REST interface developed in PHP. The ICN/SDN software is a distributed ONOS application (https://gitlab.atca.um.es/jordi.ortiz.um.es/icn_onos_app) with distributed storage running on top of ONOS 1.11 Loon version, while the HTTP proxy used is a python implementation inherited from the GN3plus project (<https://gitlab.atca.um.es/gn3plus/gn3proxy>). An Android application that continuously collects radio measurements and positioning info runs on top of the mobile phone and asks for position calculation every second.

6.2. Validation of the System. One of the trials performed using the positioning system is plotted on the building plans in Figure 4. A prior training for WiFi fingerprinting was performed, creating a grid with a resolution of three meters. Both real and estimated positions are shown in Figure 4. The test starts outside the building, and then the user enters the second floor to go downstairs (step 1). In this first stretch GPS is initially used, presenting a great error due to multipath and signal obstruction of the metal facade of the building. When the accuracy perceived by using WiFi fingerprinting is better than GPS, a technology handover is performed. Now the position accuracy is clearly better. Just before the stairs, a Bluetooth LE tag is detected and used with the proximity algorithm for a while, since it presents an error estimation better than WiFi fingerprinting. At the first floor the user goes downstairs again (step 2). For the sake of clarity the ground floor is not shown. At this floor, the user takes the elevator and returns to the first floor to walk within our department (step 3). At the middle of this stretch the user enters the central corridor and is located with another Bluetooth LE tag. Then he enters a laboratory (step 4), and the door opening infrastructure informs the floor MEC of the user location. Since this position has an optimum accuracy level, it is used the next time the mobile phone requests a location, but only for a while. Finally, the hybrid positioning performs two extra handovers to use Bluetooth LE proximity before finishing the trial.

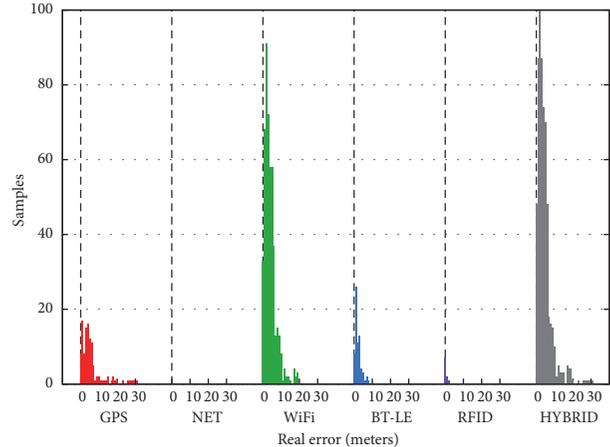


FIGURE 5: Histograms of the error committed for each technology and overall count with the hybrid approach.

TABLE 2: Statistics obtained from the error values collected.

Mean	4.62 m
Maximum	31.03 m
Minimum	0.13 m
Standard deviation	4.08 m
Confidence interval 95%	0.31 m
Samples	664

6.3. Analysis of the Positioning Error. The distribution of errors committed by the positioning system in the previous test is plotted in Figure 5. The histogram labeled with the hybrid solution includes the real error measured by the system, while the rest of plots show the error added by each technology (i.e., the sum of individual technology errors results in the hybrid histogram). As can be seen, most of the calculated positions have an error below ten meters and a great part below five meters. Most of the high-impact errors are due to GPS, since it is used near the building with a metal-made facade that reflects part of the satellite signals. However, there are some errors between 10 and 20 meters that are attributed to WiFi fingerprinting, which present deviations when the user moves near the floor edges, as can be seen in Figure 4 in steps 1 to 3. This is due to the bad distribution of fingerprints surrounding the user at these locations. Although the error is highly impacted by the wide usage of WiFi fingerprinting, overall results are improved by the small areas where Bluetooth LE proximity and NFC are used. Here the errors detected are usually below five meters. The Android Network positioning source was not used, since the accuracy value obtained for this technology (based on cell ID and WiFi proximity) was worse than the accuracy estimations of the rest of technologies. This is the reason why in Figure 5 there are no values in the NET histogram.

A summary of the error results is included in Table 2. From these results we can state that the error of the system is 4.62 ± 0.31 meters in settings equivalent to our testbed. It is important to remark the challenging scenario considered,

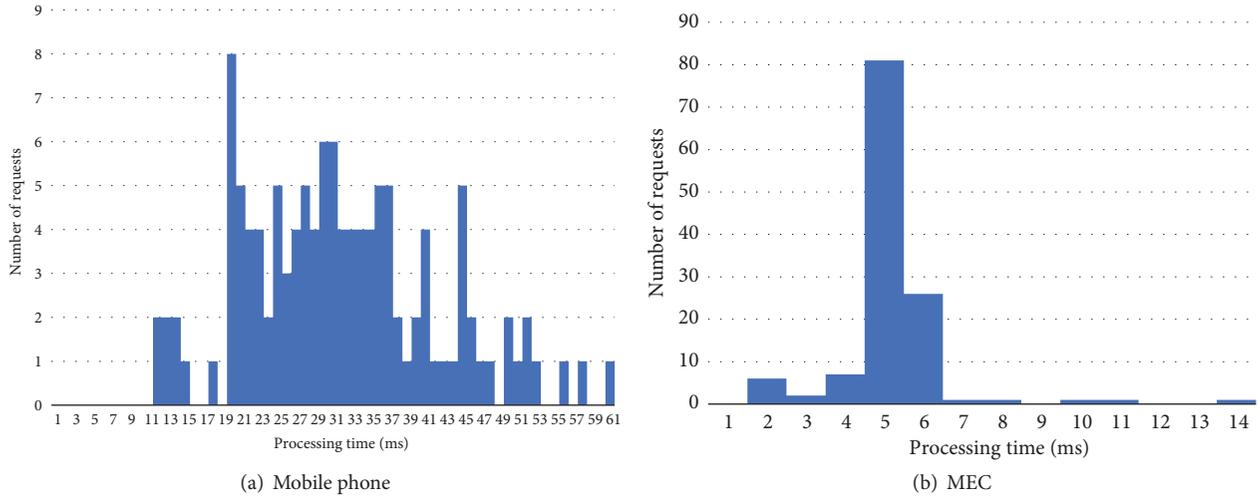


FIGURE 6: CPU time required to compute positions.

where the GPS signal is partially blocked and WiFi fingerprinting, which is the most used positioning source, operates under the worst conditions.

6.4. Performance Evaluation. In order to bound the performance improvement when moving the computation of positions from mobile devices to MEC nodes, a set of latency tests have been carried out to measure the processing time, the scalability of the solution, and the overall latency when requesting position calculations to MEC. The same equipment used in the validation tests has been used, and calculations of positions are exclusively based on the fingerprinting algorithm, since this is the one requiring more resources. A reference database of 445 fingerprints is used for computing each position, without considering any optimization.

The processing time needed to compute positions attending to the required CPU time in the mobile phone and the MEC are shown in Figure 6. Here we show the results when requiring a batch of positions for each device. The two histograms clearly evidence the lower performance of the mobile phone, which requires a time between 20 and 50 ms to compute each fix. This time is greatly reduced at the MEC, which usually requires a time between 2 and 10 ms for such a job.

Nevertheless, it is important to consider that the mobile phone case involves a full distributed computing scenario, whereas using MEC nodes could imply local bottlenecks. For this reason we have evaluated the scalability of the MEC node when requiring the computation of an increasing number of positions, Figure 7. Initially, a single process is launched requiring the computation of 1000 positions, obtaining a performance of 0.83 positions per millisecond. If request were attended in a sequential order, this performance would be more than enough even for a MEC covering a highly crowded area. However, this performance is even improved when launching parallel processes, which better emulates the behavior of the MEC node when attending REST requests. A significant speedup is experienced with two processes,

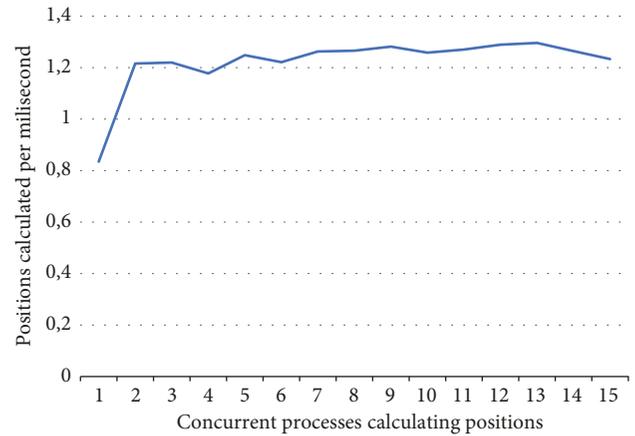


FIGURE 7: Computing performance of MEC under high loads.

given that the positioning virtual machine includes two cores. When including extra processes the performance maintains at a rate near 1.3 positions per millisecond.

With the aim of evaluating the overall latency experienced by users requiring position fixes from their mobile phones, a batch of 6000 positions have been requested from a Microsoft Surface Pro 4 (Intel i5, 8GB of RAM and 256GB of SSD disk), which offers a higher performance than the mobile phone. This way it is possible to generate more requests per second. This device is connected with the WiFi network of the building, as in the validation tests with the mobile phone. The results can be seen in Figure 8, indicating overall latency times between 6 and 18 ms. Considering that from the results shown in Figure 6(a) we obtain an average time of 32 ms for computing a position in the mobile phone and that in 95% of cases the time needed for such task is greater than 20 ms, the speedup using the MEC is clear. These results are not expected to get much worse when requiring extra load from parallel requests, as the previous scalability evaluation demonstrates. It is also important to consider the high-demanding scenario

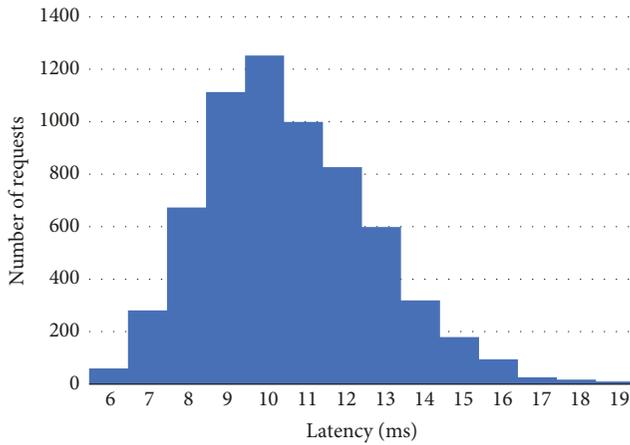


FIGURE 8: Latency of position calculation requests from the mobile device perspective.

considered, which in most of the cases exceeds the required performance levels under regular settings.

The previous validation and performance tests reveal the correct operation of the hybrid positioning system, which is able to attend positioning requests in a distributed way using the ICN/SDN solution. Low error values have been collected even under a challenging scenario that combines indoor and outdoor positioning needs, integrating heterogeneous technologies. MEC servers maintain user location information that is served to mobile devices when it is timely relevant and more accurate than a new computed fix. The performance results indicate that MEC nodes provide a higher performance when using the most demanding algorithm for computing positions (i.e., fingerprinting), and the overall latency for solving requests is impacted by neither the number of parallel requests nor the network, given the closeness of MEC nodes, as compared with potential cloud-based solutions. Moreover, mobile devices are not required to save positioning data regarding the wireless infrastructure or training fingerprints in order to compute positions, and new algorithms could be implemented transparently to end-devices.

7. Conclusions

The work proposes a positioning scheme powered with MEC, in order to improve performance and provide context-awareness computation. A hybrid positioning subsystem is placed at MEC nodes, which are directly connected at the infrastructure network edge. Then, location requests from both mobile devices and external searching entities are routed to them using an ICN overlay network, which is in charge of internally redirecting the request to the most appropriate node considering caching, proximity, and load-balancing strategies. An SDN mechanism is used to create virtual paths to address the requests to the selected nodes.

The general positioning framework, which is extensible to multiple indoor and outdoor technologies and algorithms, has been successfully tested. GPS, WiFi, Bluetooth LE, and

fingerprinting and proximity algorithms have been used in the tests, together with an extra location source from smart deployment. The results obtained in the reference indoor/outdoor scenario present real errors in the range of 4.62 ± 0.31 meters, meaning good position estimations given the challenging settings. The proper operation of the MEC nodes over the ICN network has been checked, experiencing a real-time response from the system. In particular, exhaustive performance tests indicate that the solution scales properly when the number of users in the surroundings of MECs increases, and overall latency for solving position requests are below 15 ms. This time is a half of the time needed for a high-end mobile phone to compute a position, offering the edge computing proposal extra flexibility features in the areas of maintenance, algorithm and positioning data updates, and computation offload from battery-constrained nodes.

The positioning system will be extended with new technologies and algorithms, considering our background in the field [39, 40]. Next steps also include the evolution of the position computing framework to a generic computing platform where processing and data analysis requests from end-devices are launched. This will have a significant impact on the energy and CPU cycles consumed at constrained devices within IoT deployment. Moreover, an intelligent distribution of computing tasks along the network is envisaged, in order to balance the processing of offloaded tasks across the edge network segment and coordinating this effort also with cloud nodes.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work has been sponsored by the Spanish Ministry of Economy and Competitiveness through the PERSEIDES (contract TIN2017-86885-R) and USEIT (contract PCIN-2016-010) projects and by the European Union through the ANASTACIA (contract 731558) and IoTcrawler (contract 779852) projects.

References

- [1] R. F. Brena, J. P. Garcia-Vazquez, C. E. Galvan-Tejada et al., "Evolution of indoor positioning technologies: A survey," *Journal of Sensors*, vol. 2017, Article ID 2630413, 21 pages, 2017.
- [2] R. Mautz, *Indoor positioning technologies [M.S. Thesis]*, ETH Zurich, Institute of Geodesy and Photogrammetry, Department of Civil, Environmental and Geomatic Engineering, 2012.
- [3] N. Alam and A. G. Dempster, "Cooperative positioning for vehicular networks: facts and future," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 4, pp. 1708–1717, 2013.

- [4] V. Pasku, A. De Angelis, G. De Angelis et al., “Magnetic Field-Based Positioning Systems,” *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 2003–2017, 2017.
- [5] L. Mainetti, L. Patrono, and I. Sergi, “A survey on indoor positioning systems,” in *Proceedings of the 22nd International Conference on Software, Telecommunications and Computer Networks, SoftCOM 2014*, pp. 111–120, September 2014.
- [6] P. J. Escamilla-Ambrosio, A. Rodríguez-Mota, E. Aguirre-Anaya, R. Acosta-Bermejo, and M. Salinas-Rosales, “Distributing Computing in the Internet of Things: Cloud, Fog and Edge Computing Overview,” in *NEO 2016*, vol. 731 of *Studies in Computational Intelligence*, pp. 87–115, Springer International Publishing, Cham, 2018.
- [7] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, “Edge computing: vision and challenges,” *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.
- [8] K. Zhang, Y. Mao, S. Leng, Y. He, and Y. ZHANG, “Mobile-edge computing for vehicular networks: a promising network paradigm with predictive off-loading,” *IEEE Vehicular Technology Magazine*, vol. 12, no. 2, pp. 36–44, 2017.
- [9] S. Taherzadeh, A. C. Jones, I. Taylor, Z. Zhao, and V. Stankovski, “Monitoring self-adaptive applications within edge computing frameworks: A state-of-the-art review,” *The Journal of Systems and Software*, vol. 136, pp. 19–38, 2018.
- [10] X. Xu, J. Liu, and X. Tao, “Mobile edge computing enhanced adaptive bitrate video delivery with joint cache and radio resource allocation,” *IEEE Access*, vol. 5, pp. 16406–16415, 2017.
- [11] E. Ahmed and M. H. Rehmani, “Mobile Edge Computing: Opportunities, solutions, and challenges,” *Future Generation Computer Systems*, vol. 70, pp. 59–63, 2017.
- [12] J. Liu and Q. Zhang, “Offloading schemes in mobile edge computing for ultra-reliable low latency communications,” *IEEE Access*, vol. 6, pp. 12825–12837, 2018.
- [13] J. Liu, J. Wan, B. Zeng, Q. Wang, H. Song, and M. Qiu, “A scalable and quick-response software defined vehicular network assisted by mobile edge computing,” *IEEE Communications Magazine*, vol. 55, no. 7, pp. 94–100, 2017.
- [14] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, “A survey on mobile edge computing: the communication perspective,” *IEEE Communications Surveys and Tutorials*, vol. 19, no. 4, Article ID 8016573, pp. 2322–2358, 2017.
- [15] Y. Liu, C. Xu, Y. Zhan, Z. Liu, J. Guan, and H. Zhang, “Incentive mechanism for computation offloading using edge computing: a Stackelberg game approach,” *Computer Networks*, vol. 129, pp. 399–409, 2017.
- [16] M. Goudarzi, M. Zamani, and A. T. Haghighat, “A fast hybrid multi-site computation offloading for mobile cloud computing,” *Journal of Network and Computer Applications*, vol. 80, pp. 219–231, 2017.
- [17] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, “Mobile Edge Computing: A Survey,” *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 450–465, 2018.
- [18] D. Sabella, A. Vaillant, P. Kuure, U. Rauschenbach, and F. Giust, “Mobile-edge computing architecture: the role of MEC in the internet of things,” *IEEE Consumer Electronics Magazine*, vol. 5, no. 4, pp. 84–91, 2016.
- [19] ETSI GS MEC, “Mobile Edge Computing (MEC); Location API,” Tech. Rep. ETSI GS MEC 013, European Telecommunications Standards Institute, July 2017.
- [20] E. Pencheva, I. Atanasov, K. Kassev, and V. Trifonov, “Location service in mobile edge computing,” in *Proceedings of the 9th International Conference on Ubiquitous and Future Networks, ICUFN 2017*, pp. 617–622, Italy, July 2017.
- [21] N. Feamster, J. Rexford, and E. Zegura, “The road to SDN: an intellectual history of programmable networks,” *ACM Sigcomm Computer Communication*, vol. 44, no. 2, pp. 87–98, 2014.
- [22] N. McKeown, T. Anderson, H. Balakrishnan et al., “OpenFlow: enabling innovation in campus networks,” *Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008.
- [23] B. Pfaff, B. Lantz, B. Heller et al., “OpenFlow Switch Specification 1.3.0,” *Current*, pp. 1–36, 2012.
- [24] G. Carofiglio, G. Morabito, L. Muscariello, I. Solis, and M. Varvello, “From content delivery today to information centric networking,” *Computer Networks*, vol. 57, no. 16, pp. 3116–3127, 2013.
- [25] A. Bekkelien and M. Deriaz, “Hybrid positioning framework for mobile devices,” in *Proceedings of the 2012 Ubiquitous Positioning, Indoor Navigation, and Location Based Service, UPINLBS ’12*, pp. 1–7, October 2012.
- [26] G. Lilis, A. Hoffet, and M. Kayal, “GeoAware: A hybrid indoor and outdoor localization agent for smart buildings,” in *Proceedings of the 18th Mediterranean Electrotechnical Conference (MELECON)*, pp. 1–6, 2016.
- [27] M. Ficco, F. Palmieri, and A. Castiglione, “Hybrid indoor and outdoor location services for new generation mobile terminals,” *Personal and Ubiquitous Computing*, vol. 18, no. 2, pp. 271–285, 2014.
- [28] M. Rashidian, H. A. Karimi, M. A. Rajabi, and A. Esmaeili, “Developing a hybrid GPS/Wi-Fi navigation system for user guidance,” *Advances in Computer Science: an International Journal*, vol. 5, no. 3, pp. 12–20, 2016.
- [29] G. De Angelis, A. De Angelis, V. Pasku, A. Moschitta, and P. Carbone, “A hybrid outdoor/indoor Positioning System for IoT applications,” in *Proceedings of the 1st IEEE International Symposium on Systems Engineering, ISSE 2015*, pp. 1–6, Italy, September 2015.
- [30] H. Lemelson, M. B. Kjærsgaard, R. Hansen, and T. King, “Error Estimation for Indoor 802.11 Location Fingerprinting,” in *Location and Context Awareness*, vol. 5561 of *Lecture Notes in Computer Science*, pp. 138–155, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.
- [31] X. Wang, M. Jiang, Z. Guo, N. Hu, Z. Sun, and J. Liu, “An indoor positioning method for smartphones using landmarks and PDR,” *Sensors*, vol. 16, no. 12, 2016.
- [32] X. Li, J. Wang, and C. Liu, “A bluetooth/PDR integration algorithm for an indoor positioning system,” *Sensors*, vol. 15, no. 10, pp. 24862–24885, 2015.
- [33] K. Tabata, H. Konno, K. Tsuno et al., “The design of selective hybrid positioning by utilizing accuracy information for indoor-outdoor seamless positioning and verification in tokyo station,” in *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pp. 1–7, 2015.
- [34] Q. Zeng, J. Wang, Q. Meng, X. Zhang, and S. Zeng, “Seamless Pedestrian Navigation Methodology Optimized for Indoor/Outdoor Detection,” *IEEE Sensors Journal*, vol. 18, no. 1, pp. 363–374, 2018.
- [35] R. Ravindran, X. Liu, A. Chakraborti, X. Zhang, and G. Wang, “Towards software defined ICN based edge-cloud services,” in *Proceedings of the 2013 IEEE 2nd International Conference on Cloud Networking, CloudNet 2013*, pp. 227–235, USA, November 2013.

- [36] J. Ortiz, P. Martinez-Julia, and A. Skarmeta, “6. information-centric network for future internet video delivery,” in *User-centric and Information-centric Networking and Services: Access Networks and Emerging Trends*, 2018.
- [37] S. K. Fayazbakhsh, Y. Lin, A. Tootoonchian et al., “Less pain, most of the gain: Incrementally deployable ICN,” in *Proceedings of the Annual Conference of the ACM Special Interest Group on Data Communication on the Applications, Technologies, Architectures, and Protocols for Computer Communication, ACM SIGCOMM 2013*, pp. 147–158, China, August 2013.
- [38] P. Georgopoulos, M. Broadbent, B. Plattner, and N. Race, “Cache as a service: Leveraging SDN to efficiently and transparently support video-on-demand on the last mile,” in *Proceedings of the 2014 23rd International Conference on Computer Communication and Networks, ICCCN 2014*, China, August 2014.
- [39] M. V. Moreno-Cano, M. A. Zamora-Izquierdo, J. Santa, and A. F. Skarmeta, “An indoor localization system based on artificial neural networks and particle filters applied to intelligent buildings,” *Neurocomputing*, vol. 122, pp. 116–125, 2013.
- [40] J. Santa, R. Toledo-Moreo, M. A. Zamora-Izquierdo, B. Úbeda, and A. F. Gómez-Skarmeta, “An analysis of communication and navigation issues in collision avoidance support systems,” *Transportation Research Part C: Emerging Technologies*, vol. 18, no. 3, pp. 351–366, 2010.



Hindawi

Submit your manuscripts at
www.hindawi.com

