Enabling Social- and Location-Aware IoT Applications in Smart Cities

M. Govoni[1], J. Michaelis[2], A. Morelli[1], N. Suri[2,3], M. Tortonesi[1]

[1] Department of Engineering, University of Ferrara, Ferrara, Italy
{marco.govoni,alessandro.morelli,mauro.tortonesi}@unife.it
[2] United States Army Research Lab (ARL), Adelphi, MD, USA
{james.r.michaelis2.civ,niranjan.suri.civ}@mail.mil
[3] Florida Institute for Human and Machine Cognition, Pensacola, FL, USA
nsuri@ihmc.us

Abstract: In the last decade, governments, municipalities, and industries have invested large amounts of funds on research on smart cities with the main goal of developing services to improve people's quality of life. Many proposals focus on a Cloud-centric network architecture in which all the data collected from a myriad of sensors devices is transferred to the Cloud for processing. However, this approach presents significant limitations when faced with the formidable traffic generated by the Internet of Things and with the need for low-latency services. The deployment of IoT devices in compact groups, connected to the smart city network infrastructure by relatively powerful "gateways", opens the possibility to depart from the centralized architectures and move the computation closer to the data sources. To this end, this paper proposes SPF, a new middleware solution that supports IoT application and service development, deployment, and management. SPF runs IoT services on capable devices located at the edge of the network and proposes a programming model that enables to take advantage of decentralized computation resources in a seamless fashion. SPF also leverages an information dissemination solution designed for constrained network environments and adopts Value-of-Information based methods to prioritize transmission of essential information.

Keywords: Internet-of-Things (IoT); Smart Cities; social- and location-aware IT services.

1 Introduction

Governments, municipalities, and industries around the world are investing large amounts of money to improve the quality of life in modern cities, as numerous research projects in the field of smart cities demonstrate [1]. The Yokohama Smart City Project (http://www.city.yokohama.lg.jp/ondan/english/yscp/) and the LIVE Singapore project (http://senseable.mit.edu/livesingapore/) in Asia, the SmartSantander (http://www.smartsantander.eu/), CITYKEYS (http://citykeys-project.eu), and Open Cities (http://www.opencitiesproject.org) projects in the European Union, and the City

adfa, p. 1, 2011.

Science Initiative in the USA (https://sap.mit.edu/article/standard/city-science-initiative-media-lab) are just a few examples of the many ambitious research projects that focus on one or more of the six aspects that, in accordance with the official European Union website on smart cities (http://www.smart-cities.eu), characterize modern urban realities: environment, living, mobility, governance, economy, and people [1].

Many research efforts and commercial solutions for smart cities propose complex network architectures with a Cloud-centric infrastructure [2] [3] [4]. At the edge, a multitude of sensors acquire raw data, which is continuously transferred over the Internet to the Cloud, where enough resources are available for storage and processing. From there, consumers (including citizens and policy makers) can access derived information by connecting to IT services hosted in Cloud data centers.

Despite showing early promise, Cloud-based architectures present significant limitations for use in emerging smart city infrastructures. Usage of heterogeneous and pervasive sensing and computing resources in IoT infrastructures deployments of IoT devices in smart cities produce a huge amount of information, with forecast studies predicting that, by 2019, IoT devices will generate 507.5 ZB of data per year [5]. It is very costly and inefficient to transfer, process, and store this formidable amount of raw data in Cloud-based data center.

At the same time, it is possible to envision a new generation of social- and location-aware IT services that leverage fully decentralized communication and information processing infrastructures to deliver functions that improve the citizens' quality of life significantly. Through better efficiency in existing health care environments, for instance, next-generation wellbeing oriented services can leverage remote health provision, enabling people to monitor their own health during the day, and better management of conditions such as stress [6].

In terms of preventive care, new services could take advantage of wearable activity trackers like FitBit (www.fitbit.com), which monitor heartbeat, levels of aerobic activity, and hydration status, and cross-correlate that information with the data provided by large, free Cloud datasets like OpenData (www.opendatafoundation.org), to look for and analyze specific trends and unpredictable anomalies. The study of activity tracker data across a city could give valuable insights on the "walkability" and the public safety of particular areas, favoring the detection of areas that have poor quality sidewalks, where residents suffer from a higher risk of robbery/assault, or have a lower quality of health.

In smart cities, IoT devices are often deployed in groups such as sensing systems and connected using short-range and low-power wireless communications, e.g., IEEE 802.15.4 or Bluetooth LE. These IoT networks are connected to the smart city networking infrastructure through one or more "gateway" devices that build on top of capable microprocessors, e.g., ARM Cortex A, and enable the execution of sophisticated and computationally hungry services, while still remaining fairly energy efficient.

In IoT infrastructures, gateway devices represent a promising location to deploy information processing tasks, with continuous reconfiguration of the task allocation according to real-time environmental conditions and service characteristics. In order to fully unleash the potential to develop disruptive innovations in IT services, there is the need for new models for information processing, information dissemination, and application programming.

This paper presents an analysis of opportunities and challenges involved in the development of social- and location-aware IoT services in smart city environments. The manuscript then introduces SPF, a new middleware solution to support IoT application and service development, deployment, and management. SPF enables application developers to take advantage of decentralized computation resources in a seamless fashion. To this end, SPF runs IoT services on devices at the edge of the network, proposes a programming model, leverages information dissemination solution designed for constrained network environments, and adopts Value-of-Information based concepts to prioritize information transmission.

2 Next-generation IoT Services in Smart Cities

In a smart city, as depicted in Fig. 1, applications, storage, and processing capabilities are typically concentrated in Cloud data centers at the core of the network. Connected through heterogeneous communication means to the smart city infrastructure, edge networks include Wireless Sensor and Actuators Networks (WSANs), WiFi and other public networks that provide free Internet access to the mobile citizens, smart grids for smart energy management that connect factories, buildings, and houses, smart roads with sensors and actuators to monitor and manage traffic, and so forth. The fast-paced deployment of a myriad of heterogeneous IoT devices (e.g., traffic cameras, actuators, and sensors for measuring CO_2, temperature, and acoustics) and the increasing presence of personal wireless devices such as wearables and smartphones in smart city environments both represent an increasing burden for communications infrastructures [7]. The functionality of smart city systems is directly impacted by communication infrastructure capability, in-turn affected by growing data transmission to and from Cloud services. Users might experience slower service response times, real-time data processing and information transmission to client systems might become unfeasible, and user-submitted requests may be dropped. Such conditions are particularly unacceptable for time-sensitive applications like emergency-response.

One of the most interesting approaches to reduce network bandwidth consumption and thus increase the quality of services offered to the citizens is to move data processing and computation towards the edge of the network. This idea is at the basis of the *fog computing* concept [8], which aims at integrating computational and storage resources on the Cloud with those available at the network edge. However, fog computing proposes the same application programming and management model of Cloud computing, which was designed for reliable and high-bandwidth networking and does not interact well with the heterogeneous wireless communications at the edge of the smart city network infrastructure. Instead, it is critical that any proposed solution that supports social good-oriented IoT services takes into consideration existing design features and infrastructure of smart cities, selectively adapting and extending these resources as needed.

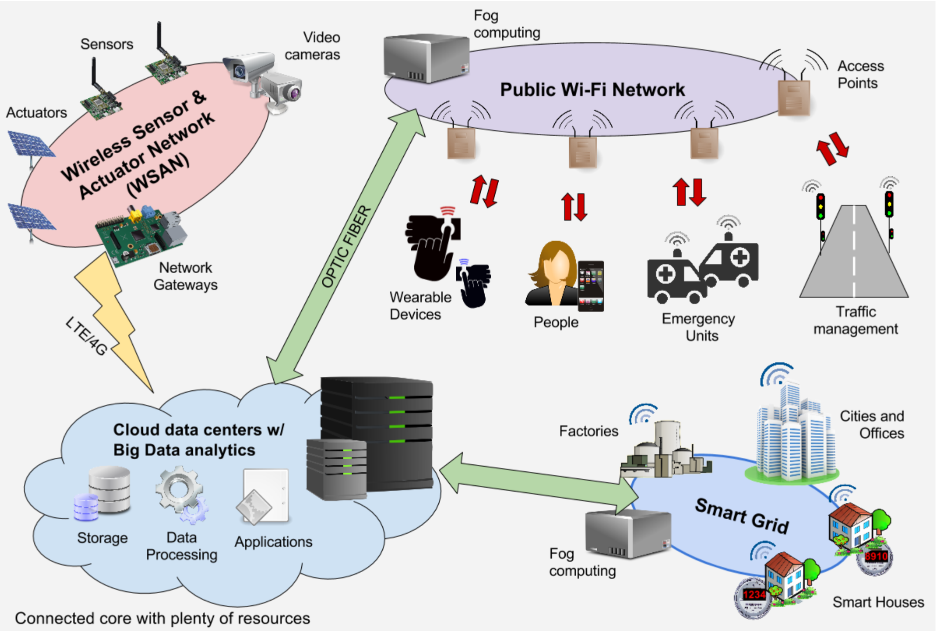

Figure 1 – Typical architecture of a Smart city.

The pervasive computing scenario enabled by IoT technology goes beyond the "decentralized data centers" vision proposed by fog computing and stands to enable the development of a new generation of IT services that significantly improves quality of life within smart cities. In fact, large and high-density IoT installations create a distributed sensing and computation infrastructure for deploying *a wide range of information-centric services* in response to the citizens' needs, whose deployment may be either planned in-advance, e.g., to support a public event [9], or unplanned / impromptu, e.g., emergency services in case of a flash mob or a subway fail.

Dynamically instantiated and short-lived services will typically perform computationally light operations on real-time data, implementing social- and context-aware information processing and dissemination. Such services will typically execute on "gateway" units deployed in proximity of the users and support devices with low computational and memory resources (such as wearable and portable gadgets) by enabling computation offloading. At the same time, resident and/or long-time running services might perform more computationally intensive operations on data collected during long periods and from a number of different sources (e.g., for traffic control and anomaly detection), also taking advantage of Cloud-based resources.

Many innovative, social good-focused IoT services will be designed around their users, with strong social components and results that depend greatly on the location of both the requesting user and the data source. Their social-awareness and information-centric nature will cause IoT services to depart from the usual one-to-one communication model in favor of the one-to-many model, which represents a better fit for social applications and other citizen-oriented services that need to communicate with a group of people/devices (e.g., public safety or emergency alerts).

The dynamic and heterogeneous nature of next-generation IoT services and the smart city environment calls for an information-centric programming model and a corresponding platform that enable and simplify the deployment and management of applications and information processing tasks across the smart city. This would considerably reduce the times and costs for the allocation/deallocation of resources to/from specific services, for instance to respond to peaks in the demand or idle times, and for the on-demand deployment and instantiation of new services, to address needs that arise only in specific situations, e.g., during a concert, a sport match, or any other social event. In addition, information-centric platforms for IoT services could provide developers with the possibility to register and deploy their own applications. In this case, those solutions need to offer a set of basic features on which applications can be built and expose a well-defined API to allow applications running on users' devices to issue requests. Besides that, the API will also have the essential task of hiding any complexities that might derive from the system architecture, the location of remote applications with respect to the data sources, the allocation/deallocation of resources, etc.

Therefore, there is the need for solutions to manage network resource consumption and reduce traffic between the nodes at the edge and the Cloud. The final target is to maximize the usefulness of the offered services as perceived by the citizens of the smart city by having a scalable system that is capable to adapt to changes in the demand and make a smart usage of all the available resources.

3 SPF

SPF (as in "Sieve, Process, and Forward") is a middleware solution for the development, deployment, and management of dynamic IoT applications in urban computing environments [9]. SPF adopts a distributed computation approach that aims at addressing the continued growth of IoT data collection by applying needed processing at the edge of the network, in close proximity to the data source.

The two main components of SPF are the SPF Controller, deployed in the core part of the smart city infrastructure, and the Programmable IoT Gateway (PIG), which is deployed in several "gateway" devices at the edge/core border of the smart city networking infrastructure, as depicted in Fig. 2. The SPF Controller provides an information-centric programming model and an accompanying development toolchain that allow to define IoT applications and to deploy them on the PIGs, where they are actually executed. At the same time, the SPF Controller is responsible for receiving service requests from application users, identifying the most appropriate course of action in terms of IoT services to activate or reconfigure, and dispatching corresponding instructions to relevant PIGs. This allows for dynamic behavior that triggers the execution of information processing and dissemination only when they are actually needed.

PIGs provide both information processing and dissemination functions, which leverage the set of filtering and communication functions implemented by the software platform according to the instructions received by the SPF Controller. PIGs could be deployed directly on the gateway nodes that connect 6LoWPAN networks to the Internet or on dedicated hardware placed in the gateway nodes proximity.

In order to save processing resources on the PIGs and further reduce network usage, SPF employs a content-based filtering on the input data. More specifically, when new data arrives at the PIG from the WSAN, it goes through a filter component that compares the new piece of information with a reference, i.e., the last piece of information processed by the PIG. A difference threshold τ determines if the difference between the new data and the reference is significant. In a positive case, the PIG processes the new data, which then becomes the new reference. Note that SPF allows each application to define the value for τ that best suits determined targets.

Following data processing, the PIG can deliver the extracted information to the requesting users. SPF relies on the DisService component of the Agile Computing Middleware (ACM) for the dissemination of responses [10]. DisService is a P2P communication middleware that handles the dissemination of the information using ad hoc communication links to set up a P2P network and deliver messages to nodes. DisService leverages network interfaces such as Bluetooth, WiFi, and other device-to-device (D2D) techniques that have already proven effective in pursuing mobile offloading [11] [12]. This way, DisService enables the offloading of the network infrastructure and ensures delivery even when the mobile network is congested or the signal is absent or bad, e.g., when one cell provides connectivity for too many devices. In addition, DisService enables the fine-tuning of delivery policies, thus allowing applications to reply with responses that are user-specific, or with responses that target larger sets of people.

The SPF platform identifies three main figures (or stakeholders): SPF administrators, application developers, and users. SPF administrators have the important task of managing deployed SPF platforms, and other tasks that include: deployment of gateways, allocation of resources for the applications, and configuration of the SPF controller. Application developers are responsible for defining and configuring IoT applications. Finally, the users need to install client versions of the IoT applications on their devices to have access to the SPF platform. Applications send requests to SPF Controllers and receive responses from the PIGs. The typical SPF user will be on a mobile device, so application requests will normally arrive at the SPF Controller via 4G/WiFi networks, while responses can reach their destinations using the peer-to-peer (P2P) ad hoc networks composed of nodes running SPF. Additionally, SPF allows for the definition of multiple user types, each one with different priorities and permissions, such as emergency units, police, and citizens. This enables the definition of applications that only a specific set of users can run and eases the allocation of bandwidth for dissemination based on the type of user that made a request. In the context of smart city infrastructures, declaration of user types can aid in prioritizing bandwidth usage for particular purposes, such as emergency response or law enforcement.

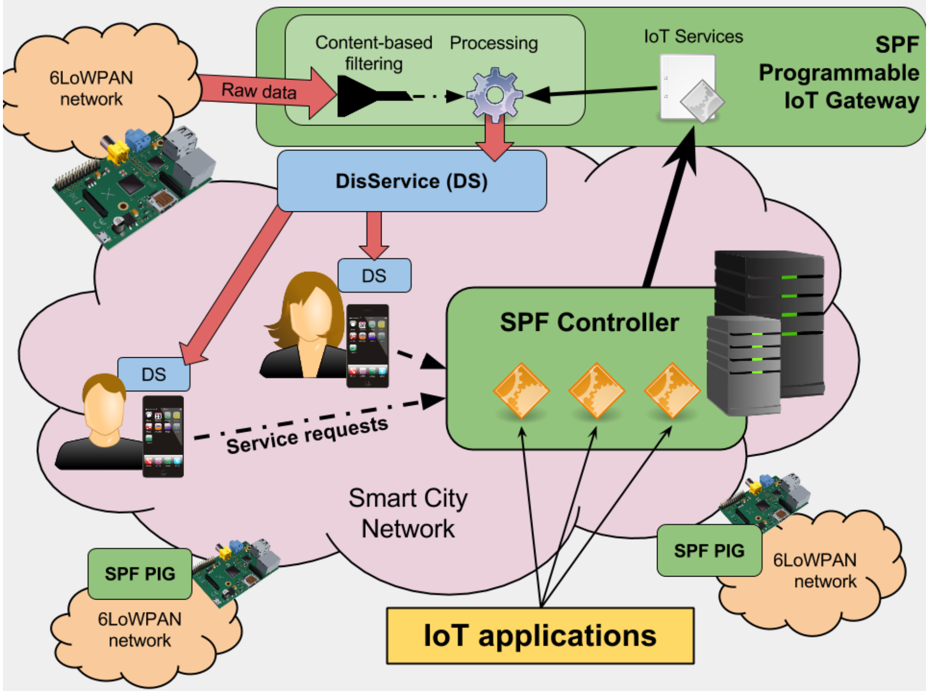

Figure 2. Adoption of SPF in smart city scenario.

4 Application Development Lifecycle in SPF

SPF defines an IoT Application as a collection of related IoT services with the same priority and the same target users. SPF allows application developers to define IoT applications and takes care of their installation and dynamic activation (and deactivation) on the PIG components. This enables the definition of several concurrent applications, each with different services and priority levels.

SPF provides developers with a dedicated Domain Specific Language (DSL) that allows the rapid definition of IoT applications and services. Each application has several aspects that can be configured, such as name, priority level, a set of allowed service types provided to users, and a set of service configurations and dissemination policies. In this way, application developers can differentiate between critical and best-effort applications, define how the application deals with user service requests, and define which dissemination policies are needed.

In SPF, IoT services are developed according to a strictly information-centric perspective: each service is implemented by 2 components that run in a sequential fashion and respectively define the processing of raw data (typically collected from IoT sensors) and the dissemination of the resulting Information Objects (IOs) behaviors for the service. Low-level raw data manipulation functions are provided by specialized components of SPF, namely the information processors, or *pipelines*. The lifetime of services mainly depends on users' requests: applications receive user service requests forwarded by an SPF Controller and activate services on PIGs accordingly, in an on-demand fashion. In addition, if users do not request a specific service for a certain amount of time, the PIG deactivates that service, which will be reactivated only upon reception of new user service request, thus allowing resource saving.

Whenever a PIG first activates a service, that service registers itself with all required pipelines. From that point on, and until the service is deactivated, running pipelines continuously analyze raw input data to obtain higher-level IOs to feed the registered services. This phase could also involve data reduction, compression, and discretization, resulting in IOs smaller than their corresponding raw data and containing only relevant information.

Once a service is deactivated, it unregisters itself from all pipelines. This also determines the pipelines' lifecycle: the PIG keeps pipelines active as long as they still have services registered with them; when all services have unregistered from some processing pipeline, the PIG deactivates it. Examples of information processors are the Optical Character Recognition (OCR), face recognition, car recognition, object tracking, and audio identification pipelines.

Another fundamental aspect in SPF is the reuse and sharing of pipeline and services. In SPF, applications can define services that take as input the output of another service and, similarly, it is possible to design processing pipelines that work in series with other pipelines. Ultimately, this enables the creation of cascades of reusable components and the sharing of processing resources in SPF. These characteristics contribute to give a very dynamic behavior to SPF, with services and pipelines as true autonomous entities that cooperate among them and that consume resources only when needed.

The dissemination of IOs follows a prioritization rule that takes into account the Value of Information (VoI) of IOs. VoI is a measure of the estimated utility of information to consumers based on their situational context, which represents one of the most promising evolutions for information filtering and prioritization in IoT applications. Services calculate VoI according to various factors: some of them are common between all services, but there is also the possibility to define service-specific factors. There are four common parameters: *Application Priority (Pa)*, *Normalized Number of Requests (RN)*, *Timeliness Relevance (of Request) Decay (TRD)* and *Proximity Relevance (of Request) Decay (PRD)*. If available, SPF also takes into consideration the geographic distance between a consumer and the location corresponding to an IO to compute its VoI. Besides common factors, developers can also define service-specific (SS) factors for VoI calculation. For example, the calculation of the VoI of an Audio Identification service could also involve an accuracy parameter that represents the quality of the audio match. The value of such a parameter could be provided, for instance, directly by the Audio Identification pipeline.

Based on the factors discussed above, SPF defines the following formula for VoI calculation:

$$VOI(o, r, t, a) = SS(o) * PA(a) * RN(r) * TRD(t, OT(o)) * PRD(OL(r), OL(o))$$

where *o* is an Information Object, *r* the requestor recipient, *t* the current time, *a* the application, and *OT* and *OL* are operators that return the time and location of origin of objects and requestors, respectively. The result is the tuple <*IO, VoI*>, which is dispatched to the Dissemination Component for forwarding.

5 Conclusions and Future works

The decentralized computing and information-centric approaches adopted by SPF seem to be effective in enabling the development of social good-oriented and citizen-focused IoT applications and services. Comforted by the first positive results, we are planning to extend our work in several directions.

One direction to be further investigated in SPF is on the usage of semantics based methods for defining and supporting IoT applications. Semantic Web technologies [13] focus on enabling both integration of data and corresponding machine interpretation, through use of Ontologies (structured representations of domain knowledge) and reasoning engines. In prior IoT research efforts, usage of semantics in data representation has been applied in a variety of settings that include: dynamic service discovery [14], pervasive computing infrastructures [15], and context-aware asset search [16].

We are also planning to evaluate the adoption of the ICeDiM middleware (http://endif.unife.it/dsg/research-projects/icedim) as an alternative to DisService for information dissemination. ICeDiM is an innovative solution that leverages the concept of virtual dissemination channels with tunable permeability to facilitate the delivery of public and/or unclassified information and, at the same time, enable the constraining of sensitive information to a subset of authorized devices.

Another interesting future objective is to define an extended, acceleration-aware programming model for IoT applications and services that allows their efficient execution on high performance gateways with accelerator-based heterogeneous hardware. More specifically, the acceleration-aware programing model should provide developers with abstractions and functions to write code that can be easily and efficiently run in a parallel fashion on a wide range of parallel hardware platforms and whose parallelism can be safely changed at run time (acceleration-friendly code). This programming model will also provide functions that enable the code to inquire at run time the current computational resources available on the hardware platform and request the PIGs to increase or decrease the execution parallelism dynamically, e.g., for performance, cost, and/or energy saving purposes (acceleration-aware code). This would enable to take advantage of highly innovative, computationally capable, and relatively low-energy consuming hardware solutions based on neuro-morphic processors (such as IBM’s True North Chip), hybrid CPU/manycore (such as Adapteva’s Parallela board) or CPU/FPGA architectures (such as Xilinx’s Zynq-7000 SoC), thus significantly improving the performance of IoT applications.

6 References

1. R. Khatoun, S. Zeadally, “Smart cities: concepts, architectures, research opportunities”, Communications of the ACM, Vol. 59, No. 8, pp. 46-57, July 2016
2. Y. Fu, S. Jia, J. Hao, “A Scalable Cloud for Internet of Things in Smart Cities”, Journal of future trends in computing. Vol. 26, No. 3, pp. 63-75, October 2015
3. A. Alamri, W. S. Ansari, M. M. Hassan, M. S. Hossain, A. Alelaiwi, M. A. Hossain, “A Survey on Sensor-Cloud: Architecture, Applications, and Approaches”, International Journal of Distributed Sensor Networks, Vol. 9, No. 2, pp. 917-923, February 2013
4. Huawei Technologies Co., ltd., “Huawei Smart City Solution”, white paper, available online at: http://enterprise.huawei.com/ilink/cnenterprise/download/HW_315743
5. CISCO, “Cisco Global Cloud Index: Forecast and Methodology, 2014–2019 White Paper”, April 2016, available online at : http://www.cisco.com/c/en/us/solutions/collateral/service-provider/global-cloud-index-gci/Cloud_Index_White_Paper.pdf
6. P. Biggs, J. Garrity, C. LaSalle, A. Polomska, R. Pepper, “Harnessing the Internet of Things for Global Development”, 8th annual Conference on Information and Communication Technologies for Development (ICT4D), 2016, available online at: http://www.itu.int/en/action/broadband/Documents/Harnessing-IoT-Global-Development.pdf
7. A. B. Zaslavsky, C. Perera, D. Georgakopoulos, “Sensing as a Service and Big Data”, Proceedings of the International Conference on Advances in Cloud Computing (ACC), July 2012
8. A. V. Dastjerdi, R. Buyya, "Fog Computing: Helping the Internet of Things Realize Its Potential," in Computer, Vol. 49, No. 8, pp. 112-116, August 2016
9. M. Tortonesi, J. Michaelis, A. Morelli, N. Suri, M. A. Baker, "SPF: An SDN-based middleware solution to mitigate the IoT information explosion", IEEE Symposium on Computers and Communication (ISCC 2016), pp. 435-442, 2016
10. G. Benincasa, A. Morelli, C. Stefanelli, N. Suri, M. Tortonesi, "Agile Communication Middleware for Next-generation Mobile Heterogeneous Networks", IEEE Software, ISSN 0740-7459, Vol. 31, No. 2, pp. 54-61, March-April 2014
11. A. Orsino, G. Araniti, L. Militano, J. Alonso-Zarate, A. Molinaro, A. Iera., “Energy Efficient IoT Data Collection in Smart Cities Exploiting D2D Communications”, Sensors, Vol. 16, No. 6, June 2016
12. A. Morelli, C. Stefanelli, N. Suri, M. Tortonesi, “Mobility Pattern Prediction to Support Opportunistic Networking in Smart Cities”, 6th International ICST Conference on MOBILe Wireless MiddleWARE (Mobilware 2013), November 2013
13. P. Barnaghi, W. Wang, C. Henson, K. Taylor, "Semantics for the Internet of Things: Early Progress and Back to the Future", International Journal on Semantic Web and Information Systems (IJSWIS), Vol. 8, No. 1, pp. 1-21, 2012
14. S. Chun, S. Seo, B. Oh, K. Lee, "Semantic Description, Discovery and Integration for the Internet of Things", 2015 IEEE International Conference on Semantic Computing (ICSC), pp. 272-275, 2015
15. J. Kiljander, A. D’elia, F. Morandi, P. Hyttinen, J. Takalo-Mattila, A. Ylisaukko-Oja, J. Soininen, T. S. Cinotti, "Semantic Interoperability Architecture for Pervasive Computing and internet of things", IEEE access 2, Vol. 2, pp. 856-873, 2014
16. C. Perera, A. Zaslavsky, P. Christen, M. Compton, D. Georgakopoulos, "Context-aware Sensor Search, Selection and Ranking Model for Internet of Things Middleware", IEEE 14th International Conference on Mobile Data Management, Vol. 1, pp. 314-322, 2013