Design and Evaluation of a Semantic Gateway Prototype for IoT Networks

Francisco L. de Caldas Filho
Cybersecurity INCT, Electrical Eng. Dept., University of Brasilia
Brasilia, DF, Brazil
francisco.lopes@uiot.org

Lucas M. C. e Martins
Cybersecurity INCT, Electrical Eng. Dept., University of Brasilia
Brasilia, DF, Brazil
lucas.martins@redes.unb.br

Ingrid Palma Araújo
UIoT Laboratory, Electrical Eng. Dept., University of Brasilia
Brasilia, DF, Brazil
ingrid.palma@uiot.org

Fábio L. L. de Mendonça
Cybersecurity INCT, Electrical Eng. Dept., University of Brasilia
Brasilia, DF, Brazil
fabio.mendonca@redes.unb.br

João Paulo C. L. da Costa
Cybersecurity INCT, Electrical Eng. Dept., University of Brasilia
Brasilia, DF, Brazil
joaopaulo.dacosta@ene.unb.br

Rafael T. de Sousa Júnior
Cybersecurity INCT, Electrical Eng. Dept., University of Brasilia
Brasilia, DF, Brazil
rafael.desousa@redes.unb.br

ABSTRACT

In Internet of Things (IoT) networks, an IoT gateway is a type of middleware used to solve issues related to device variability, such as the heterogeneity of communication protocols, as well as to respond to segmentation and modularization needs in such networks. This paper proposes a semantic gateway for IoT, a middleware capable to forward data streams to and from devices by means of different protocols such as MQTT, Socket and even non-IP device specific protocols. Also, the proposed middleware, which includes a REST/JSON programming interface for other middleware and for IoT applications, is considered to be semantic because it has the ability to receive data from sensors that communicate using different protocol suites, such as the TCP/IP stack and wireless sensor network protocols, and then performs data transformations, semantic intermediation and routing to the cloud. The proposed gateway is designed as a component to allow the application of fog computation principles in IoT networks. Based on tests performed with a developed prototype, this paper presents evaluation results regarding the functionality and the performance of the proposed gateway.

CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability;

KEYWORDS

IoT; IoT gateway; Semantic gateway; Middleware; Fog computing.

ACM Reference Format:
Francisco L. de Caldas Filho, Lucas M. C. e Martins, Ingrid Palma Araújo, Fábio L. L. de Mendonça, João Paulo C. L. da Costa, and Rafael T. de Sousa Júnior. 2017. Design and Evaluation of a Semantic Gateway Prototype for IoT Networks. In *UCC'17 Companion: UCC '17: 10th International Conference on Utility and Cloud Computing Companion, December 5–8, 2017, Austin, TX, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3147234.3148091

UCC'17 Companion, December 5–8, 2017, Austin, TX, USA

ACM ISBN 978-1-4503-5195-9/17/12…$15.00
https://doi.org/10.1145/3147234.3148091

1 INTRODUCTION

Internet of Things (IoT) is a paradigm for building computer systems distributed throughout the Internet, in which, in principle, the most diverse devices, objects and things will be connected and interacting with applications to extend various services to people [4, 13]. Thus, it is envisaged that IoT can allow people and things to be connected at any time, anywhere, with anything or anyone, ideally using any path and any device [17].

As of now, following this paradigm, there is a great number of things connected to the Internet and this number should continue to grow vertiginously [13], so that it can be effectively believed that IoT allows to create a world where numerous objects are connected to the Internet and communicate with each other requiring minimum human intervention [17], so as to succeed in providing services that are resourceful and ingenious to human beings [20].

In this context, however, one of the obstacles to the development of IoT is the lack of standardization of communications, especially considering the heterogeneity of protocols for communication with the devices and the high variability of these devices, which leads to the so-called interoperability crisis in IoT [6]. Indeed, it is common that sensor manufacturers adopt closed standard protocols and/or different messaging protocols, which prevent or make their utilization by third-party applications difficult, thus causing semantics issues. This problem becomes more serious for devices that operate only on layer two of the OSI-RM[1], so they depend on the gateway to send data that is used by IoT applications. Moreover, there is still no consensus on the syntax and semantics of the conversions performed in that component[6].

An IoT gateway is usually one of the middleware types used to solve device variability problems and the heterogeneity of protocols for these devices communications, as well as to respond to segmentation and modularization needs of such networks. This paper proposes an Semantic Gateway for IoT, capable of forwarding

[1]Open Systems Interconnection Reference Model) from ISO (International Organization for Standardization

data streams to and from devices through different protocols such as MQTT[2], Socket and even for devices without IP transmission capability. The proposed gateway offers an interface through directed REST[3] / JSON[4] calls which will be used by various other middleware and IoT applications.

The means employed by the proposed gateway, including device and services registration, authentication, token-based identification management and abstraction of devices services, constitute together a semantic intermediation process that allows the devices to be seen as registered smart objects able to interact with applications, by abstracting physical details of the devices and their communication procedures.

To describe the proposal and the results obtained, besides this introduction, this work is organized as follows: in Section 2, we present related work. In Section 3, we present the Semantic Gateway proposal and, in Section 4, the validation results based on a prototype evaluation. We finish in Section 5, with a brief conclusion.

2 RELATED WORK

With the advent of IoT, the variety of objects connected to the Internet and the associated data volume in the Web are increasing [13]. In this sense, a Semantic Gateway does not only solve the interoperability problem, through data classification, data translation and the processing of heterogeneous information belonging to the fog, but, mainly, it also identifies and registers each unknown device and/or service in the IoT network as an unique virtual object. As illustrated in Figure 4, the strategy of unique identification for services and devices that can either be physical or virtual, even if they are immersed in different contexts, allows the proposed Gateway to optimize and manage the semantic interoperability in the IoT network instance.

The base of numerous IoT system architectures comprehends, in general, smart objects and protocols responsible for data manipulation and transportation [3]. Examples of these protocols are described in the Subsection 3.1. The vision of smart objects, a concept enveloping sensors and actuators, includes intelligent behavior, performed under restrictions related to power, energy, memory, processing and bandwidth, as discussed in [1, 22, 23]. Those limitations may prevent such objects from connecting directly to middleware systems and applications, which operate on the full TCP/IP stack. This problem is due to the lack of standardization of the messaging protocols. Thus, it is necessary the intermediation of a gateway with the ability to extract information from the operations performed, translating and transforming the different data formats [2].

Authors in [24] argue that the interoperability between heterogeneous devices and applications, specially when these are inserted in different contexts, is a challenge correlated to project standards and technologies, such as SOA and Web Services. However, the cited paper discusses how the semantic interoperability is solved, usually, through ontologies, in approaches where the challenge is the use of different ontological models.

Also, in [24], the authors use the term "user interoperability framework (UIF)" to explain the development of a prototype to discover services and devices in an IoT network. Such technique, which is analogous to our proposal, utilizes the classification of ontological contexts and the separation of devices and users in three distinct classes: i) real, in which each device has its own metadata and ontological model; ii) virtual, in which each user has a set of devices and their respective services; and iii) regular, in which a set of devices shares the same semantic model. One of the UIF models main objectives is to catalog existing services in a way that they can easily be discovered (evidenced) and associated to the relevant users and devices.

The same paper [24] also argues that, despite the use of open standards as IEEE 802.15.4, ZigBee and WirelessHART, the incompatibility among these protocols is still high. Nevertheless, the focus on the utilization of ontologies to promote the semantic interoperability in such situations is not considered in our approach, because when the interoperability is treated with web technologies, the devices with computational restrictions are excluded from the model, and the use of other middleware architectures is still necessary for an IoT environment [7–10]. In this sense, we understand that the semantic must also be treated beyond the service/middleware layers (or transport and application layers, in the OSI-RM). In other words, we suggest to optimize the semantic from the sensorial layer (physical, link and network layers, in the OSI-RM).

Paper [6] suggests a semantic gateway for the description of smart objects using ontologies and description logics to enable semantic interoperability utilizing, specifically, the technologies MQTT and CoAP[5], which are application layer protocols for IoT. Thereby, endpoint devices can communicate by XMPP, MQTT or CoAP protocols, but, since those protocols are not interoperable semantically, the solution again requires a semantic gateway capable of translating the different messages received. Furthermore, the authors emphasize that the different network protocols that work up to the layer two of the OSI-RM, such as Bluetooth, ZWave and ZigBee. The communication between this elements of different networks happens through a gateway.

The paper [19] proposes a mobile gateway that receives and manages data and information related to the health of monitored patients remotely through a network of heterogeneous sensors in a dynamic topology. The authors propose a mobile IoT gateway in an ecosystem that supports the operation of Intelligent Personal Assistants and their integration into ubiquitous computing environments in the context of health-care scenarios. That gateway platform receives and manages data and information of patients monitored remotely by a network of heterogeneous sensors in a dynamic IoT topology and provides users with the ability of controlling and monitoring the data of a patient. Although [19] discuss only the 3G and Bluetooth technologies, the authors developed a model of gateway capable to interact with devices from different manufacturers. In our proposal, the devices services abstraction is independent of the application sector.

3 IOT GATEWAY IMPLEMENTATION PROPOSAL

For the description of the proposal, an abstraction of the structure of an IoT network instance is used according Figure 1. It is

[2]Message Queue Telemetry Transport.
[3]Representational State Transfer.
[4]JavaScript Object Notation.
[5]Constrained Application Protocol.

formed mainly by the following components: *things*, *gateways* and *middleware*. Things refer to physical and virtual objects in the IoT ecosystem.

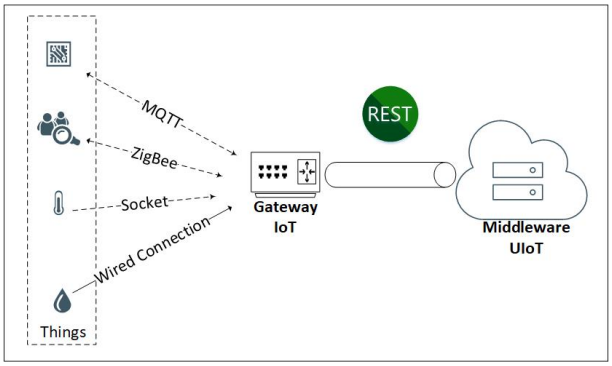


Figure 1: Abstraction of an IoT network instance

Thus, the gateway has the function of translating, authenticating and authorizing the data packets and devices requests to the upper layers (e.g.: middleware), allowing, through the integration of different protocols, control and management of devices and their respective services, working in a cohesive and coordinated way in the interconnection between IoT applications and devices [18] over the Internet. In this way, according to [4, 6], the Gateway is a crucial element for the IoT architecture, being responsible for receiving the data of the sensors using different transmission technologies, such as Radio Frequency, Bluetooth, LoRAWAN[6], 6LoWPAN[7], and for performing the necessary conversion for communications through the full stack of IP protocols.

Another important role of the IoT Semantic Gateway is the admission control of new devices and services [4, 6, 21], receiving requests from the objects to join the network, validating these demands with an authentication server and, depending on the response, performing the inclusion of the new member or preventing the requester from being part of the new network.

These capacities, which typically would otherwise be made in a cloud computing IoT middleware, being implemented in the gateway, are then near the edge of the IoT instance, and, since a group of such gateways can communicate through an IoT middleware and provide services to IoT applications, the utilization of these gateways follow a fog computing model.

In our proposal, the Semantic Gateway, called UIoT Gateway, has a modular structure comprising a Channel Handlers Module, a Coordinator Module, and a Middleware Handlers Module, as shown in Figure 2.

The *Channel Handlers Module* contains the handlers of each communication channel supported by the gateway. In this module, a channel handler is responsible for receiving the sensor and WSN data and passing them on to the Coordinator Module that will handle the data.

When it is necessary to enable some hardware component to perform communications, such as wireless ZigBee, Bluetooth and

[6]Low-Power Wide Area Network.
[7]IPv6 over Low-Power Wireless Personal Area Networks.

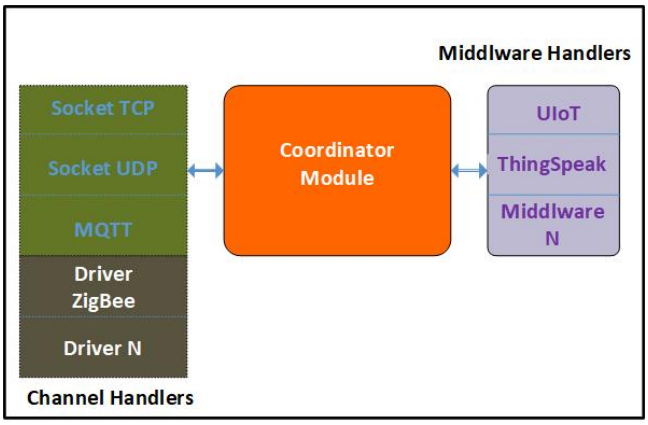


Figure 2: UIoT Gateway architecture

433 Mhz communications, the handler needs a software component similar to an operating system driver. Section 3.1 brings more details about this component.

The channel handler is responsible for defining the processes and means for reading the message content and identifying the customer. For example, if the communications occurs through the IP protocol, the handler can use the MAC and/or the IP address of its client. For ZigBee devices with no IP support, the ZigBee Channel Handler needs to use some feature available on the channel, such as the device physical address.

The UIoT Gateway prototype that was implemented for the validation of our proposal supports the following channel types: MQTT, as a messaging protocol, Socket TCP, Socket UDP and ZigBee.

The *Coordinator Module* is responsible for the translation of messages formats, the control of objects admission and buffer utilization. Based on the information received from the channel handler, the Coordinator component parses the message, identifying the services specified by the sending device, and preparing the message for the proper Middleware handler.

At the other end, the *Middleware Handlers Module* provides a set of interfaces for to be plugged into IoT middleware so that they can communicate with the corresponding objects. By this means, the Gateway can connect its managed objects to other existing IoT platforms, making it easier to send data from these objects to the concerned middleware. Details of this process are described in Section 3.3

3.1 Communication interfaces

The gateway is crucial for most IoT applications, since there is currently no secure and consolidated communication infrastructure, specifically via Low Power Wide Area Network, between numerous devices and the Internet. This communication is still practically unfeasible, either due to the restricted technological capabilities of the devices and the mentioned problems of semantic, scalability, interoperability, cost, performance, privacy and security required by the IoT network.

Therefore, the proposed Gateway marks the logical separation of the drivers belonging to the upper layers (e.g. middleware and application) from those pertinent to the Perception and Network

Layers, specifically the Physical Layer and MAC Sublayer. These drivers/protocols enable the Gateway to recognize and handle different IoT devices without any additional technical interference to translate the connection with end nodes. This principle is similar to the generic drivers of operating systems.

Drivers for IEEE 802 and RF 433 MHz[8]. The IoT Gateway aims to support the well-known IEEE 802.15.4, 802.15.1 and 802.11 standards for communication with Zigbee, Bluetooth and/or Wi-Fi devices.

In addition to basic transmitting and receiving operations, there is the demand management for the channel and the handling of device transceivers used by the network (e.g. optical, infrared or radio frequency). In this sense, the IoT Gateway design gives preference to using the 433 MHz Radio Frequency Transmitter and Receiver module (RF 433), since this solution, although consuming more energy due to the modulation type, data rate and transmission energy, is the one that has the highest acceptance being very ease to use.

3.2 Communication between device and Gateway

The proposed Semantic Gateway implements the *Device-to-Gateway* communication model, set by RFC 7452[9], to support the communications and data transport between the devices and the cloud. The challenge in this model is to ensure the IoT network ubiquity by using generic protocols, fostering interoperability, control and better device management.

In this model, IoT devices need to send identification data along with sensor originated data inserted into a single, pre-built string which comprises five fields separated by semicolons, the first three of which contain information about the device and the last two for the service.

The three fields in the first part of the string are used to identify the device for the middleware. These fields focus on identifying the network device according to its characteristics, such as physical address, network address and name.

The second part of the string carries the name of the service and the measured value. Each device can perform several simultaneous measurements, such as temperature, humidity, and soil moisture. Also, a device can have actuators which will receive external commands, such as turning the light on or off, turning the engine on or off, and so. Each of the services offered by the device must be given an unique Middleware ID. Since sending a measurement can be done at different times, the device must use for each measurement a single communication operation to send the value.

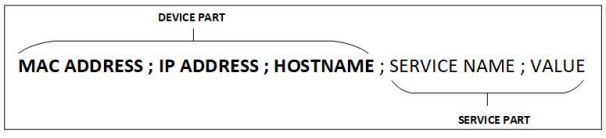


Figure 3: String Format

[8]ISO/IEC 18000-7:2014 defines the air interface for Radio Frequency Identification (RFID) devices operating as an active RF tag in the 433 MHz band used in item management applications

[9]https://tools.ietf.org/html/rfc7452

Upon receiving the string, whose format is detailed in Figure 3, the Gateway is responsible for performing the process of admission and control of the devices and services associated with it.

3.3 Communication between Gateway and a Middleware server

Once the sensor data has been received by the defined channel handler and the controller has already performed the necessary processing of that information, it is forwarded to the handler responsible for communicating with the IoT Middleware. For example, data from a device communicating through the MQTT protocol is received by the MQTT Channel Handler and then passed to the Coordinator Module and then to the chosen middleware. The proposed semantic gateway currently uses the UIoT Middleware module called RAISe, which was defined in [20].

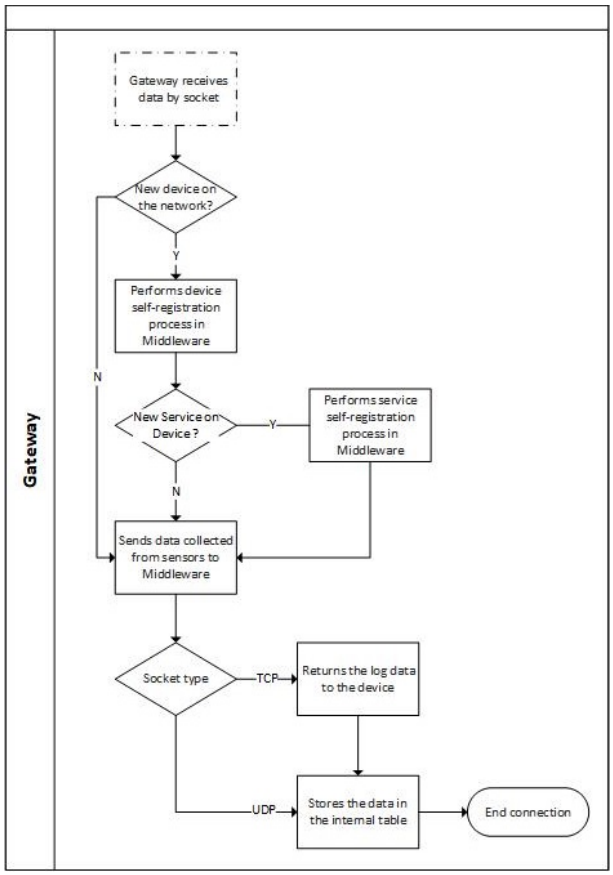


Figure 4: Self-registration decision flow performed by the UIoT Gateway

It is important to note that the middleware handler is responsible for implementing each middleware specific feature, leaving the rest of the gateway generic enough to be used by other middleware without being affected by any middleware idiosyncrasies. For the purpose of validation tests, our prototype gateway implements a RAISe Handler (RH) which is the module responsible for handling the UIoT middleware.

According to the RAISe security protocol, which is defined in [21], a device is only able to send or receive data when it has a valid token. In this way, to ensure that the device is registered, the RH performs the process described in Figure 4.

Thus, when receiving the data in the pre-built string in Section 3.2, the Gateway will check, through the information contained in the device portion, whether this device is already part of its set of managed devices. If not, it performs the self-registration process for that device and its services.

Then, after enabling the device to the RAISe middleware, RH starts sending the readings made by the corresponding sensors to that middleware.

As well as other IoT middlewares described by [5, 12, 14–16, 25, 26], RAISe provides its interface via REST services, with the communication content in the JSON format. In this way, the communication managed by the RAISe Handler is completely carried out in REST. Since the RAISe REST API primitives are synchronous, the RH sends the HTTP request to the middleware, receives the result of that request, and the returns it to the Coordinator Module. This last in turn performs the treatments and validations regarding the registration and the security of the transaction and sends the result to the handler of the channel used by the device.

4 RESULTS

To validate the proposal, we developed a gateway prototype, named UIoT Gateway, and nine smart objects. This prototype runs in a Raspbery Pi 3 under Debian Linux.

For the sake of our validation, the physical devices were built in Arduino and ESP8266 and were put in operation, for twenty-four hours, seven days a week, sending data pertaining to fifteen different services to the UIoT Gateway.

In order to increase the volume of devices and test the load capacity of the Gateway, we elaborated Python programs to simulate smart devices, creating a scenario where the number of sensors grows in a short period of time and they remain sending data for an interval of two hours. Table 1 summarizes this test scenario.

In this scenario, we monitor the amount of sensors and services that send data, as presented in Figure 5a, as well as the CPU and memory consumption in the Raspberry that runs the Gateway. As can be seen in Figure 5b, the use of the hardware resources is not directly related to the growth in the number of devices and services, remaining stable over time.

5 CONCLUSIONS

As [4], points out, a gateway is a fundamental component of the IoT architecture because it connects the most diverse devices to the components responsible for data processing.

This paper presents a proposal for an Semantic gateway that works on two fronts. First, it intermediates the connection of heterogeneous devices to the Internet, even in the case of different protocols being used by these devices. The proposed gateway provides compatibility, conversion, and necessary treatment to match these different configurations. Second, it abstracts the middleware that can be used by the devices, allowing the middleware to be chosen independently of the choice of components and the topology of the IoT network. As a consequence, it is possible to use different IoT platforms and data visualization services.

The experimental results show that the Semantic Gateway is efficient in processing and forwarding data from a large volume of objects, thus complying with the construction characteristics described in this paper. We emphasize that our design operates with heterogeneous devices, throughout different protocols and channels.

As future work, we envisage studies on the following issues:

Redundancy and ambiguity. Utilization of redundancy in the IoT Gateway, using the Virtual Router Redundancy Protocol (VRRP) and synchronizing the device and service tables, thus allowing a Backup Gateway to take over all functions of the main Gateway, in case of failures.

According to [24], we must ensure the registration of new devices and services through the Gateway without any semantic ambiguity.

Messaging semantics. One interesting idea is to convey semantic metadata in the header and in the message to make the proposed gateway capable to identify the protocol being used and to reason about the communication content.

Ontology. The application of ontologies is crucial for a successful semantic interoperability among things, protocols, middleware and applications. An ontology can be used for their semantic annotation, managing access, resource discovery and knowledge extraction. [11] points out several ontological models for IoT.

Traffic optimization. There are interesting possibilities to gain performance using streaming optimization techniques, such as segmenting data into queues, performing packet prioritization on certain queues, and queuing compression and deduplication for transmission.

Middleware abstraction. As described in the proposal, the gateway must intermediate smart devices and the UIoT middleware. Thus, another aspect that can be addressed is to connect the gateway to other middleware but avoiding impacts to the operation of the devices. With such feature, it is possible to connect to various middleware available in the market, such as Amazon AWS IoT and the IBM Watson Internet of Things. Also, as suggested in [6], the gateway can interface with data visualization services such as Xively and Thinkspeak.

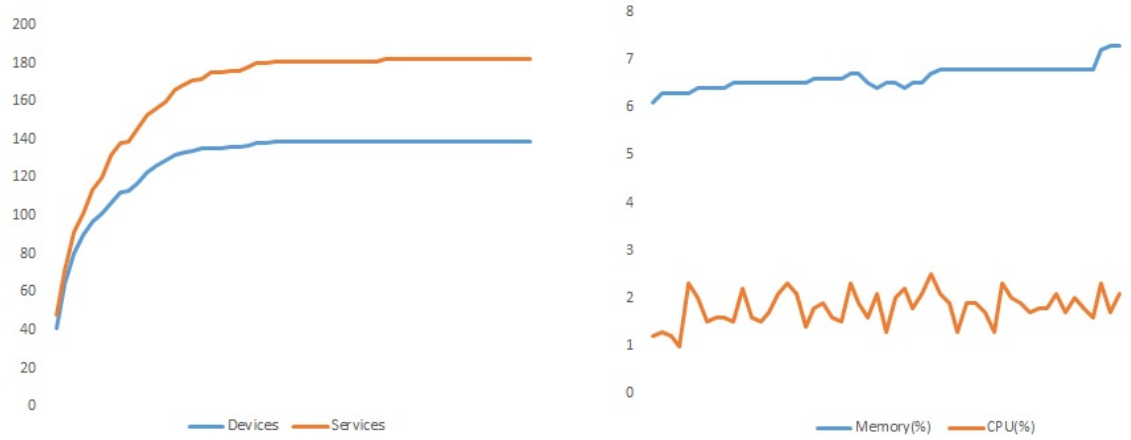
Other drivers and protocol. In order to increase the range of devices and WSNs reached, it would be interesting to incorporate other protocols such as Bluetooth, NRF24L01, CoAP and 6LowPAN. To enable these protocols, it is necessary to implement specific drivers and handlers for each one.

Atomic vs persistent connection. A promising study is to perform communications between the devices and the gateway via WebSocket, since this technology eliminates the need for devices to make requests at all times. It is interesting to evaluate this approach on the devices and at the gateway by comparing it to the currently used atomic connections.

Discovery. A promising idea is to implement discovery protocols such as Universal Plug and Play (UPnP), which allows the inclusion

Table 1: Configuration of the devices used in the test

Number of devices	Number of services	Type	Protocol	Services description	Interval (sec)
1	2	Arduino	Socket TCP	Soil moisture, luminosity	120
2	3	Arduino	MQTT	Temperature, humidity, luminosity	30
1	5	Arduino	MQTT	Current, power, temperature, humidity, luminosity	20
3	3	Arduino	ZigBee	Temperature, humidity, luminosity	30
2	2	ESP8266	Socket UDP	Temperature, humidity	10
42	1	Virtual	MQTT	Random integer values	30
42	1	Virtual	Socket UDP	Random integer values	30
44	1	Virtual	Socket TCP	Random integer values	30



(a) Amount of devices and services over time

(b) Memory and CPU usage over time

Figure 5: Experimental data summary

of devices that do not have the capacity to send data in the format preestablished by the gateway.

ACKNOWLEDGMENTS

This research work has the support of the Brazilian research and innovation Agencies CAPES – Coordination for the Improvement of Higher Education Personnel (Grant 23038.007604/2014-69 FORTE – Tempestive Forensics Project), CNPq – National Council for Scientific and Technological Development (Grant 465741/2014-2 Science and Technology National Institute – INCT on Cyber Security), and FAPDF – Research Support Foundation of the Federal District (Grant 193.000976/2015), as well as the Brazilian Ministry of Planning, Development and Management (Grants 005/2016 DIPLA – Planning and Management Directorate, and 11/2016 SEST – State-owned Federal Companies Secretariat) and the DPGU – Brazilian Union Public Defender (Grant 066/2016).

REFERENCES

[1] Antar Shaddad Abdul-Qawy, PJ Pramod, E Magesh, and T Srinivasulu. 2015. The Internet of Things (IoT): An Overview. *International Journal of engineering Research and Applications* 1, 5 (2015), 71–82.

[2] Ala Al-Fuqaha, Mohsen Guizani, Mehdi Mohammadi, Mohammed Aledhari, and Moussa Ayyash. 2015. Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys & Tutorials* 17, 4 (2015), 2347–2376.

[3] Zainab H Ali, Hesham A Ali, and Mahmoud M Badawy. 2015. Internet of Things (IoT): Definitions, Challenges and Recent Research Directions. *International Journal of Computer Applications (0975–8887) Volume* (2015).

[4] Eleonora Borgia. 2014. The Internet of Things vision: Key features, applications and open issues. *Computer Communications* 54 (Dec. 2014), 1–31. https://doi.org/10.1016/j.comcom.2014.09.008

[5] Francesco G. Brundu, Edoardo Patti, Anna Osello, Matteo Del Giudice, Niccolò Rapetti, Alexandr Krylovskiy, Marco Jahn, Vittorio Verda, Elisa Guelpa, Laura Rietto, and Andrea Acquaviva. 2017. IoT Software Infrastructure for Energy Management and Simulation in Smart Cities. *IEEE Transactions on Industrial Informatics* 13, 2 (2017), 832–840.

[6] Pratikkumar Desai, Amit Sheth, and Pramod Anantharam. 2015. Semantic gateway as a service architecture for iot interoperability. In *Mobile Services (MS), 2015 IEEE International Conference on*. IEEE, 313–319.

[7] Hiro G. C. Ferreira. 2014. Arquitetura de Middleware para Internet das Coisas. (2014).

[8] Hiro G. C. Ferreira, Edna D. Canedo, and Rafael T. de Sousa Jr. 2013. IoT architecture to enable intercommunication through REST API and UPnP using IP, ZigBee and Arduino. In *2013 IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. 53–60. https://doi.org/10.1109/WiMOB.2013.6673340

[9] Hiro G. C. Ferreira, Edna D. Canedo, and Rafael T. de Sousa Jr. 2014. A ubiquitous communication architecture integrating transparent UPnP and REST APIs. *International Journal of Embedded Systems* 6, 2/3 (2014), 188. https://doi.org/10.1504/IJES.2014.063816

[10] Hiro G. C. Ferreira, Rafael T. de Sousa Jr., Flávio E. G. de Deus, and Edna D. Canedo. 2014. Proposal of a secure, deployable and transparent middleware for Internet of Things. In *2014 9th Iberian Conference on Information Systems and Technologies (CISTI)*. 1–4. https://doi.org/10.1109/CISTI.2014.6877069

[11] Maria Ganzha, Marcin Paprzycki, Wieslaw Pawlowski, Pawel Szmeja, and Katarzyna Wasielewska. 2016. Semantic Technologies for the IoT-An Inter-IoT Perspective. In *Internet-of-Things Design and Implementation (IoTDI), 2016 IEEE First International Conference on*. IEEE, 271–276.

[12] Aitor Gómez-Goiri, Pablo Orduña, Javier Diego, and Diego López-De-Ipiña. 2014. Otsopack: Lightweight semantic framework for interoperable ambient intelligence applications. *Computers in Human Behavior* 30 (2014), 460–467.

[13] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. 2013. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems* 29, 7 (2013), 1645–1660. https://doi.org/10.1016/j.future.2013.01.010

[14] Jaeho Kim and Jang-Won Lee. 2014. OpenIoT: An open service framework for the Internet of Things. In *Internet of Things (WF-IoT), 2014 IEEE World Forum on*. IEEE, 89–93. https://doi.org/10.1109/WF-IoT.2014.6803126

[15] Alexandr Krylovskiy, Marco Jahn, and Edoardo Patti. 2015. Designing a Smart City Internet of Things Platform with Microservice Architecture. IEEE, 25–30. https://doi.org/10.1109/FiCloud.2015.55

[16] Edoardo Patti and Andrea Acquaviva. 2016. IoT platform for Smart Cities: Requirements and implementation case studies. IEEE, 1–6. https://doi.org/10.1109/RTSI.2016.7740618

[17] Charith Perera, Arkady Zaslavsky, Peter Christen, and Dimitrios Georgakopoulos. 2014. Context Aware Computing for The Internet of Things: A Survey. *IEEE Communications Surveys Tutorials* 16, 1 (2014), 414–454. https://doi.org/10.1109/SURV.2013.042313.00197

[18] Akbar Rahman, Dorothy Gellert, and Dale Seed. 2011. A gateway architecture for interconnecting smart objects to the internet. In *Proceedings of the Workshop Interconnecting Smart Objects with the Internet, Prague, Czech Republic*, Vol. 25.

[19] João Santos, Bruno MC Silva, Joel JPC Rodrigues, João Casal, and Kashif Saleem. 2015. Internet of things mobile gateway services for intelligent personal assistants. In *E-health Networking, Application & Services (HealthCom), 2015 17th International Conference on*. IEEE, 311–316.

[20] Caio C. de M. Silva, Hiro G. C. Ferreira, Rafael T. de Sousa Jr., Fábio Buiati, and Luis J. García Villalba. 2016. Design and Evaluation of a Services Interface for the Internet of Things. *Wireless Personal Communications* (Jan. 2016). https://doi.org/10.1007/s11277-015-3168-6

[21] Caio C. M. Silva, Francisco L. de Caldas, Felipe D. Machado, Fábio L. L. Mendonça, and Rafael T. de Sousa Jr. 2016. Proposta de auto-registro de serviços pelos dispositivos em ambientes de IoT. Santarém-PA.

[22] Hannes Tschofenig and Jari Arkko. 2012. *Report from the smart object workshop*. Technical Report.

[23] Jean-Philippe Vasseur and Adam Dunkels. 2010. *Interconnecting smart objects with ip: The next internet*. Morgan Kaufmann.

[24] Guangyi Xiao, Jingzhi Guo, Li Da Xu, and Zhiguo Gong. 2014. User interoperability with heterogeneous IoT devices through transformation. *IEEE Transactions on Industrial Informatics* 10, 2 (2014), 1486–1496.

[25] Andrea Zanella, Nicola Bui, Angelo Castellani, Lorenzo Vangelista, and Michele Zorzi. 2014. Internet of Things for Smart Cities. *IEEE Internet of Things Journal* 1, 1 (Feb. 2014), 22–32. https://doi.org/10.1109/JIOT.2014.2306328

[26] Herwig Zeiner, Michael Goller, Víctor Juan Expósito Jiménez, Florian Salmhofer, and Werner Haas. 2016. SeCoS: Web of Things platform based on a microservices architecture and support of time-awareness. *e & i Elektrotechnik und Informationstechnik* 133, 3 (June 2016), 158–162. https://doi.org/10.1007/s00502-016-0404-z