KICS The Korean Institute of Communications and Information Sciences

Available online at www.sciencedirect.com

ScienceDirect

ICT Express ▮ (▮▮▮▮) ▮▮▮–▮▮▮

www.elsevier.com/locate/icte

Integration of Wi-Fi mobile nodes in a Web of Things Testbed

Luca Davoli*, Laura Belli, Antonio Cilfone, Gianluigi Ferrari

Wireless Ad-Hoc and Sensor Networks (WASN) Lab, Department of Information Engineering, University of Parma, Italy

Received 30 April 2016; accepted 21 July 2016

Abstract

The Internet of Things (IoT) is supposed to connect billions of devices to the Internet through IP-based communications. The main goal is to foster a rapid deployment of Web-enabled everyday objects, allowing end users to manage and control smart things in a simple way, by using Web browsers. This paper focuses on the integration of Wi-Fi nodes, hosting HTTP resources, into a Web of Things Testbed (WoTT). The main novelty of the proposed approach is that the WoTT integrates new nodes by using only standard mechanisms, allowing end-users to interact with all Smart Objects without worrying about protocol-specific details.

Keywords: Wi-Fi; Internet of Things; IoT; Web of Things; Constrained devices; IP camera

1. Introduction

The ubiquity of Wi-Fi technology in modern scenarios and its interoperability with widely used application protocols (e.g., HTTP) makes it a promising technology for the future. Wi-Fi guarantees large-area coverage, high bandwidth, robustness, and cost-effectiveness. An interesting paradigm that is gaining significant momentum is the Internet of Things (IoT), i.e., a technology scenario in which billions of devices can be connected to the Internet through IP-based communications. Therefore, the IoT is likely to face challenges similar to those historically faced by the Internet. The implementation of IoT systems has the double objective of lowering the entry barrier in connecting things to each other and to the Web, promoting a rapid deployment of Web-enabled objects and giving end-users simple methods by which to access things. To achieve this, one can leverage existing HTTP- and Wi-Fi-based infrastructures, following two different approaches for connecting things to the Web: (i) make a physical object smart by attaching to it an IoT node, then take advantage of its built-in sensors and actuators (e.g., a temperature sensor installed on a window); (ii) modify the physical object itself by electrically connecting an IoT node, thereby offering connectivity to actuators of the physical one (e.g., connect a lamp to a Wi-Fi node to create a smart lamp remotely manageable via HTTP). Moreover, research in the IoT field has been first focused on how to build IP-based architectures in Wireless Sensors Networks (WSNs), generally involving IP adaptations to enable resource-constrained things to be seamlessly connected to the Internet through lossy networks.

Wi-Fi technology adoption seems to be a good solution in some IoT scenarios (e.g., video monitoring). This observation motivates this work, which describes the integration of Wi-Fi nodes in an existing and already-deployed testbed, denoted as a Web of Things Testbed (WoTT) [1], composed of heterogeneous nodes hosting resources that can be accessed through the Constrained Application Protocol (CoAP) [2]. The integration of new Wi-Fi nodes hosting HTTP resources, using only standard mechanisms, dynamic nodes and resources discovery paradigms, represents the main novelty of the proposed approach, which allows end-users to easily interact with all Smart Objects (SOs), without worrying about specific protocol translation needs. The rest of the paper is organized as follows. In Section 2, an overview on related works on the integration of Wi-Fi-based WSNs in IoT scenarios is provided. In Section 3, all deployed modules of the proposed approach are detailed. Finally, in Section 4, some conclusions are drawn.

* Corresponding author.

E-mail addresses: luca.davoli@studenti.unipr.it (L. Davoli), laura.belli@unipr.it (L. Belli), antonio.cilfone@studenti.unipr.it (A. Cilfone), gianluigi.ferrari@unipr.it (G. Ferrari).

Peer review under responsibility of The Korean Institute of Communications Information Sciences.

http://dx.doi.org/10.1016/j.icte.2016.07.001

Please cite this article in press as: L. Davoli, et al., Integration of Wi-Fi mobile nodes in a Web of Things Testbed, ICT Express (2016), http://dx.doi.org/10.1016/j.icte.2016.07.001

2. Related works

In recent years, the theme of integrating Wi-Fi technology in IoT scenarios has been widely analyzed, often in a comparative way with respect to other existing RF techniques. In [3], the ZigBee protocol [4], widely used in WSNs, has been compared with Wi-Fi-based systems in IoT and Smart Grid scenarios. ZigBee is a low-power, low-rate and short-range technology that relies, at the bottom layers, on the IEEE 802.15.4 standard. The latter, however, suffers from some limitations, such as low data transmission rate. Wi-Fi-based WSNs, instead, provide some significant advantages: (i) high bandwidth, support for real-time and low-delay communications; (ii) large coverage (e.g., 50 m range for ZigBee nodes, 100 m indoor to 300 m outdoor for Wi-Fi technology); (iii) robustness, with quick installation and reliable fault-recovery; (iv) cost-effectiveness, as hardware cost is reduced using pre-existing Wi-Fi infrastructures. A comparison between IEEE 802.15.4 systems with a 6LoWPAN adaptation layer and low-power Wi-Fi is carried out in [5]. The recent development of power-efficient Wi-Fi components has the advantage of easy integration with existing infrastructures and built-in IP network compatibility, thus offering key cost savings and faster deployment. These features motivate the adoption of Wi-Fi technologies for all applications with real-time requirements, for which response time and reliability become critical. In [6], it is shown that, considering high OSI layers, Wi-Fi modules can achieve long battery lifetime, despite the use of HTTP over TCP/IP. For this reason, standards and paradigms already employed on the Web can be leveraged to manage resources of SOs, with the further advantage of no need for protocol translation. This fosters the creation of fully Web-enabled devices [7], through the adoption of Internet protocols for constrained devices (e.g., CoAP). Nevertheless, an approach in which an HTTP request obtains, as a response, an HTML document with a graphical representation of the resource, is heavy for both client and server, which must to parse and generate a complete HTML document, respectively. This evidence motivates the novel approach proposed in this work, in which a lighter paradigm for resource representation is used and an external entity for nodes and resource discovery is deployed. In [8], WebPlug is proposed to represent HTTP resources introducing some ontology-related concepts. However, this solution seems to be in conflict with known axioms of Uniform Resource Identifier (URI) representation and requires HTTP nodes to handle another REST-like paradigm, instead of reusing those adopted by CoRE WG [9].

3. Implementation

3.1. Architecture

The goal of the proposed architecture, whose main component modules are shown in Fig. 1, is to exploit the basic concept of the IoT, integrating Wi-Fi-based devices in the WoTT testbed, mainly composed of CoAP-enabled IEEE 802.15.4 nodes.

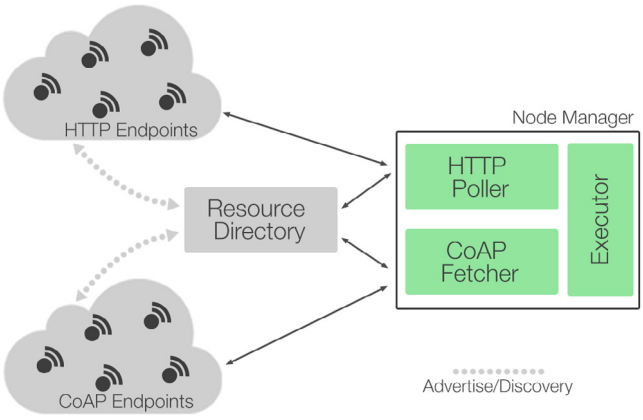

Fig. 1. Overview of the proposed architecture.

3.1.1. Resource Directory Module

The Resource Directory (RD) module [10] acts as an information repository for resources hosted by the endpoints in an IoT network. An endpoint is an IP-enabled entity associated with an address and a port: therefore, a physical node can host one or more endpoints, each one owning one or more resources. The RD is also able, in conjunction with a proxy module, to handle requests through different application protocols (e.g., HTTP and CoAP). In the proposed architecture, all endpoints advertise themselves through the JmDNS library [11], a Java-based implementation of the mDNS discovery protocol, regardless of the supported application protocol, so that resources hosted by endpoints are automatically added to the RD.

3.1.2. Node manager module

The Node Manager module is a Java-based entity managing the communication with all nodes integrated in the WoTT. It is composed of several sub-modules. (i) The HTTP Poller, implemented using the Jetty library, retrieves the list of available HTTP resources from the RD, looking for the desired ones accordingly with CoRE interface (*if*) and resource type (*rt*) attributes. Once the resource address is retrieved, the HTTP Poller starts the interaction with them. Because the CoAP-defined *observe* option is not natively supported by HTTP, the module must send periodic HTTP GET requests to the available HTTP resources. (ii) The CoAP Fetcher, implemented with the Californium (*Cf*) library, is responsible for interacting with CoAP-based endpoints. (iii) The Executor module processes the responses received by other modules (encoded in JSON format to allow a light data exchange) and decides the actions to be performed, if the responses satisfy some criteria (e.g., forwarding the obtained data to the Cloud [12] or to remote processing infrastructures [13,14], or making new requests to an actuator in the WoTT).

3.1.3. Wi-Fi nodes

Among many options, the Wi-Fi node integrated in the WoTT is the TI SimpleLink Wi-Fi CC3200 LaunchPad [15]. This board is equipped with two processors, which control the integrated Web server and the TCP/IP stack for networking operations, and a radio interface through the IEEE 802.11

Please cite this article in press as: L. Davoli, et al., Integration of Wi-Fi mobile nodes in a Web of Things Testbed, ICT Express (2016), http://dx.doi.org/10.1016/j.icte.2016.07.001

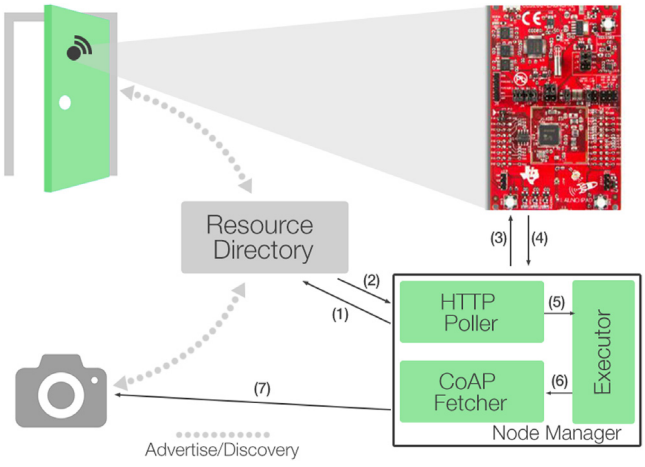

Fig. 2. IoT-based surveillance infrastructure.

b/g/n protocol. The board is also equipped with temperature and three-axis accelerometer sensors: sensed values can be extracted by means of internal prioritized tasks. The LaunchPad board has been chosen because it is one of the most complete devices available for IoT application development and can be battery powered. These features make it mobile, thus increasing the heterogeneity of the WoTT.

To smartly manage the new Wi-Fi node, the proposed networking architecture abides by the following principles: the internal representation of the resource list and the representation of the values of each sensor must be maintained on the LaunchPad board. The former aspect has been solved by "re-engineering" the board and enabling to create, at start-up time, a new HTTP resource reachable at */.well-known/core* URI: this represents the entry-point used for resource discovery and returns the list of all available HTTP resources in CoRE Link-Format fashion [16], in which each resource is described by its URI, *if* and *rt* CoRE attributes. Furthermore, the criticality related to the representation of the resource values is because, by default, the LaunchPad node responds to an HTTP GET request encapsulating the last retrieved value into an HTML page, which is difficult to read and parse. Considering this observation, in the proposed approach the measured value is written in a text file by using the JSON format. This solution is more readable, lightweight, and easily interpreted than that provided by TI.

3.2. Use case

The proposed architecture has been tested by implementing the Wi-Fi node-based IoT-based surveillance system shown in Fig. 2: a security camera takes a snapshot of a monitored environment if a sensor detects an unexpected presence.

The system is managed by the Node Manager module running on a Raspberry Pi 1 Model B, which starts querying the RD to discover the movement sensor and the surveillance camera. The former is handled by the accelerometer sensor on the TI LaunchPad Wi-Fi board, which is installed on the door in the supervised environment. The board starts advertising itself, to be discovered by the RD module. The camera entity is represented by another Raspberry Pi 1 Model B, already integrated in the WoTT testbed and equipped with a PiNoIR camera module (able to see in the dark with infrared lighting), running a built-in HTTP server (written in Python language) and a CoAP server (written with *Cf* library). Both of these servers allow the capture of pictures upon a request to specific URIs. First, the Node Manager discovers the movement resource on the Wi-Fi node, identified by the CoRE attributes *if=core.s* and *rt=accl*, and the surveillance camera, with *if=core.s* and *rt=camera*. The HTTP Poller then starts to periodically poll the LaunchPad board with HTTP GET requests, whose responses are finally handled by the Executor module. The current value received from the Wi-Fi node is then compared with two thresholds and, if a movement is detected (namely, the acceleration intensity is above a pre-set threshold), the Executor asks the CoAP Fetcher module to send a synchronous CoAP GET request to the Raspberry Pi board handling the surveillance camera, which captures a new snapshot (at a resolution of 320 × 240 px) and forwards it back to the Executor module. In Table 1, the experimentally measured times (in terms of average values and standard deviations), with reference to the different steps highlighted in Fig. 2, are shown.

Table 1
Measured times.

Operation	Average duration [s]	Standard deviation [s]
Intrusion sensor polling (steps 3–5)	0.91	0.24
Snapshot request (steps 6–7)	1.7	0.47

The choice to send a CoAP request, instead of an HTTP one, is to highlight the heterogeneity of the analyzed use-case and of the proposed architecture, in which there is no physical connection between the Wi-Fi board and the camera, because they interact only owing to the presence of the RD.

4. Conclusions

In this paper, we have introduced a novel approach to integrate Wi-Fi nodes into a Web of Things testbed (denoted as WoTT), in which the use of a Resource Directory module, endpoints and resource discovery mechanisms, in conjunction with a CoRE Link-Format resource representation, makes the proposed architecture scalable and lightweight, because HTTP and CoAP resources on heterogeneous devices are managed in a similar way. Future developments will involve the adoption of the Webhooks approach [17].

References

[1] L. Belli, S. Cirani, L. Davoli, A. Gorrieri, M. Mancin, M. Picone, G. Ferrari, Design and deployment of an IoT application-oriented testbed, Computer 48 (9) (2015) 32–40. http://dx.doi.org/10.1109/MC.2015.253.
[2] Z. Shelby, K. Hartke, C. Bormann, The Constrained Application Protocol, CoAP, RFC 7252 (Jun 2014).
[3] L. Li, H. Xiaoguang, C. Ke, H. Ketai, The applications of WiFi-based wireless sensor network in internet of things and smart grid, in: 2011 6th IEEE Conference on Industrial Electronics and Applications, ICIEA, 2011, pp. 789–793. http://dx.doi.org/10.1109/ICIEA.2011.5975693.
[4] S.C. Ergen, ZigBee/IEEE 802.15.4 Summary. http://pages.cs.wisc.edu/~suman/courses/707/papers/zigbee.pdf, 2016 (accessed 14.04.16).

Please cite this article in press as: L. Davoli, et al., Integration of Wi-Fi mobile nodes in a Web of Things Testbed, ICT Express (2016), http://dx.doi.org/10.1016/j.icte.2016.07.001

[5] S. Tozlu, Feasibility of Wi-Fi enabled sensors for internet of things, in: 2011 7th International Wireless Communications and Mobile Computing Conference, IWCMC, 2011, pp. 291–296. http://dx.doi.org/10.1109/IWCMC.2011.5982548.
[6] B. Ostermaier, M. Kovatsch, S. Santini, Connecting things to the web using programmable low-power WiFi modules, in: 2nd International Workshop on Web of Things, WoT'11, ACM, New York, NY, USA, 2011, pp. 2:1–2:6. http://dx.doi.org/10.1145/1993966.1993970.
[7] S. Tozlu, M. Senel, W. Mao, A. Keshavarzian, Wi-Fi enabled sensors for internet of things: A practical approach, IEEE Commun. Mag. 50 (6) (2012) 134–143. http://dx.doi.org/10.1109/MCOM.2012.6211498.
[8] B. Ostermaier, F. Schlup, K. Römer, WebPlug: A framework for the web of things, in: 2010 8th IEEE International Conference on Pervasive Computing and Communications Workshops, PERCOM Workshops, 2010, pp. 690–695. http://dx.doi.org/10.1109/PERCOMW.2010.5470522.
[9] IETF CoRE Working Group. https://tools.ietf.org/wg/core, 2016 (accessed 15.04.16).
[10] S. Cirani, L. Davoli, G. Ferrari, R. Léone, P. Medagliani, M. Picone, L. Veltri, A scalable and self-configuring architecture for service discovery in the internet of things, IEEE Internet Things J. 1 (5) (2014) 508–521. http://dx.doi.org/10.1109/JIOT.2014.2358296.
[11] JmDNS. http://jmdns.sourceforge.net, 2016 (accessed 16.04.16).
[12] L. Belli, S. Cirani, L. Davoli, L. Melegari, M. Mónton, M. Picone, An open-source cloud architecture for big stream IoT applications, in: Interoperability and Open-Source Solutions for the Internet of Things, Springer International Publishing, 2015, pp. 73–88. http://dx.doi.org/10.1007/978-3-319-16546-2_7.
[13] L. Belli, S. Cirani, L. Davoli, G. Ferrari, L. Melegari, M. Montón, M. Picone, A scalable big stream cloud architecture for the internet of things, Int. J. Syst. Serv.-Oriented Eng., IJSSOE 5 (4) (2015) 26–53. http://dx.doi.org/10.4018/IJSSOE.2015100102.
[14] L. Belli, S. Cirani, L. Davoli, G. Ferrari, L. Melegari, M. Picone, Applying security to a big stream cloud architecture for the internet of things, Int. J. Distrib. Syst. Technol., IJDST 7 (1) (2016) 37–58. http://dx.doi.org/10.4018/IJDST.2016010103.
[15] SimpleLink Wi-Fi CC3200 LaunchPad. http://www.ti.com/tool/cc3200-launchxl, 2016 (accessed 15.04.16).
[16] Z. Shelby, Constrained RESTful Environments (CoRE) Link Format, RFC 6690 (Aug 2012).
[17] Webhooks: building a more programmable web. http://www.webhooks.org/, 2016 (accessed 11.04.16).

Please cite this article in press as: L. Davoli, et al., Integration of Wi-Fi mobile nodes in a Web of Things Testbed, ICT Express (2016), http://dx.doi.org/10.1016/j.icte.2016.07.001