Journal of Ambient Intelligence and Humanized Computing (2024) 15:2757–2764
https://doi.org/10.1007/s12652-023-04552-6

ORIGINAL RESEARCH

A complete architecture for Ambient Assisted Living scenarios using a cross protocol proxy

Paola Pierleoni[1] · Alberto Belli[1] · Lorenzo Palma[1] · Roberto Concetti[1] · Luisiana Sabbatini[1] · Sara Raggiunto[1]

Received: 18 June 2020 / Accepted: 30 January 2023 / Published online: 8 February 2023

Abstract
Nowadays, in the most developed countries, modern society is moving towards scenarios in which the self-sufficiency elderly people live alone in their homes. An automatic remote monitoring system using wearable sensors is becoming even more important in Ambient Assisted Living applications. For this type of services, it is important that IoT sensors networks, which are generally composed of devices with limited computing power and storage, implement reliable communication among sensors and the Internet. There are several specialized protocols for the Internet of Things proposed by the scientific community, each characterized by its own levels of Quality of Services. The emergence of new protocols forces the need for developing proxying systems able to intermediate among different types of networks and to translate the relative protocols. In this paper, we propose a complete architecture for monitoring and managing wearable devices, and, in particular, fall detection ones. Our system uses a cross protocol proxy and a device with CoAP and MQTT as application level protocols, while it exploits the NB-IoT at physical and data-link levels. The goal of this work is the performance evaluation of the proposed solution in terms of Throughput, Round Trip Time and Delay. The results highlight the low latency reached by the proposed system architecture thanks to the implemented protocols.

Keywords Cross protocol proxy · CoAP · IoT · MQTT · Narrowband IoT · Wearable sensors

1 Introduction

Western countries are particularly sensible to the demographic aging process. Specifically, in Italy the increase in elderly population is even more pronounced. An increase in average life expectancy of more than 5 years is foreseen for both sexes by 2065: the life expectancy of Italian men and women (which was 80.6 and 85 years respectively in 2016) will reach 86.1 ± 2 years and 90.2 ± 3 years (Mediano and Al 2018). In this aging society scenario the relevance of using Information and Communication Technology (ICT) for elderly people's well-being is known. Nowadays, Ambient Assisted Living (AAL) is an extremely relevant theme being related to vital state monitoring, activity recognition, balance disorder management, aging supervision (Pierleoni et al. 2019a), and many other practices. In this regard, technologies should help in facing the multitude of daily life challenges. It is extremely important to evaluate, validate and integrate the existing AAL solutions with new ones. At the same time, attention should be paid to investigating and understanding of the connection between concrete users' needs and the proposed solutions (Calvaresi et al. 2017). The adoption of Cloud capabilities and Internet of Things (IoT) architectures can contribute substantially to improve and increase in number e-health services (Darwish et al. 2019), because elders' life quality can be strongly improved thanks to wearable sensors and ICT.

Through a flexible architecture, able to meet constraints like the small available energy, it is possible to inter-connect thousands of different devices to the Internet (Palattella et al. 2012; Rghioui et al. 2016). In this regard, an accurate protocol analysis is essential for the development of reliable architectures. During the last few years a light-protocol for IoT sensor networks has been developed, namely the Constrained Application Protocol (CoAP). In order to make a sensors network efficient, the direct interaction capability

✉ Sara Raggiunto
s.raggiunto@pm.univpm.it

Paola Pierleoni
p.pierleoni@univpm.it

[1] Department of Information Engineering (DII), Università Politecnica delle Marche, Via Brecce Bianche 12, 60131 Ancona, Italy

with every IoT application level protocol should be provided (Jin and Kim 2018; Atzori et al. 2010). Currently, among the application level protocols within the IoT domain, HTTP (HyperText Transfer Protocol), MQTT (Message Queuing Telemetry Transport), and CoAP (Constrained Application Protocol) are the mostly employed. The adoption of a cross protocol proxy, capable of connecting different application level protocols, is essential to successfully integrate a CoAP-based sensor network with the HTTP and MQTT protocols.

In this work, a complete architecture for monitoring and managing generic IoT devices is presented. Part of this architecture are any IoT device implementing the CoAP protocol and a cross protocol proxy able to connect the CoAP device with the HTTP, CoAP or MQTT protocols.

Different papers in literature present a qualitative analysis of the messaging protocols MQTT, CoAP, and HTTP for IoT systems without providing any experimental performance evaluation (Liu et al. 2020; Glaroudis et al. 2020; Naik 2017; van der Westhuizen and Hancke 2018). Philip et al. (2021) reviewed the use of IoT in home systems and applications for health monitoring. They presented the latest advances of the protocol stacks like our architecture in order to gain a better understanding of their strengths and limitations. However, the evaluation of performance metrics is fundamental to provide new insights about the effectiveness of different IoT protocols. Only a few works illustrate the efficiency of architecture in terms of performance metrics, such as Throughput, Round Trip Time (RTT), and latency (Moraes et al. 2019; Seoane et al. 2021; Laaroussi and Novo 2021). Zahran et al. (2019) proposed an IoT system for health applications and their results showed the effectiveness of the proposed system in minimizing the IoT network power consumption. Moreover, system performance is represented by the Throughput which illustrates the rate of successful message delivery over a communication channel. Finally, system efficiency is evaluated to show the effect of alarm delay on the patients' life. These studies take into account the RTT, Throughput and delay, but have not thoroughly evaluated the performance metrics as a function of the number of concurrent Clients. For example Hamdani and Sbeyti (2019) compare the MQTT and CoAP performance metrics through the implementation of only two Client applications. However, the performance metrics as a function of the number of concurrent Clients is essential in order to better understand the proposed architecture strengths and limitations. In this perspective, the presented work aims to fill this gap and provide the performance of these widely used IoT protocols taking into account an increasing number of concurrent Clients. The proposed architecture has been tested both in a simulated environment and a real AAL scenario in which the interaction with a fall detection device has been evaluated.

Aging is characterized by the progressive and gradual deterioration of functional mobility which can lead to falls. These critical events are a widespread problem among the elderly and the people living independently. In fact, falls can cause physical and psychological effects, particularly among older people, and therefore require timely rescue. The intent of the test inside the AAL scenario considered in this paper is to use the proxy for managing the fall detection device, with the requirement to timely report every fall. For what it concerns the protocol choice, the fall detector implements LPWAN Narrowband IoT (NB-IoT) protocol on a dedicated Telecom Italia Mobile (TIM) network, MQTT for sending alarms, and CoAP for exchanging information between the proxy. In the test inside the AAL scenario, the latency of the alarm transmission has been analyzed in order to examine the behavior of delays during communication, and to detect the cause of any delay higher than the system requirements defined by the Third Generation Partnership Project (3GPP).

The paper is organized as follows. Section 2 introduces the proposed architecture and the hardware of the wireless wearable device for fall detection. Section 3 describes the experimental setup used in the simulated environment and in the real AAL scenario. The obtained results are discussed in the Sect. 4. Conclusions end the paper in the last Sect. 5.

2 Materials and methods

In this work, we propose a complete architecture for monitoring and managing generic devices that are part of the growing world of the IoT. As shown in Fig. 1, the proposed architecture is based on a cross protocol proxy allowing data exchange with a fall detection wearable device. Our device exchanges information through the CoAP protocol, implementing both a CoAP Client and Server. The CoAP Client is used to store data on remote servers, while the CoAP Server is used to allow real-time data requests. The cross protocol proxy is essential for making these data suitable to be used from the majority of Servers and Clients in the Internet. Using this proxy, HTTP Clients can set configuration

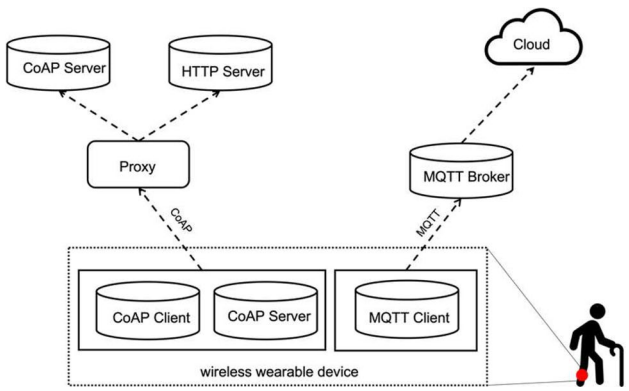

Fig. 1 Developed system schematic representation

parameters and read in real-time data acquired by the sensing units embedded into the device. Moreover, the fall detection device can upload data to a remote database reachable via the HTTP, CoAP, and MQTT protocols. However, the fall detection application requires transmission reliability, and for this purpose MQTT over Transmission Control Protocol (TCP) is the best possible solution, as it is mandatory to achieve very low packet loss rate. Accordingly, MQTT is used for the alert transmission, which is a task requiring reliable reception with low latency as QoS (Quality of Service) fundamental parameters.

2.1 Cross protocol proxy

CoAP is designed to meet specific requirements, such as simplicity and low overhead, in Wireless Sensor Network characterized by limited computational power and storage capability. For this reason a cross protocol proxy is necessary for communication between AAL systems and the Internet. The developed cross protocol proxy (Pierleoni et al. 2019b) is able to receive requests from an HTTP or CoAP Client, convert them into the receiving Server's protocol (HTTP, CoAP or MQTT), and vice versa, convert back the answer to forward it to the source Client. Moreover, the caching function allows to store the answer to a request to speed up the processing of similar requests in the future, thus reducing the traffic and improving the overall response time of the network. In the cross protocol proxy, four types of proxying, related to the interaction of the CoAP protocol with HTTP and MQTT, have been developed and tested (Pierleoni et al. 2019b). The first type is the CoAP-HTTP proxying, where the proxy forwards and translates the CoAP Client's request to the HTTP Server and vice versa. The second type is the CoAP-CoAP proxying. In this case, the proxy is necessary if the CoAP Client has not the required privileges, so the proxy forwards the request to the CoAP Server and gives the answer back to the Client. The third type is the HTTP-CoAP proxying. The proxy forwards the HTTP Client's request to the CoAP Server and maps the CoAP response into an equivalent one that must be sent to the Client. Finally, the CoAP-MQTT proxying is necessary whenever a CoAP Client needs the access to an MQTT broker. Specifically, the proxy as MQTT Client is able to mediate between the CoAP Client and the MQTT Server containing the needed resource.

The Eclipse Californium framework (Cf) has been adopted (Californium 2019) for CoAP-HTTP, CoAP-CoAP, and HTTP-CoAP proxying types. It is a Java library implementing the CoAP protocol. Compliant with the Standard IETF RFC-8075, it allows to convert the HTTP requests into the CoAP protocol-based requests, and vice versa (Castellani et al. 2017). For what it concerns the CoAP-MQTT proxying, the Paho Java Client library has been used. In this proxy, it has been necessary to add an MQTT Client, and the imported library provides APIs for the creation of Clients able to communicate with the MQTT Servers, in synchronous and asynchronous way. In this study, an asynchronous MQTT Client has been implemented to ensure the processing of incoming requests also during the messages exchange with the broker.

2.2 Fall detection device

The architecture proposed in this paper integrates a wireless wearable device for fall detection (Pierleoni et al. 2014). The device can be attached to the ankle or integrated into an elderly's shoe. Compared to the previous work, in this paper we aim at testing the transmission quality using IoT solutions for Internet communications. In this new scenario, information transmission through Bluetooth connection by a smartphone, has been replaced with the nRF9160 Development Kit. It is a low power highly-integrated System-in-Package (SiP) integrating a NB-IoT modem, a GPS, and an Arm Cortex-M33 application processor. The device is equipped with a 3-axis accelerometer, a 3-axis gyroscope and a 3-axis magnetometer, realizing an Attitude Heading Reference System (AHRS). The AHRS provides the correct 3D orientation through an implemented data fusion algorithm (Pierleoni et al. 2014). The sensors signals are transmitted via I2C bus to the processor of the nRF9160 Development Kit, that processes them to derive the orientation measurement through the implemented orientation filter. The block diagram of the hardware is shown in Fig. 2.

On the nRF9160 Development Kit was also implemented the walk analysis algorithm and the automatic fall detection algorithm (Pierleoni et al. 2014). The former, is able to handle the three main parameters useful for the mobility assessment, obtained from the orientation data furnished by the AHRS: double support, stride speed, and stride length. These parameters are uploaded into the Cloud via CoAP.

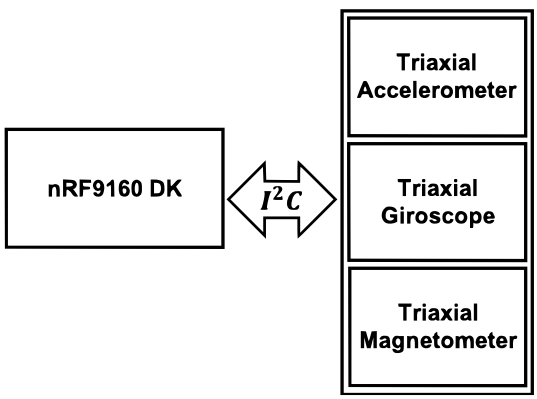

Fig. 2 Block diagram of wireless wearable device

The Cloud application processes them and computes the fall detection threshold to be set into the device. Instead, the automatic fall detection algorithm generates alarms in cases of critical events via MQTT over the NB-IoT service. The entire protocol stack implemented into the device is shown in Fig. 3.

3 Experimental setups

In this paper two experimental configurations have been proposed to validate the developed proxy. At first, the implemented cross protocol proxy has been tested under four operating conditions in simulated environments. Subsequently, a real AAL scenario has been set up to evaluate the interaction with a fall detection device.

3.1 The simulated environments setup

The simulated environments setup are based on the adoption of a single machine implementing both the developed proxy and the software employed in the tests to simulate concurrent Client requests. Compared to our previous work (Pierleoni et al. 2019b), we aim at testing the cross protocol proxy using a much better performing machine. The main characteristics of the used machine are: Intel Xeon X5650 (x2) CPU, 12 MB cache, 2.66 GHz, 16 GB RAM, and Ubuntu 18.04.1 LTS as operating system.

Simultaneous submission of requests from many competing Clients to the proxy have been tested in the experimental setup. The performances evaluation of the four different types of proxying is based on two main parameters namely Throughput (i.e. the number of requests processed per second) and Round Trip Time (RTT, i.e. the time for answering computed as the difference between the request submission from the Client and the answer receipt to the Client). Moreover, the four proxying types have been monitored with and without the caching function. Multiple tests have been made for each scenario to extract an average behavior of the two performance metrics taken into account.

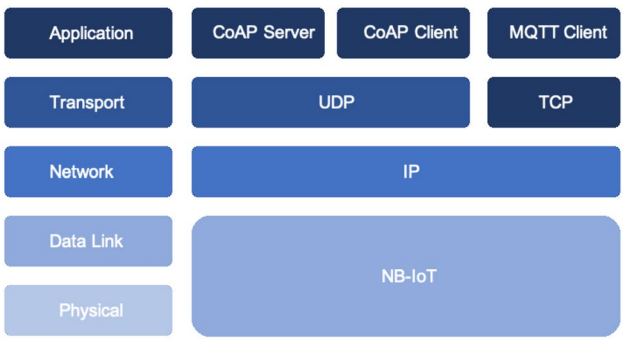

Fig. 3 The device protocol stack

3.2 The setup of the real AAL scenario

In this paper, we have also presented a real world use case in the AAL scenario, specifically the delay measurement of the alarm sent by the device subsequent to a fall detection.

Since the performance of MQTT over TCP is well known when the system is not overloaded, and the trend of latency with increasing number of devices (Larmo et al. 2019), we have decided to implement into the wireless wearable device an MQTT Client only for the transmission of the fall detection alert, and a CoAP Client for data storage and device management. The device was registered to the TIM network by creating a PDN (Packet Data Network) Context, negotiating connection parameters, and requiring information about the DNS (Domain Name System) Server through PCO (Protocol Configuration Option). Subsequently, once defined the IP address of the MQTT broker, a connection to the broker through TCP was realized. Whenever an alarm signaling from the device to the broker occurred, one packet containing the identification number and the Timestamp of the moment of transmission (UTC time, Coordinated Universal Time, in milliseconds) is published in a special topic. The broker forwards the content received from the device to the Cloud where an MQTT Client is subscribed to the same topic. It saves its own Timestamp of the moment when the message is received in order to take trace of the packet arrival time. These time recordings allow to compute the duration of the packet's travel inside the different components of the system. Specifically, we are interested in quantifying the latency in the connection from the device to the broker, i.e. through the NB-IoT service. During the experiment, around 800 packets with a fixed dimension of 21 bytes have been sent for each iteration, with an interval of 10 s between a packet and the next one. The time between a transmission and the subsequent is irrelevant for our test purposes.

4 Results

The results of the simulated environments are illustrated in Figs. 4 and 5. In Fig. 4a the Throughput trend is shown without cache function. The HTTP-CoAP, CoAP-CoAP, and CoAP-HTTP proxying are not linear as a function of the number of Clients. Very good performance is reached in the CoAP-HTTP scenario. Finally, in the CoAP-MQTT proxying, the Throughput drastically drops down to hundred requests per second when using level 1 of QoS. In Fig. 4b the same tests have been performed with the cache function on. It is noticeable that the adoption of the

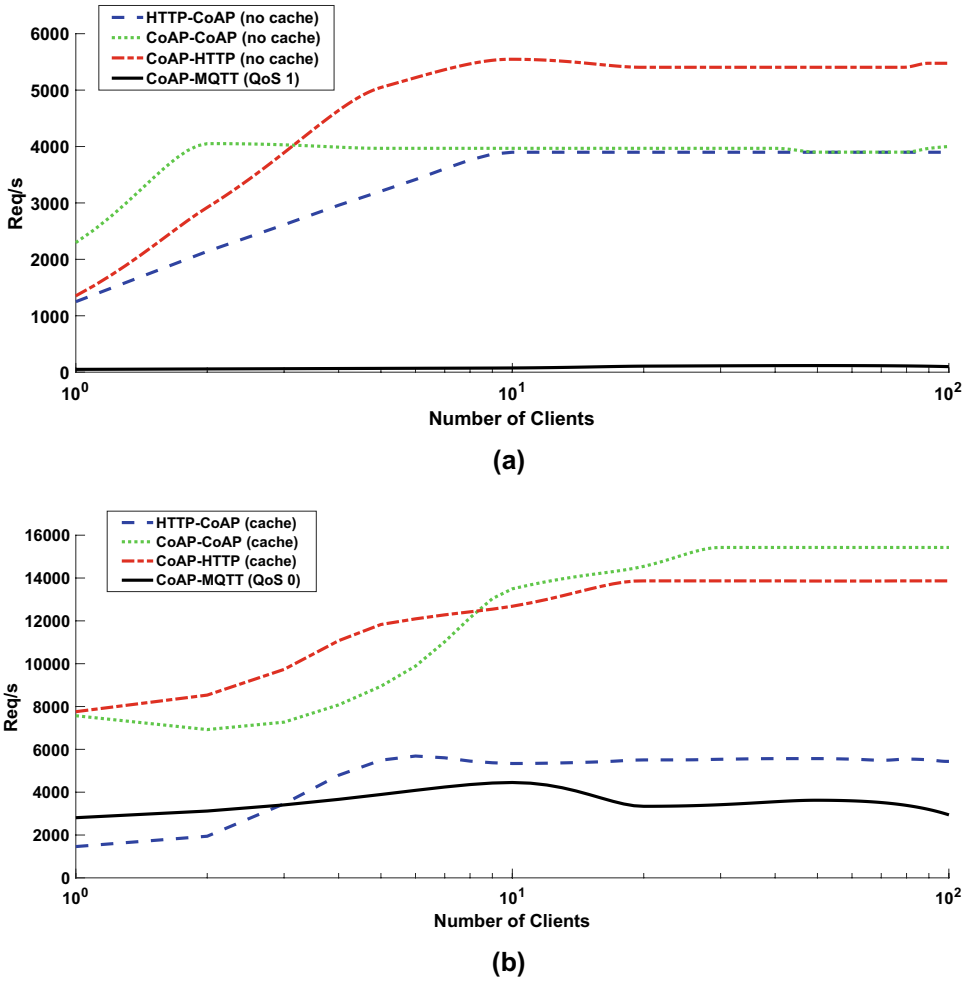

Fig. 4 Trends of the average Throughput as a function of the number of Clients in the four proxying scenarios: **a** with caching function off and level 1 of QoS for MQTT; **b** with caching function on and level 0 of QoS for MQTT

caching improves the performances for what it concerns the Throughput in all types of proxying. Specifically, in the HTTP-CoAP proxying the cache function allows to serve about 6000 requests per second in the configuration of 40 concurrent Clients. It strongly improves the performances respect to the tests without cache function. In the CoAP-CoAP proxying the cache function allows to serve about 15,000 requests per second in the configuration of 40 concurrent Clients. Finally, in the CoAP-HTTP proxying it allows to reach the capacity of serving up to 13,000 requests with 40 concurrent Clients. Instead, in the CoAP-MQTT proxying, despite the Throughput is much lower then that obtained in the previous testing scenarios based on the CoAP protocol, the performances are improved using level 0 of QoS.

In Fig. 5 the RTT trends for the four proxying types are shown. In all scenarios the response times have a linear trend as a function of the number of concurrent Clients. From the figure it is evident how much the response times have decreased significantly with the caching function (Fig. 5b). Moreover, in the CoAP-MQTT proxying the RTT is very high when using the level 1 of QoS (Fig. 5a).

The results obtained in the simulated environments confirm the theoretical broad assessment presented in literature (Philip et al. 2021; Liu et al. 2020; Glaroudis et al. 2020; Naik 2017; van der Westhuizen and Hancke 2018) and are coherent with the studies evaluating the performance metrics of IoT protocols (Moraes et al. 2019; Zahran et al. 2019). Moreover, the results show that the best performance are obtained with the caching function enabled for CoAP, and the 0 level of QoS for MQTT. These conclusions are in line with what Laaroussi and Novo (2021) described.

The goal of the test in the real AAL scenario has been to quantify the effect of latency on the system. The system performances have been evaluated taking into account the trend of delays. The results of several tests sending the same number of packets are shown in Fig. 6. The average behavior of the latency presents occasional peaks of delay which can reach up to 10 s in the route through the NB-IoT network. With the aim of determining the reason causing these peaks, we analysed the Acknowledgement (ACK) that the board receives from the broker whenever a sent package is correctly received at destination. In fact, we adopted QoS equal to 1 for MQTT. Therefore, the broker should send

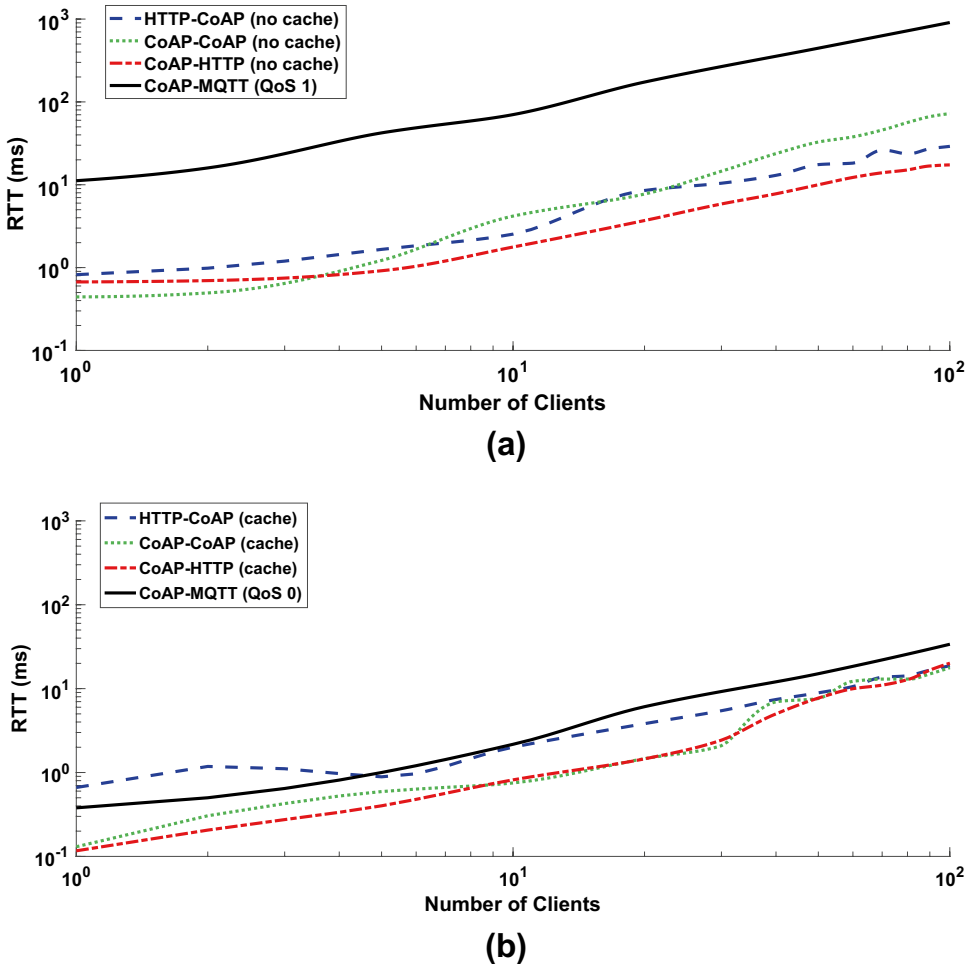

Fig. 5 Trends of the average Round Trip Time as a function of the number of Clients in the four proxying scenarios: **a** with caching function off and level 1 of QoS for MQTT; **b** with caching function on and level 0 of QoS for MQTT

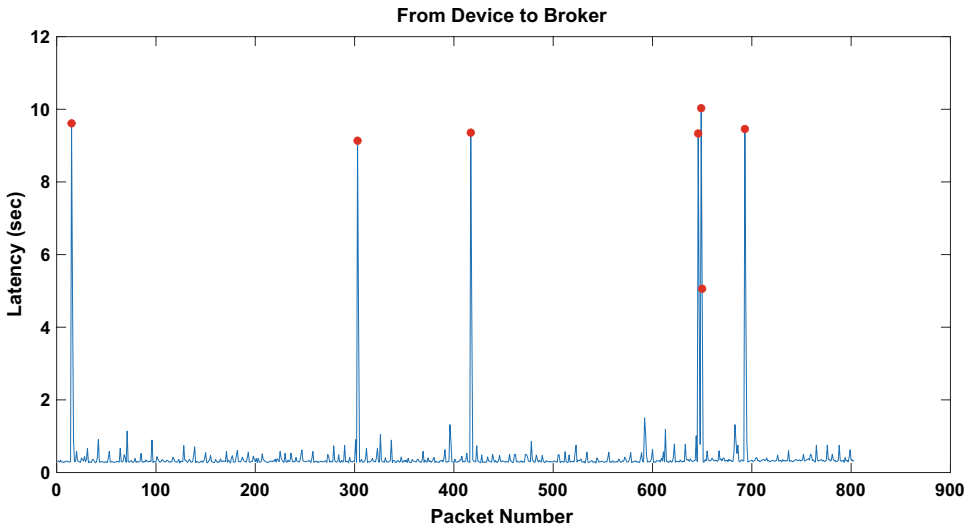

Fig. 6 Latency measured for the transmission of 800 packets each one of 21 bytes. The red points highlight the major delays

ACK messages and the board waits for them, to be sure the transmission was successful.

Packets detection is carried out by monitoring the number of ACKs and looking for those whose difference between the ACK number and the packet number is higher than zero. In case of packet loss there is a mismatch between the number of ACK and the packet number, in this case, in order to maintain the same distance between the ACK and the packet

number, a realignment of the sequence number of the following ACKs is necessary. Whenever the ACK and the packet numbers disagree, the latency value of previous packet is set equal to the average computed in a neighborhood of each packet. As can be noted in Fig. 7, the peaks disappear after this analysis, showing that the above mentioned points are the reason for the sudden delay raising. In Fig. 7 the behavior of latency, neglecting the delay related to the aforementioned packets, is shown. The loss of an ACK can be attributed to a packet's loss by the network, or to the simple ACK loss. In this latter case, the board interprets the ACK loss in the same way as if the packet has never been received by the broker. The receiving confirmation and the queue creation mechanisms allow to keep the integrity of messages sent in MQTT. It is evident how much this translates into higher delays. In fact, those peaks do not allow us to reach latencies in the order of milliseconds, but in a system implementing NB-IoT we cannot expect such short delays. After iterating the same experiment several times using the same number of packets and dimension (21 bytes), latency still remains under 2 s, with a mean of about 0.4 s. The packet loss rate is negligible, being a little higher than 0.1%. Moreover, these lost packets are recovered through re-transmissions. The adoption of MQTT over TCP provides guarantees at QoS level and packets monitoring. As previously showed in the test, the QoS equal to 1 causes additional delays due to re-transmissions. In a system such as the CoAP one, any packet loss would have been ignored both from the Client and the board, without the creation of any transmission queue.

We can deduce from the experiment that the adoption of QoS level 1 is still able to provide good performances. The delays associated with the message sending are limited and the device can be considered well served. Therefore, the proposed system is efficient enough for being used to send alerts in case of fall detection in an AAL scenario.

5 Conclusions

In this paper a complete architecture for monitoring and managing a wireless wearable sensor for fall detection has been presented. The developed system has been tested and validated through two experimental setups. The results of simulated environment tests show that the integration of two macro-solutions of compatibility, HTTP-CoAP and CoAP-MQTT, definitely enlarges the application contexts of the solution inside the IoT scenario. The adoption of a cross protocol proxy is essential for implementing a scenario composed of different IoT devices able to monitor the state of an older person, such as his mobility and the fall detection.

The device, realized by the nRF9160 Development Kit provided with sensors such as accelerometer, gyroscope, and magnetometer, has been tested implementing Client's and Server's requests, therefore evaluating the latency in the transmission of the alarm signal in occurrence of fall. The tests in a real AAL scenario show how the latency perfectly fits the tolerable limits. Specifically, MQTT over TCP allows to keep good performance, even though a QoS equal to 1 necessarily causes an occasional increase of delays. Despite this, the efficiency of the application is appropriate for the AAL scenario. The development of a low-cost wearable sensor together the wide scalability of the network, make the Narrowband IoT a promising solution regarding the broad Internet of Things domain. In detail, as shown in the results, the limited transmission speed, the reduced latency, and the limited size of the messages, allow the protocol to find wide adoption for AAL applications, as that we have proposed. Our system proved to be very effective in AAL environment because capable of inter-operating with the IoT and its main protocols. In the AAL context the inter-operability of a solution is a fundamental requirement, and the adoption of our proposal aims at reducing the technical complexity avoiding

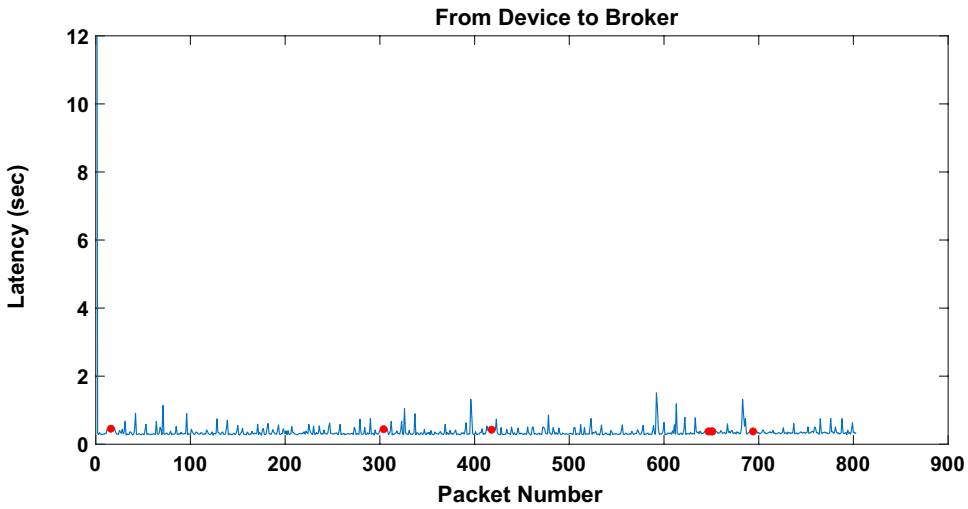

Fig. 7 Latency evaluated by ignoring the packets for which the acknowledgment message has not been received (red points)

the management of advanced communication protocols for the IoT.

Acknowledgements This work is supported by the Marche Region through the financial programme POR MARCHE FESR 2014-2020, project "Miracle" (Marche Innovation and Research fAcilities for Connected and sustainable Living Environments), CUP B28I19000330007.

Funding Open access funding provided by Università Politecnica delle Marche within the CRUI-CARE Agreement.

References

Atzori L, Iera A, Morabito G (2010) The internet of things: a survey. Comput Netw 54(15):2787–2805. https://doi.org/10.1016/j.comnet.2010.05.010

Californium (2019) Californium coap framework. https://www.eclipse.org/californium/. Accessed 18 Feb 2019

Calvaresi D, Cesarini D, Sernani P, Marinoni M, Dragoni AF, Sturm A (2017) Exploring the ambient assisted living domain: a systematic review. J Ambient Intell Humaniz Comput 8(2):239–257. https://doi.org/10.1007/s12652-016-0374-3

Castellani A, Loreto S, Rahman A, Fossati T, Dijk E (2017) Guidelines for mapping implementations: http to the constrained application protocol (coap). Internet Engineering Task Force (IETF), Fremont, pp 1721–2070. https://doi.org/10.17487/RFC8075

Darwish A, Hassanien AE, Elhoseny M, Sangaiah AK, Muhammad K (2019) The impact of the hybrid platform of internet of things and cloud computing on healthcare systems: opportunities, challenges, and open problems. J Ambient Intell Humaniz Comput 10(10):4151–4166. https://doi.org/10.1007/s12652-017-0659-1

Glaroudis D, Iossifides A, Chatzimisios P (2020) Survey, comparison and research challenges of iot application protocols for smart farming. Comput Netw 168:107037. https://doi.org/10.1016/j.comnet.2019.107037

Hamdani S, Sbeyti H (2019) A comparative study of coap and mqtt communication protocols. In: 2019 7th International Symposium on digital forensics and security (ISDFS), IEEE, pp 1–5, https://doi.org/10.1109/ISDFS.2019.8757486

Jin W, Kim D (2018) Development of virtual resource based iot proxy for bridging heterogeneous web services in iot networks. Sensors 18(6):1721. https://doi.org/10.3390/s18061721

Laaroussi Z, Novo O (2021) A performance analysis of the security communication in coap and mqtt. In: 2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC), IEEE, pp 1–6, https://doi.org/10.1109/CCNC49032.2021.9369565

Larmo A, Ratilainen A, Saarinen J (2019) Impact of coap and mqtt on nb-iot system performance. Sensors 19(1):7. https://doi.org/10.3390/s19010007

Liu X, Zhang T, Hu N, Zhang P, Zhang Y (2020) The method of internet of things access and network communication based on mqtt. Comput Commun 153:169–176. https://doi.org/10.1016/j.comcom.2020.01.044

Mediano PRIIS, Al EIDC (2018) Il futuro demografico del paese. Centro 12:1

Moraes T, Nogueira B, Lira V, Tavares E (2019) Performance comparison of iot communication protocols. In: 2019 IEEE International Conference on systems, man and cybernetics (SMC), IEEE, pp 3249–3254, https://doi.org/10.1109/SMC.2019.8914552

Naik N (2017) Choice of effective messaging protocols for iot systems: Mqtt, coap, amqp and http. In: 2017 IEEE International Systems Engineering Symposium (ISSE), IEEE, pp 1–7, https://doi.org/10.1109/SysEng.2017.8088251

Palattella MR, Accettura N, Vilajosana X, Watteyne T, Grieco LA, Boggia G, Dohler M (2012) Standardized protocol stack for the internet of (important) things. IEEE Commun Surv Tutor 15(3):1389–1406. https://doi.org/10.1109/SURV.2012.111412.00158

Philip NY, Rodrigues JJ, Wang H, Fong SJ, Chen J (2021) Internet of things for in-home health monitoring systems: current advances, challenges and future directions. IEEE J Sel Areas Commun 39(2):300–310. https://doi.org/10.1109/JSAC.2020.3042421

Pierleoni P, Belli A, Palma L, Pernini L, Valenti S (2014) A versatile ankle-mounted fall detection device based on attitude heading systems. In: Biomedical Circuits and Systems Conference (BioCAS), 2014 IEEE, IEEE, pp 153–156, https://doi.org/10.1109/biocas.2014.6981668

Pierleoni P, Belli A, Concetti R, Palma L, Pinti F, Raggiunto S, Sabbatini L, Valenti S, Monteriù A (2019a) Biological age estimation using an ehealth system based on wearable sensors. J Ambient Intell Humaniz Comput. https://doi.org/10.1007/s12652-019-01593-8

Pierleoni P, Belli A, Palma L, Incipini L, Raggiunto S, Mercuri M, Concetti R, Sabbatini L (2019b) A cross-protocol proxy for sensor networks based on coap. In: 2019 IEEE 23rd International Symposium on consumer technologies (ISCT), IEEE, pp 251–255, https://doi.org/10.1109/ISCE.2019.8900987

Rghioui A, Sendra S, Lloret J, Oumnad A (2016) Internet of things for measuring human activities in ambient assisted living and e-health. Netw Protoc Algorithms 8(3):15–28. https://doi.org/10.5296/npa.v8i3.10146

Seoane V, Garcia-Rubio C, Almenares F, Campo C (2021) Performance evaluation of coap and mqtt with security support for iot environments. Comput Netw 197:108338. https://doi.org/10.1016/j.comnet.2021.108338

van der Westhuizen HW, Hancke GP (2018) Practical comparison between coap and mqtt-sensor to server level. In: 2018 Wireless Advanced (WiAd), IEEE, pp 1–6, https://doi.org/10.1109/WIAD.2018.8588443

Zahran F, Hamada AO, Azab M (2019) Cooperative heterogeneous iot for health. In: 2019 IEEE 9th Symposium on computer applications & industrial electronics (ISCAIE), IEEE, pp 352–357, https://doi.org/10.1109/ISCAIE.2019.8743973