IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 28 March 2023, accepted 26 April 2023, date of publication 1 May 2023, date of current version 8 May 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3271773

RESEARCH ARTICLE

Internet of Things for Earthquake Early Warning Systems: A Performance Comparison Between Communication Protocols

PAOLA PIERLEONI[1], ROBERTO CONCETTI[1,2], SIMONE MARZORATI[3], ALBERTO BELLI[1], AND LORENZO PALMA[1], (Member, IEEE)

[1]Department of Information Engineering (DII), Università Politecnica delle Marche, 60131 Ancona, Italy
[2]Istituto di Istruzione Superiore Carlo Urbani, 63821 Porto Sant'Elpidio, Italy
[3]Centro Nazionale Terremoti, Istituto Nazionale di Geofisica e Vulcanologia (INGV), 60131 Ancona, Italy

Corresponding author: Alberto Belli (a.belli@univpm.it)

This work was supported in part by INGV Projects "Rete Multiparametrica" (art. 1 comma 1110, Legge di Bilancio 2018, piano straordinario) under Grant CUP D53C18000150001, and in part by "Ricostruzione Centro Italia" (D.L.50/2017) under Grant D53C21000140001.

ABSTRACT Earthquake Early Warning Systems (EEWSs) characterize seismic events in real time and estimate the expected ground motion amplitude in specific areas to send alerts before the destructive waves arrive. Together with the reliability of the results, the rapidity with which an EEWS can detect an earthquake becomes a focal point for developing efficient seismic node networks. Internet of Things (IoT) architectures can be used in EEWSs to expand a seismic network and acquire data even from low-cost seismic nodes. However, the latency and the total alert time introduced by the adopted communication protocols should be carefully evaluated. This study proposes an IoT solution based on the message queue-telemetry transport protocol for the waveform transmission acquired by seismic nodes and presents a performance comparison between it and the most widely used standard in current EEWSs. The comparison was performed in evaluation tests where different seismic networks were simulated using a dataset of real earthquakes. This study analyzes the phases preceding the earthquake detection, showing how the proposed solution detects the same events of traditional EEWSs with a total alert time of approximately 1.6 seconds lower.

INDEX TERMS Earthquake early warning systems, Internet of Things, message queue telemetry transport protocol, SeedLink protocol, miniSEED packet.

I. INTRODUCTION

Victims, injuries, and economic losses are only some catastrophic consequences of earthquakes. Natural disasters are impossible to predict [1], but real-time seismic monitoring is possible [2] using modern technologies, such as telecommunications and sensor networks, and edge and cloud computing. These technologies allow the development of integrated architectures of hazard monitoring, disaster risk assessment, communication and preparedness activities, and processes enabling individuals, communities, governments, businesses, and others to take timely action to reduce disaster risks before hazardous events [3]. These architectures are known as Earthquake Early Warning Systems (EEWSs) [4], [5]. Their primary goal is to provide information as quickly as possible about the location, size, and detection time of the earthquake to support emergency response effectively. Cooper [6] proposed an EEWS based on the greater speed with which information can travel through an electromagnetic signal compared to the propagation of seismic waves. Therefore, when a system close to a seismic zone detects an event, an alarm can be sent to one or more distant targets. When an earthquake occurs, the originated seismic waves are body waves and surface waves. Each wave type has its characteristic speed and destructive potential. Body waves comprise primary waves (P-waves) and secondary waves (S-waves). S-waves are slower than P-waves and carry most of the energy. The different characteristics of these waves allow

The associate editor coordinating the review of this manuscript and approving it for publication was Hayder Al-Hraishawi.

detecting P-waves to promptly alert people in specific places before the more destructive S-waves reach them. The delay between the arrival times of P-waves and S-waves allows EEWSs to generate and disseminate timely warnings. Based on the approach used to generate an alarm, EEWSs can be grouped into on-site and regional systems [7]. The on-site system should detect an earthquake and predict the peak shaking at the same location where the seismographic data are recorded. In this case, suitable algorithms are implemented to detect the P-waves using, for example, the scaling relation between the peak initial displacement amplitude and the peak ground motion displacement and velocity [8]. In a regional system (also known as a network-based system), a network of seismic stations is placed near an epicentral area. A processing hub processes the collected data using algorithms that detect, locate, and characterize an earthquake and sends alarms to distant targets. However, a common approach is combining the on-site and regional systems [9]. Such EEWSs analyze an early portion of the P-wave to forecast the S-wave and warn whether the shaking could be higher than a threshold. The critical parameter for any EEWS is the lead time. In an on-site system, the lead time is equal to the difference between the arrival time of the P-wave and S-wave where the device is situated, whereas, in a regional system, it is the difference between the P-wave arrival time at the source and the S-wave arrival time at the target.

The elapsed time between the origin of an earthquake and the moment of the first issued alarm and the accuracy of the estimated parameters are the critical performance metrics of an EEWS. Each EEWS should deliver consistent alarms in the fastest way, with an obvious trade-off between these metrics. Allen et al. [10] reviewed the history and the status of EEWSs globally, and Clinton et al. [11] presented the fundamental recent developments in this research field in Europe. The literature highlighted that many studies evaluated the performance of different algorithms or complete systems for early earthquake warnings. The performance evaluation of the Japanese EEWS is presented in [12]. Bose et al. [13] presented an algorithm for on-site early earthquake warnings in California, and Kuyuk et al. [14] assessed the performance of three network-based systems in the Marmara region (Turkey). Chung et al. [15] proposed the performance improvement of the ElarmS algorithm from the 1.0 version [16] to the current 3.0 version. The ElarmS 1.0v delay times were also detailed in [17], where the system was assessed in real-time [18], [19] in California and offline with a dataset for Japanese earthquakes. Behr et al. [20] proposed a complete methodology to study the contributions of delay of each EEWS component [21]. However, its applicability to actual Internet of Things (IoT) solutions [22] has yet to be evaluated. IoT seismic nodes can be used in an EEWS to provide additional tools to sense, process, and analyze environmental data [23]. In the last decade, devices, sensors, and actuators connected to the virtual world of the Internet have become widespread to generate added value in a growing number of application fields such as smart cities [24], [25], structural health monitoring [26], [27], and environmental monitoring [28], [29]. Mei et al. [30] presented a survey of IoT research and technological developments applied in geohazard monitoring and prevention, including EEWSs. In [31] and [32], the authors presented IoT solutions for detecting shaking caused by an earthquake and sending warning messages. The National Taiwan University developed a network of 581 low-cost accelerometers to provide on-site warnings using P-wave displacement thresholds [33], [34]. Other studies presented networks of low-cost accelerometers to provide rapid information after ground shaking [35], [36]. The Earthquake Network [37] and MyShake projects [38] used private smartphone sensors to collect measurements, identify earthquakes, and send data to a central site. Research on IoT technologies for EEWSs has focused on developing new low-cost sensing units, such as low-cost acceleration sensors. However, the EEWSs can benefit from the IoT solutions, as efforts are still required to optimize the seismic station networks by providing minimal data latency at processing hubs alongside highly reliable communications [11]. All components of an EEWS add a delay to the alert time. However, in some seismic networks, the method used to process, trigger, locate, and size an event could have minor effects on the final alert time [20]. Therefore, a crucial contribution to the total alert time is from the latency, primarily due to the transmission time and the propagation delay.

Many international seismic networks comprised stations and are heterogeneous regarding their geographical extent, number of sensors, and density. Each station can have various seismic sensors connected to a datalogger to store data locally and send it to a remote processing hub. The seismic stations of most traditional EEWSs are typically connected to the processing hub via various wired or wireless network topologies and transmit sensor data using the SeedLink protocol [39]. The SeedLink protocol is a de-facto standard for real-time seismic data transmission and uses miniSEED packets. The miniSEED is a subset of the Standard for the Exchange of Earthquake Data (SEED) [40], an international standard format for exchanging digital seismological data. The main problem is that the SeedLink protocol requires fixed-length packets, typically dependent on the sampling rate and proprietary datalogger characteristics. For example, a compression (e.g., Steim 1 or 2 [41]) can represent data in a packet, with a consequent difference in the number of samples contained in each packet. Another option can be to insert a specific time range of acquisitions into the packet (e.g., 1 s). Whatever the mode, a specific amount of time before sending a miniSEED packet must be waited, and this packetization time can significantly delay the final alarm time. Therefore, a more efficient type of packaging and a different communication protocol should be considered to evaluate the benefits regarding alert time compared to traditional EEWS solutions.

This study evaluates the latency and the alert time introduced by an IoT solution for EEWS to transmit data acquired by low-cost seismic stations in real-time. Specifically, this study presents a performance comparison between the de-facto standard for real-time data transmission in traditional EEWSs and a solution based on the Message Queue Telemetry Transport (MQTT) protocol [42], one of the most used protocols in the IoT. In addition to evaluating the latency times achieved by these communication protocols, this study presents a performance analysis of a solution based on the MQTT protocol and structured adjustable-length data packets compared to a traditional one based on SeedLink protocol and miniSEED packets.

The rest of this article is organized as follows. Section II discusses the technical background of the different technologies relevant to the context of this work. Next, the dataset used to simulate a seismic station network is described, and the metrics for performance comparison are identified. Subsequently, the proposed solution is presented, describing the evaluations performed for the analysis (section V)). Section VI discusses the results obtained in the performed tests, and their analysis is presented in Section VII. Finally, conclusions and future developments of our work are discussed in Section VIII.

II. TECHNICAL BACKGROUND

Most traditional EEWSs use the SeedLink protocol for real-time data acquisition from seismic stations. The seismic stations are typically organized in subnets connected with real-time communications to a Local Control Center (LCC) and then to a Network Control Center (NCC). The NCC embeds the processing hub to detect, locate, and characterize an earthquake and send alarms to distant targets. SeedLink is the most widely used communication protocol for exchanging seismic data between seismic station networks and is based on the SEED format. The SEED format is divided into two parts: the time-series data (miniSEED) and the information, such as the network and station identifiers and the instrument responses (dataless SEED). The miniSEED contains a fixed section of the data header that provides information about data encoding and basic information about network and station identifiers, locations, channel identifiers, time-series data length, and time-series data. The miniSEED records are transmitted via SeedLink in 512-byte packets; therefore, waiting for each packet to fill before its transmission is necessary. These miniSEED packets of 512 bytes can be filled with a variable number of samples, depending on the sampling rate and the data compression used by the data logger. If the acquired samples cannot reach the fixed packet length, the datalogger can choose two alternatives: wait for the packetization time necessary to fill a packet or send it by inserting null content up to 512 bytes. If the first approach allows fully exploiting the available bandwidth, the second one is more appropriate in an early warning context, maintaining a constant packetization time. A SeedLink client initiates the connection with a SeedLink server, and during the handshaking phase, it subscribes to specific stations and streams. After the handshaking phase, data are sent to the client as 512-byte miniSEED packets with an 8-byte SeedLink header containing the sequence number. After the TCP/IP connection has been established, SeedLink expects a first handshaking phase, during which the client sends commands to the server, including selecting the stream according to the Seed Station Naming Convention (SSNC). This convention assigns appropriate codes to identify a stream through network, station, location, and channel codes. The International Federation of Digital Seismic Networks assigns the network code and comprises an abbreviation of one or two letters identifying the network (e.g., IV is the Italian National Seismic Network, IX the Irpinia Seismic Network [43]). The station code refers to a physical location where the instruments are sited, represented by five or fewer letters typically registered in the International Registry of Seismograph Stations [44]. The location code is two letters or digits to distinguish similarly named channels or instruments at the same station. Finally, the channel code represents the single data stream and comprises three letters indicating the band code (e.g., H for high broadband), the instrument (e.g., N for accelerometers), and the orientation (e.g., N for north-south).

Data streamed from the seismic stations are typically processed in the processing hub using algorithms for real-time earthquake location, magnitude estimation, and damage assessment. A commonly adopted software platform integrating these algorithms is the PRobabilistic and Evolutionary early warning SysTem (PRESTo) [45], which is under active experimentation in southern Italy on the Irpinia Seismic Network [46]. PRESTo is also in real-time evaluation at the KIGAM network in South Korea, the RoNet network in Romania, and the KOERI network in Turkey. The seismic stations used in PRESTo store the data locally and create miniSEED packets that SeedLink clients receive. Whatever the sampling rate and the data compression used by dataloggers, the seismic stations insert a specific number of samples in miniSEED packets to obtain waveforms of time windows up to one second long. Therefore, each station used in PRESTo maintains the packetization time at 1 s and sends the miniSEED packets to the processing hub via SeedLink protocol. PRESTo processes the data collected from the SeedLink clients using the integrated software for real-time localization and magnitude estimation. The software infrastructure of PRESTo is based on four modules [45] (Fig. 1). After the waveform acquisition from the seismic stations and the preprocessing for quality controls, the system detects the P-wave through an automatic picking algorithm known as FilterPicker [47]. Subsequently, PRESTo processes the P-wave arrivals at the stations to trigger a new event and estimate the ground motion at the targets using RTLoc [48] and RTMag [49] algorithms.

The possibility of implementing efficient EEWS is strongly linked to creating dense heterogeneous seismic station networks comprising IoT nodes with limited power and processing resources [37], [38]. Therefore, the performance of the IoT solutions adopted in EEWSs should be carefully considered. This study evaluates the MQTT protocol

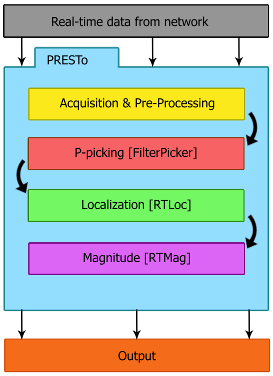

FIGURE 1. Block diagram of the PRESTo components.

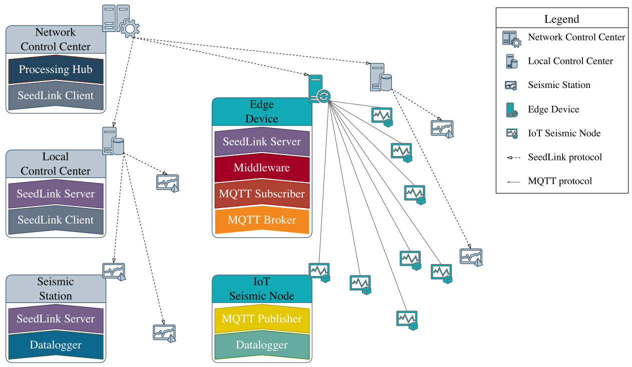

FIGURE 2. Architecture of an EEWS integrating an IoT solution to acquire seismic data from low-cost IoT nodes.

for real-time data acquisition from a seismic station. Due to its simple structure, MQTT can be easily implemented in devices with low-power, low-performance processors and is widely used in IoT data exchange [50] and the main cloud platforms [51]. The MQTT protocol is based on the TCP/IP stack and a publisher/subscriber mechanism [52], enabling distributed, asynchronous, and loosely coupled communication between message producers and consumers. The publish/subscribe messaging relies on a broker with an intermediary role between publishing client messages and receiving subscriber messages. Therefore, the message broker is an intermediary that receives messages from publishers and forwards them to subscribers who have declared that they want to receive them. The publisher does not need to know the destination client, as the broker who keeps track of all the client's subscription requests forwards the message to any subscribers. The broker uses a topic-based hierarchical system to send and receive messages. While publisher clients can arbitrarily choose topics on which to send data, subscriber clients can request subscription to specific topics using a filter created with wildcard characters (i.e. # char, + char). As shown in Fig. 2, an IoT solution based on the MQTT protocol can be fully integrated into an EEWS where the IoT nodes and edge devices function as seismic stations and LCCs, respectively. While each IoT node embeds a low-cost accelerometer and an MQTT publisher to transmit the acquired seismic data, an edge device integrates an MQTT broker and subscriber to receive them. Furthermore, an edge device implements a middleware server to convert the seismic data into miniSEED packets and make them available to a SeedLink server. The server provides seismic data to SeedLink clients, such as the one integrated into an NCC capable of detecting, locating, and characterizing an earthquake and generating an eventual alarm. By using the proposed architecture, it is possible to use topics according to the SSNC (e.g., network/station/location/channel) and any data format for the message payload, such as miniSEED packets or custom structured data packets. With the latter data format, it is possible to use adjustable-length packets and choose the number of samples to be inserted into the packet payload, starting from the theoretical sample-by-sample transmission.

This study evaluates the latency of the MQTT protocol in the EEWS application and demonstrates that such a protocol with adjustable-length packets can significantly reduce the delay introduced by a seismic station using the SeedLink protocol and miniSEED format with fixed-length packets. Several studies in the literature show that the MQTT protocol achieved the best performance for the end-to-end delay and bandwidth consumption compared to other protocols used in IoT applications, such as hypertext transfer protocol [53], advanced message queuing protocol [54], or constrained application protocol [55]. However, the MQTT protocol performance for EEWSs has yet to be evaluated, and this study aims to cover this gap.

III. DATASET

The performance comparison between the solutions based on the MQTT protocol and SeedLink was performed through simulations on a dataset of real earthquakes managed by the Italian National Institute of Geophysics and Volcanology (INGV). The seismic events to be analyzed with such solutions have been selected from the INGV Strong Motion Database (ISMD) [56], containing data recorded by the National Seismic Network (RSN). The RSN is the national permanent seismological network managed by INGV, with contributions from institutions and observers collaborating with them. Over time, it has grown in the number and the quality of the tools installed, also becoming an essential research infrastructure. To date, the RSN comprises approximately 500 monitoring stations distributed throughout the national territory and is enriched by the stations of various Italian and foreign networks and temporary stations that are installed when necessary (e.g., in case of seismic swarm monitoring in geographic subregions of interest). To allow the seismic monitoring of the national territory, each seismic station sends data to the INGV seismic room, where a 24-h service is available for locating and assessing the magnitude of the seismic events in Italy. Data recorded by the RSN and

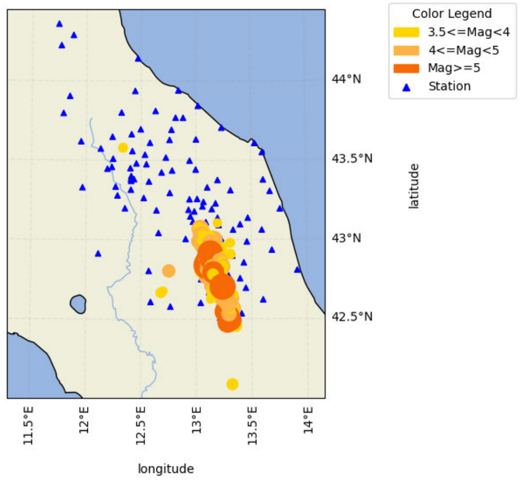

FIGURE 3. Location of the seismic stations and the events considered in the dataset. The blue triangles identify the seismic stations, and the circles indicate the events with different color and magnitude size.

stored in ISMD [56] provide the waveforms and the related metadata of the earthquakes with magnitude ≥ 3.0 in the Italian territory and localized by INGV in quasi-real-time.

All the events with magnitude ≥ 3.5 from 2016-01-01 to 2020-05-27 between 42.02 and 43.82 latitude and 12.04 and 13.4 longitude were selected from the ISMD to provide a performance analysis of the MQTT protocol compared with the SeedLink protocol. The chosen area primarily refers to the epicentral area defined by the earthquakes in Central Italy in 2016-2017, representing a real use case for assessing EEWSs in Italy and an excellent example of complex emergency management [57], [58]. Based on the selected earthquakes, the waveforms of all the stations of the IV seismic network that recorded the data on at least one event were collected, creating the seismic network shown in Fig.3. The map of Fig.3 also reports the geographical distribution of the 200 selected earthquakes with their magnitudes, which are proportional to the circle radius. Data stored by the INGV web service providers were collected by selecting six minutes for each trace on each channel (three minutes before and after the event declaration), obtaining 13,818 traces in the Seismic Analysis Code file format [59]. Subsequently, these traces were processed using the following operations:

1) Obtain the minimum start-time of all traces.
2) Calculate of the time difference between the minimum start time of the dataset and the start time of each trace.
3) Compute of the mean value and standard deviation of the first 800 samples of each trace (quiet moment).
4) Add white noise at the beginning of each trace up to the minimum start time value.

After these operations, a dataset of traces with the same start time was obtained for evaluation. For each test, a suitable script that reads the contents of the dataset containing the accelerometric traces of real seismic events was developed.

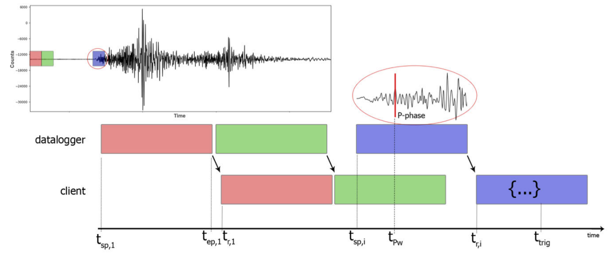

FIGURE 4. Summary of events during earthquake detection. The datalogger of a seismic station fills each packet with a portion of the waveform and sends it to the client. An earthquake starts during the packetization of the blue packet, delaying its detection on the client.

IV. PERFORMANCE METRICS

This study proposes the MQTT protocol in EEWSs and compares its performance with those obtained from a traditional system based on SeedLink. The performance evaluation metrics are defined to study the time differences introduced by the systems based on the MQTT and SeedLink protocols. Referring to Fig.4, on the datalogger of a seismic station, it is possible to define

$$T_{pack,i} = t_{ep,i} - t_{sp,i} \tag{1}$$

where $T_{pack,i}$ is the packetization time of the i-th packet obtained from $t_{sp,i}$ and $t_{ep,i}$, defined as the time of the first and last samples in the i-th packet. $T_{pack,i}$ depends on the sampling rate and the number of samples in each packet.

From this perspective, $t_{r,i}$ is defined as the arrival time of packet i at the client. Therefore, the latency $T_{lat,i}$ of the i-th packet is the difference between the last sample in the packet and $t_{r,i}$,

$$T_{lat,i} = t_{r,i} - t_{ep,i} \tag{2}$$

The picker time T_{pick} is the time required to detect the P-wave once the packet arrives at the processing hub. It is obtained from the difference between the timestamp of the triggering algorithm t_{trig} and $t_{r,i}$,

$$T_{pick} = t_{trig} - t_{r,i} \tag{3}$$

Finally, it is possible to define the total alert time T_{tot} as the sum of packetization, latency, and picker times,

$$T_{tot} = T_{pack,i} + T_{lat,i} + T_{pick} \tag{4a}$$

that is equal to the difference between the trigger time and the timestamp of the first sample in the packet:

$$T_{tot} = t_{trig} - t_{sp,i} \tag{4b}$$

Another relevant performance metric is the delay in detecting the P-phase, which is the difference between the instant of P-phase detection by the triggering algorithm (i.e., the time stamps of P-wave picking, t_{trig}) and the actual instant when the P-wave is generated (t_{Pw}),

$$T_{del} = t_{trig} - t_{Pw} \tag{5}$$

V. EVALUATION TESTS

The MQTT and SeedLink protocols were compared by evaluating the performance in three different tests. The first test compared the latency of the SeedLink and MQTT protocols, which is the time difference between receiving and sending the same miniSEED packet. The second test analyzed the results obtained from the P-wave picking with PRESTo and our proposed solution for P-phase detection. The third test evaluated the delay in detecting the P-phase and the total alert time by comparing the solution based on SeedLink and miniSEED packets with one based on MQTT and structured adjustable-length data packets.

The experimental setup for each test comprised a machine simulating the dataloggers of the seismic stations and another simulating the clients of a processing hub. An appropriate script processing the dataset containing the accelerometer traces of real seismic events was developed for the simulated datalogger side of each test. The script created two processes for each dataset trace addressing packetization and transmission. For the simulated client side of each test, an appropriate script for the real-time reception of the traces transmitted by the simulated dataloggers was developed. A configuration file containing the data on the simulated stations and the related parameters according to the SSNC was passed to the script managing two clients. Each client created as many processes as the stations contained in the file, and each process analyzes the traces received by the simulated dataloggers. Therefore, in each test, it was possible to compare two EEWSs comprising the dataloggers of a generic seismic network and a client in the processing hub role. During the tests, the events on the datalogger side and the client side were recorded through network analyzers and specific log files. The machines used to simulate the dataloggers and clients were synchronized using the network time protocol [60] to obtain the performance metrics. The dataloggers and the clients were simulated on two machines with the following features: Intel Xeon X5650 (x2) central processing unit, 12 MB cache, 2.66 GHz, 16 GB RAM with Ubuntu 18.04.1 LTS.

A. LATENCY TEST

The script developed for the latency test sent the same miniSEED packets via SeedLink and as a payload of an MQTT message to compare the two protocols. Two processes dealing with the same packetization and two transmission modes were activated simultaneously for each dataset trace. The former used the traditional packetization and transmission methods (SeedLink protocol and miniSEED packets), and the second implemented the MQTT protocol to send miniSEED packets. Regarding packetization, both processes created one-second moving-time windows to produce miniSEED packets. While the traditional method sent the miniSEED packets to an internal SeedLink server, the other process sent them to an MQTT broker. Therefore, in the datalogger side of the latency test, the miniSEED packets were transferred every second to a SeedLink server and an MQTT broker installed on the same machine, simulating the real-time behavior of a datalogger network. Concerning the SeedLink server, the Ringserver [61] distributed by the Incorporated Research Institutions for Seismology [62] was installed and configured. As the MQTT broker, Mosquitto was chosen [63]. On the client side of the latency test, a script for the real-time reception of the traces transmitted via the two protocols was developed and installed in a machine under a different network. A SeedLink client and an MQTT client were implemented using the Obspy library [64] and the Paho-MQTT [65] library, respectively. The MQTT protocol supports three levels of quality of service [66], and in our simulations, it was 1 (i.e., at least once) to reproduce the behavior of the SeedLink protocol. Therefore, the simulations performed during this test were complied with a system comprising seismic station dataloggers and a generic seismic network processing hub.

The latencies of the two protocols were evaluated using the data recorded by the network analyzers and the log files, as the analysis conditions were the same (bandwidth, network traffic, and time for packetization, etc.). The latency is the time elapsed between the packet sending and the instant of arrival ($T_{lat,i}$ of Equation 2). To avoid errors due to different processing techniques for each received packet, the receiving time is the instant when the message payload is extracted and processed as a miniSEED packet. All measurements were performed at the end of the connection and subscription phases of the MQTT protocol and after the connection opening and handshaking phases of the SeedLink protocol.

B. P-WAVE PICKING TEST

In the P-wave picking test, the Python implementation of the FilterPicker [67] was evaluated as thetriggering algorithm to be implemented in the proposed EEWS solution. Specifically, two analyses were performed to compare the PRESTo and FilterPicker solutions and verify implementing the latter in a solution based on structured adjustable-length data packets. This analysis was conducted to avoid evaluation errors due to different implementation choices. Specifically, the purpose was to determine whether the time required to detect the P-phase (T_{pick} of Equation 3) in a seismic wave injected into PRESTo was the same as those detected by the solution based on the FilterPicker. PRESTo can be configured and adapted to different networks, providing the configuration files with the details of the seismic stations, the velocity model, the regression law coefficients, and tuning the parameters controlling the data analysis algorithms. In collaboration with the INGV office of Ancona (INGV-AN), the configurations required by PRESTo for the seismic stations considered in the dataset were developed. The speed models for each station, the regression laws coefficients, and the configuration parameters for picking the P-phase were calculated using INGV-AN (Table 1). The acceleration traces of the dataset were injected into PRESTo using a specific script, generating a log file containing the timestamps for seismic event detection. The

TABLE 1. Configuration parameters for the picking of the P-phase.

Parameter	Value
Filter Window (T_{filter})	1.0 s
Long Term Window (T_{long})	5.0 s
Threshold 1 (S_1)	15
Threshold 2 (S_2)	15
tUpEvent (T_{up})	0.1 s

output log file of PRESTo was used as the gold standard to compare the P-wave detection times obtained from the solution based on the FilterPicker.

In the datalogger side of the P-wave picking test, an appropriate script was installed in the machine to process the dataset content to simulate the dataloggers. Two processes addressing packaging and transmission in two methods were activated simultaneously for each dataset trace. While the first process used the traditional packetization and transmission methods, the second implemented a solution based on the MQTT protocol and structured adjustable-length data packets. The traditional method split the trace into one-second windows and filled a miniSEED packet for each window, sending it to an internal SeedLink server. However, the proposed method sent structured adjustable-length data packets via the MQTT. At the client side of the P-wave picking test, specific clients were developed for receiving the traces transmitted through the two methods in real-time and for the P-phase detection. The clients created a process for each simulated datalogger using the Python implementation of the FilterPicker [67] and the same input parameters calculated by INGV-AN for the PRESTo. Each process performed the P-phase picking each time a packet was received over a time window of 10 s.

The client for the real-time reception of the traces transmitted via the traditional method provided the detection timestamp of the P-phase. By comparing this output with the reference log file of PRESTo, almost identical results were obtained for the detection time and real-time calculation on the trace. This simulation was repeated 10 times on each trace, obtaining coincident results for each trial and confirming the correctness of the solution using the FilterPicker for P-wave picking.

The client for the real-time reception of the traces transmitted with the proposed method performed the same operations as the SeedLink client but processed a structured data packet instead of the miniSEED packet. Regarding the format of the structured data packet, the following fields were chosen for the payload structure:

- Timestamp: the instant when the first sample of the value field was acquired.
- Values: the variable-length array containing the samples collected and encoded according to the encoding field.
- Encoding: the string indicating the number of samples in the packet, the sampling rate, and the data format [68].

During the ten simulations, the miniSEED packet of the SeedLink protocol was always filled with 1-s of samples sampled at a 200 Hz. The number of samples in the structured data packets of the MQTT protocol was varied to evaluate the computation load of the machine implementing the MQTT client. Starting from a structured data packet containing one sample at the sampling rate, the number of samples in the payload of the MQTT message was 2^{n-1} in each trial, where n indicates the progressive number of performed trials. The FilterPicker algorithm was invoked on the client side each time a packet was received over a time window of 10 s. The simulation results showed that this operation is too expensive for the trials with less than 16 samples due to the very high central processing unit usage of the machine simulating the client.

C. ALERT TIME TEST

This study evaluates methodologies and protocols used before the event detection performed by an algorithm for P-wave picking. Therefore, in the alert time test, the solution based on MQTT and the one based on SeedLink implemented the same triggering algorithm to provide a performance comparison of the P-wave detection time. In the datalogger side of the alert time test, the proposed method was modified to send structured data packets containing 250 ms of samples via MQTT. According to the previous P-wave picking test, this number of samples per packet is a good trade-off between the correctness of the results obtained by the triggering algorithm and its execution time. Compared to the previous test, the client of the proposed method was modified to process the structured data packet of the chosen length and invoke the triggering algorithm for each packet received.

The delay in detecting the P-phase (T_{del} of Equation 5) and the total alert time (T_{tot} of Equation 4b) were evaluated using the same methodology as in the latency test. The delay in detecting the P-phase is the time difference between the P-phase detection performed by the triggering algorithm of the processing hub and the actual instant where the P-wave was generated. This time difference considers the uneven packetization times because the P-wave could be in a random position in the packet that fills the datalogger before sending it. The total alert time is the time difference between the instant where the P-wave is detected by the triggering algorithm of the client and the beginning of the packet containing the wave itself.

VI. RESULTS

Table 2 summarizes the results obtained in the latency test. The MQTT protocol reported the best results for latency, reaching an average delivery time of 33.13 ms, with a standard deviation of 17.03 ms. The SeedLink protocol achieved an average delay of 696.13 ms, with a standard deviation of 439.90 ms. Table 2 also reports the median and 90th percentile measurements for completeness of the information.

Fig.5 shows the probability distribution graph obtained from the simulations, dividing the abscissa axis into bins

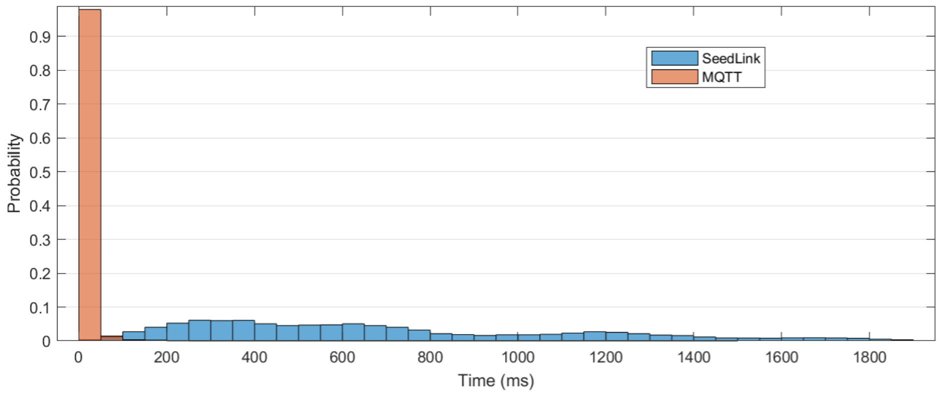

FIGURE 5. The probability distribution of latency, T_{lat}, for the MQTT and SeedLink protocols.

TABLE 2. Statistics of the latency.

Protocol	Mean (ms)	St. Dev. (ms)	Median (ms)	90th perc. (ms)
MQTT	33.13	17.03	30.79	33.37
SeedLink	696.13	439.90	592.83	1346.64

of 50 ms. The histograms are normalized, and the relative frequencies of the total analyzed samples equal 13,818 traces multiplied by the 200 samples in each second of trace. The graph shows different distributions among the MQTT and SeedLink protocols. The MQTT protocol distribution has a positive asymmetry, with an interquartile range of 1.074 ms. Compared with the 90th percentile and the median value in Table 2, this value shows that the protocol performance was stable during all simulations, with sporadic outliers. The SeedLink protocol distribution shows the approximation to a multimodal function, with a first mode value of 250-300 ms, a second of 600-650 ms, and a third value of 1150-1220 ms. Three mean and standard deviation values were obtained for a distribution suitable for these characteristics. The average value and the standard deviation for the approximation to the first, second, and third distributions are 280 ± 78, 597 ± 89, and 1187 ± 235, respectively. Therefore, the simulation analysis conducted in the latency test showed a nonuniform behavior of the SeedLink protocol, with great variability of the results given the range of the values obtained. The latency test was repeated by installing the SeedLink client in the same machine that simulates the dataloggers to confirm that network instability factors did not influence the results, thus eliminating latency due to the network. Repeating the latency test achieved almost identical distributions compared to previous simulations. These results confirm the higher instability of the SeedLink than the MQTT protocol and significantly higher average latency times. Therefore, in the latency test, the MQTT protocol achieved better performance for latency and stability than the SeedLink protocol, making it an excellent choice for EEWS applications where time and reliability are critical.

TABLE 3. Statistics of the delay in detecting the P-Phase.

Protocol	Mean (ms)	St. Dev. (ms)	Median (ms)	90th perc. (ms)
MQTT	751.93	195.69	685.32	1236.74
SeedLink	2057.52	639.28	1958.57	2684.61

TABLE 4. Statistics of the total alert time.

Protocol	Mean (ms)	St. Dev. (ms)	Median (ms)	90th perc. (ms)
MQTT	910.69	395.06	840.34	1491.73
SeedLink	2558.94	828.48	2388.80	3311.35

Table 3 shows the simulation for the delay in detecting the P-phase evaluated in the alert time test. By comparing the obtained average values, it is possible to observe that the proposed solution allowed the detection of the P-phase with an advance of approximately 1300 ms compared to the traditional miniSEED packetization and the SeedLink protocol. This value is also similar comparing the median and the 90th percentile values. Furthermore, the standard deviation analysis shows that the values obtained using the SeedLink protocol were much wider than those obtained by the MQTT protocol, confirming that the proposed solution also contributes to uncertainty regarding detection times.

Fig. 6 shows the distributions of the values from the simulations of the alert time test to provide additional information regarding the probability distributions. These graphs are normalized for probabilities and the bins are set at 100 ms. The traditional methods show uneven behavior, confirming that the contribution of latency was significant in this test. Greater uniformity can be seen from the distribution of the values from the simulations performed using the MQTT protocol. This behavior was expected after analyzing the protocol latencies, as the distribution had a positive asymmetry but an interquartile range in the order of a millisecond.

Table 4 shows the statistics regarding the total alert time. By comparing the average and median values, the proposed solution showed a performance improvement of

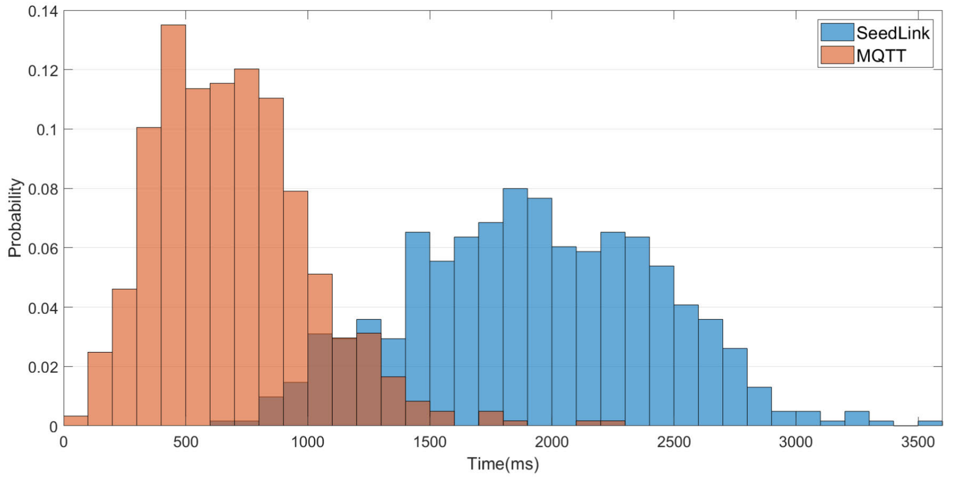

FIGURE 6. The probability distribution of the delay in detecting the P-phase, T_{del}, for the solutions based on the MQTT and SeedLink protocols.

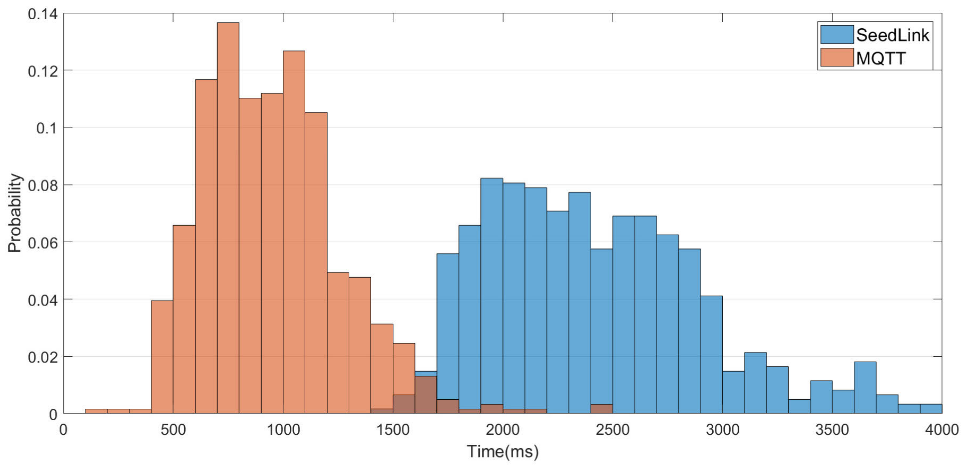

FIGURE 7. The probability distribution of the total alert time, T_{tot}, for the solutions based on the MQTT and SeedLink protocols.

approximately 1600 ms. Furthermore, the standard deviation values suggest uniform behavior. The average displacement of approximately 260 ms of the obtained results in the proposed solution correlates with what was found in the P-wave picking test. The SeedLink protocol results confirm what occurred previously in the latency and P-wave picking tests.

Fig.7 shows the probability distributions of the total alert time evaluated in the alert time test. Comparing it with that obtained from the previous analysis (Fig. 6), the proposed solution was more stable and constantly influenced by packaging. The differences in the distribution shapes obtained by SeedLink suggest that packaging and latency contributed heavily to the total detection times.

VII. DISCUSSION

EEWSs represent one of the most promising practical approaches to reducing damages caused by earthquakes. The rapidity with which these systems can generate an alert becomes essential to evacuate the population and mobilize the emergency response promptly. An alert generated only a few seconds before the actual arrival of the destructive waves of an earthquake can prove a considerable advantage for human lives saved [23].

This study evaluates the possibility of use a solution based on the MQTT protocol to reduce the latency of the communication protocol and the total alert time of an EEWS. Therefore, evaluation tests were conducted to compare the obtained performances with those achievable by a traditional EEWS. In the first test, the latency of the MQTT protocol was compared to that of the SeedLink protocol. The latency test results showed that the MQTT protocol reports an average latency per packet of 33.13 ms, with stable performance during all simulations. However, the SeedLink protocol showed nonuniform results and an average latency of 696.13 ms. The difference between the performances is due to the inherent latencies of the specific application layer protocol used in each solution. Each protocol has various latencies due to the protocol itself and varies considerably in the SeedLink protocol. Given the reduced latency per packet achieved with the MQTT protocol, this protocol can send structured data packets with fewer samples, and its benefits for detection and total alert times were evaluated in subsequent tests. In the alert time test, a solution based on the MQTT protocol and structured data packets with 250 ms of samples was compared with a traditional one based on the SeedLink protocol and one-second miniSEED packets. The results showed that the solution based on the MQTT protocol and structured data packets detected the P-phase of the seismic events of the dataset, with an average advance of approximately 1300 ms. Additionally, the MQTT protocol obtained a performance improvement of 1600 ms for total alert time. The probability distributions obtained in the Alert Time Test are comparable with those obtained in other studies [20], [45] but depend on the seismic events with which the dataset was created and the chosen algorithm for P-phase detection. Comparing the probability distributions of the solution based on the MQTT with that related to SeedLink makes it possible to analyze the results obtained from the different packetization and transmission modes in this specific case study. The lower latency of the communication protocol and its reduced packetization time improved the performance of the MQTT solution.

The analysis presented in this study was based on the simulation results of different packetization and transmission modes of seismic data. These simulations were performed starting from the dataset of earthquakes data relating to the seismic sequence in Central Italy in 2016-2017. Therefore, the results were validated through simulations of this case study but were not analyzed in a real scenario where the alert times are influenced by the latency due to propagation time, congestion, etc. Thus, the simulations based on datasets of different case studies and evaluation tests of the performance in a real scenario should be performed to further validate the results.

Based on the results of this study and the effectiveness of data collection from low-cost sensors, the proposed solution can be easily integrated into existing early warning systems for other crises, such as floods, earthquakes, tsunamis, and landslides. By developing a dedicated middleware (Fig. 2) for each application context, the proposed IoT architecture can be used to allow a denser presence of alert networks on the territory and provide timely warnings and essential data and instruments to the authorities for assuring economic and societal benefits by reducing risks associated with disastrous events [23].

VIII. CONCLUSION

In an EEWS, the greater the time available before a seismic event hits a target, the more effective actions can be taken. Therefore, studying the delays that each system component adds is critical. This study compared a traditional seismic monitoring system with a typical IoT architecture to evaluate the delay introduced by the protocol used for seismic wave transmission. With specific reference to Italy, where the most active seismic zones are very close to urban centers, even small improvements regaring time can be fundamental. Therefore, the performance of a traditional EEWS, which uses the SeedLink protocol, and our solution based on the transmission of waveforms via the MQTT protocol were compared. Starting from a dataset of over 13,000 accelerometric traces of real seismic events, the latencies of the two protocols were analyzed, sending the same miniSEED packet via SeedLink and as a payload of an MQTT message. After demonstrating that the MQTT protocol achieved advantages in latency and stability, the performance of a solution based on the MQTT and structured adjustable-length data was evaluated. The P-phase sampling results and those obtained using traditional EEWS showed that the proposed solution could anticipate the detection by approximately 1.3 s, obtaining a total alert time of approximately 1.6 seconds lower.

As future prospects, it will be interesting to create IoT seismic nodes implementing the proposed solution and install them near active seismic stations to verify directly in the field the improvements this work has suggested regarding detection times. Another future application could be studying the integration of devices with edge/fog computing technologies. For example, verifying the possibility of triggering directly on the datalogger and then sending the waves to a processing hub, which will apply data fusion algorithms and generate the alert, might be intriguing.

APPENDIX A ABBREVIATIONS

EEWS: Earthquake Early Warning System
EEWSs: Earthquake Early Warning Systems
INGV: Italian National Institute of Geophysics and Volcanology
INGV-AN: INGV, office of Ancona
IoT: Internet of Things
ISMD: INGV Strong Motion Database
LLC: Local Control Center
MQTT: Message Queue Telemetry Transport
NCC: Network Control Center
P-waves: Primary waves
PRESTo: PRobabilistic and Evolutionary early warning SysTem
RSN: National Seismic Network
S-waves: Secondary waves
SEED: Exchange of Earthquake Data
SSNC: Seed Station Naming Convention

REFERENCES

[1] H. Kanamori, E. Hauksson, and T. Heaton, "Real-time seismology and earthquake hazard mitigation," *Nature*, vol. 390, no. 6659, pp. 461–464, Dec. 1997.
[2] T.-L. Teng, L. Wu, T.-C. Shin, Y.-B. Tsai, and W. H. K. Lee, "One minute after: Strong-motion map, effective epicenter, and effective magnitude," *Bull. Seismol. Soc. Amer.*, vol. 87, no. 5, pp. 1209–1219, Oct. 1997.
[3] B. Dorsemaine, J.-P. Gaulier, J.-P. Wary, N. Kheir, and P. Urien, "Internet of Things: A definition and taxonomy," in *Proc. 9th Int. Conf. Next Gener. Mobile Appl., Services Technol.*, Sep. 2015, pp. 72–77.
[4] Y. Nakamura, "Development of earthquake early-warning system for the Shinkansen, some recent earthquake engineering research and practical in Japan," Jpn. Nat. Committee Int. Assoc. Earthq. Eng., 1984, pp. 224–238.
[5] T. H. Heaton, "A model for a seismic computerized alert network," *Science*, vol. 228, no. 4702, pp. 987–990, May 1985. [Online]. Available: https://www.science.org/doi/10.1126/science.228.4702.987

[6] W. H. K. Lee and J. M. Espinosa-Aranda, "Earthquake early warning systems: Current status and perspectives," in *Early Warning Systems for Natural Disaster Reduction*, J. Zschau and A. Küppers, Eds. Berlin, Germany: Springer, 2003, doi: 10.1007/978-3-642-55903-7_53.

[7] C. Satriano, Y.-M. Wu, A. Zollo, and H. Kanamori, "Earthquake early warning: Concepts, methods and physical grounds," *Soil Dyn. Earthq. Eng.*, vol. 31, no. 2, pp. 106–118, Feb. 2011.

[8] Y.-M. Wu, "Rapid assessment of damage potential of earthquakes in Taiwan from the beginning of *P* waves," *Bull. Seismol. Soc. Amer.*, vol. 95, no. 3, pp. 1181–1185, Jun. 2005.

[9] O. Kamigaichi, M. Saito, K. Doi, T. Matsumori, S. Tsukada, K. Takeda, T. Shimoyama, K. Nakamura, M. Kiyomoto, and Y. Watanabe, "Earthquake early warning in Japan: Warning the general public and future prospects," *Seismol. Res. Lett.*, vol. 80, no. 5, pp. 717–726, Sep. 2009.

[10] R. M. Allen, P. Gasparini, O. Kamigaichi, and M. Bose, "The status of earthquake early warning around the world: An introductory overview," *Seismol. Res. Lett.*, vol. 80, no. 5, pp. 682–693, Sep. 2009.

[11] J. Clinton, A. Zollo, A. Marmureanu, C. Zulfikar, and S. Parolai, "State-of-the art and future of earthquake early warning in the European region," *Bull. Earthq. Eng.*, vol. 14, no. 9, pp. 2441–2458, Sep. 2016.

[12] K. Doi, "The operation and performance of earthquake early warnings by the Japan meteorological agency," *Soil Dyn. Earthq. Eng.*, vol. 31, no. 2, pp. 119–126, Feb. 2011.

[13] M. Böse, E. Hauksson, K. Solanki, H. Kanamori, Y. Wu, and T. Heaton, "A new trigger criterion for improved real-time performance of onsite earthquake early warning in southern California," *Bull. Seismol. Soc. Amer.*, vol. 99, no. 2A, pp. 897–905, Apr. 2009.

[14] H. S. Kuyuk, A. Pinar, M. Comoglu, and M. O. Erdik, "Performance of network based EEW systems in Marmara region; VS, ElarmS-2 and PRESTo," in *Proc. AGU Fall Meeting Abstr.*, 2015, pp. S33B–2780.

[15] A. Chung, I. Henson, and R. Allen, "Optimizing earthquake early warning performance: ElarmS-3," *Seismol. Res. Lett.*, vol. 90, pp. 727–743, Mar. 2019.

[16] R. M. Allen, "The ElarmS earthquake early warning methodology and application across California," in *Earthquake Early Warning Systems*. Berlin, Germany: Springer, 2007, pp. 21–43.

[17] H. M. Brown, R. M. Allen, M. Hellweg, O. Khainovski, D. Neuhauser, and A. Souf, "Development of the ElarmS methodology for earthquake early warning: Realtime application in California and offline testing in Japan," *Soil Dyn. Earthq. Eng.*, vol. 31, no. 2, pp. 188–200, Feb. 2011.

[18] G. Festa, M. Picozzi, A. Caruso, S. Colombelli, M. Cattaneo, L. Chiaraluce, L. Elia, C. Martino, S. Marzorati, M. Supino, and A. Zollo, "Performance of earthquake early warning systems during the 2016–2017 Mw 5–6.5 central Italy sequence," *Seismol. Res. Lett.*, vol. 89, no. 1, pp. 1–12, Jan. 2018.

[19] C. Ladina, S. Marzorati, A. Amato, and M. Cattaneo, "Feasibility study of an earthquake early warning system in eastern central Italy," *Frontiers Earth Sci.*, vol. 9, Aug. 2021, Art. no. 685751.

[20] Y. Behr, J. Clinton, P. Kastli, C. Cauzzi, R. Racine, and M.-A. Meier, "Anatomy of an earthquake early warning (EEW) alert: Predicting time delays for an end-to-end EEW system," *Seismol. Res. Lett.*, vol. 86, no. 3, pp. 830–840, May 2015.

[21] G. Cua and T. Heaton, "The virtual seismologist (VS) method: A Bayesian approach to earthquake early warning," in *Earthquake Early Warning Systems*. Berlin, Germany: Springer, 2007, pp. 97–132.

[22] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Generat. Comput. Syst.*, vol. 29, no. 7, pp. 1645–1660, Sep. 2013.

[23] M. Esposito, L. Palma, A. Belli, L. Sabbatini, and P. Pierleoni, "Recent advances in Internet of Things solutions for early warning systems: A review," *Sensors*, vol. 22, no. 6, p. 2124, Mar. 2022.

[24] M. Talebkhah, A. Sali, M. Marjani, M. Gordan, S. J. Hashim, and F. Z. Rokhani, "IoT and big data applications in smart cities: Recent advances, challenges, and critical issues," *IEEE Access*, vol. 9, pp. 55465–55484, 2021.

[25] P. Pierleoni, A. Belli, L. Palma, S. Valenti, S. Raggiunto, L. Incipini, and P. Ceregioli, "The Scrovegni chapel moves into the future: An innovative Internet of Things solution brings new light to Giotto's masterpiece," *IEEE Sensors J.*, vol. 18, no. 18, pp. 7681–7696, Sep. 2018.

[26] M. Abdelraheem, M. Abdelhafeez, and A. Nassr, "IoT-based interdigital capacitance sensing system for damage detection in CFRP-concrete structures," *IEEE Access*, vol. 9, pp. 138658–138667, 2021.

[27] S. Valenti, M. Conti, P. Pierleoni, L. Zappelli, A. Belli, F. Gara, S. Carbonari, and M. Regni, "A low cost wireless sensor node for building monitoring," in *Proc. IEEE Workshop Environ., Energy, Struct. Monit. Syst. (EESMS)*, Jun. 2018, pp. 1–6.

[28] X. Geng, Q. Zhang, Q. Wei, T. Zhang, Y. Cai, Y. Liang, and X. Sun, "A mobile greenhouse environment monitoring system based on the Internet of Things," *IEEE Access*, vol. 7, pp. 135832–135844, 2019.

[29] P. Pierleoni, S. Marzorati, C. Ladina, S. Raggiunto, A. Belli, L. Palma, M. Cattaneo, and S. Valenti, "Performance evaluation of a low-cost sensing unit for seismic applications: Field testing during seismic events of 2016–2017 in central Italy," *IEEE Sensors J.*, vol. 18, no. 16, pp. 6644–6659, Aug. 2018.

[30] G. Mei, N. Xu, J. Qin, B. Wang, and P. Qi, "A survey of Internet of Things (IoT) for Geohazard prevention: Applications, technologies, and challenges," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 4371–4386, May 2020.

[31] J. Lee, I. Khan, S. Choi, and Y.-W. Kwon, "A smart IoT device for detecting and responding to earthquakes," *Electronics*, vol. 8, no. 12, p. 1546, Dec. 2019.

[32] A. Alphonsa and G. Ravi, "Earthquake early warning system by IoT using wireless sensor networks," in *Proc. Int. Conf. Wireless Commun., Signal Process. Netw. (WiSPNET)*, Mar. 2016, pp. 1201–1205.

[33] Y.-M. Wu, D.-Y. Chen, T.-L. Lin, C.-Y. Hsieh, T.-L. Chin, W.-Y. Chang, W.-S. Li, and S.-H. Ker, "A high-density seismic network for earthquake early warning in Taiwan based on low cost sensors," *Seismol. Res. Lett.*, vol. 84, pp. 1048–1054, Nov. 2013.

[34] Y. Wu, W. Liang, H. Mittal, W. Chao, C. Lin, B. Huang, and C. Lin, "Performance of a low-cost earthquake early warning system (P-alert) during the 2016 ML 6.4 Meinong (Taiwan) earthquake," *Seismol. Res. Lett.*, vol. 87, no. 5, pp. 1050–1059, Sep. 2016.

[35] R. W. Clayton, T. Heaton, M. Kohler, M. Chandy, R. Guy, and J. Bunn, "Community seismic network: A dense array to sense earthquake strong motion," *Seismological Res. Lett.*, vol. 86, no. 5, pp. 1354–1363, Sep. 2015.

[36] E. S. Cochran, J. F. Lawrence, C. Christensen, and R. S. Jakka, "The quake-catcher network: Citizen science expanding seismic horizons," *Seismol. Res. Lett.*, vol. 80, no. 1, pp. 26–30, Jan. 2009.

[37] F. Finazzi, "The earthquake network project: Toward a crowdsourced smartphone-based earthquake early warning system," *Bull. Seismol. Soc. Amer.*, vol. 106, no. 3, pp. 1088–1099, May 2016.

[38] Q. Kong, R. M. Allen, L. Schreier, and Y.-W. Kwon, "MyShake: A smartphone seismic network for earthquake early warning and beyond," *Sci. Adv.*, vol. 2, no. 2, Feb. 2016, Art. no. e1501055.

[39] *Iris: Seedlink*. Accessed: Mar. 30, 2020. [Online]. Available: http://ds.iris.edu/ds/nodes/dmc/services/seedlink/

[40] T. Ahern and B. Dost. *Seed: Standard for the Exchange of Earthquake Data*. Accessed: Oct. 24, 2022. [Online]. Available: http://www.fdsn.org/pdf/SEEDManual_V2.4.pdf

[41] S. Anandakrishnan, *Steim Compression*. New York, NY, USA: Apress, 2018, pp. 17–23.

[42] G. Kim, S. Kang, J. Park, and K. Chung, "An MQTT-based context-aware autonomous system in oneM2M architecture," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8519–8528, Oct. 2019.

[43] *Seismic Networks*. Accessed: Oct. 24, 2022. [Online]. Available: http://terremoti.ingv.it/instruments

[44] *International Registry of Seismograph Stations*. Accessed: Oct. 24, 2022. [Online]. Available: http://www.isc.ac.uk/registries/

[45] C. Satriano, L. Elia, C. Martino, M. Lancieri, A. Zollo, and G. Iannaccone, "PRESTo, the earthquake early warning system for southern Italy: Concepts, capabilities and future perspectives," *Soil Dyn. Earthq. Eng.*, vol. 31, no. 2, pp. 137–153, Feb. 2011.

[46] E. Weber, V. Convertito, G. Iannaccone, A. Zollo, A. Bobbio, L. Cantore, M. Corciulo, M. Di Crosta, L. Elia, C. Martino, A. Romeo, and C. Satriano, "An advanced seismic network in the southern apennines (Italy) for seismicity investigations and experimentation with earthquake early warning," *Seismol. Res. Lett.*, vol. 78, no. 6, pp. 622–634, Nov. 2007.

[47] A. Lomax, C. Satriano, and M. Vassallo, "Automatic picker developments and optimization: FilterPicker—A robust, broadband picker for real-time seismic monitoring and earthquake early warning," *Seismol. Res. Lett.*, vol. 83, no. 3, pp. 531–540, May 2012.

[48] C. Satriano, A. Lomax, and A. Zollo, "Real-time evolutionary earthquake location for seismic early warning," *Bull. Seismol. Soc. Amer.*, vol. 98, no. 3, pp. 1482–1494, Jun. 2008.

[49] M. Lancieri and A. Zollo, "A Bayesian approach to the real-time estimation of magnitude from the early *P* and *S* wave displacement peaks," *J. Geophys. Res.*, vol. 113, p. B12302, Dec. 2008.

[50] F. Chen, Y. Huo, J. Zhu, and D. Fan, "A review on the study on MQTT security challenge," in *Proc. IEEE Int. Conf. Smart Cloud (SmartCloud)*, Nov. 2020, pp. 128–133.

[51] P. Pierleoni, R. Concetti, A. Belli, and L. Palma, "Amazon, Google and Microsoft solutions for IoT: Architectures and a performance comparison," *IEEE Access*, vol. 8, pp. 5455–5470, 2020.
[52] D. Happ, N. Karowski, T. Menzel, V. Handziski, and A. Wolisz, "Meeting IoT platform requirements with open pub/sub solutions," *Ann. Telecommun.*, vol. 72, no. 1, pp. 41–52, Feb. 2017.
[53] T. Yokotani and Y. Sasaki, "Comparison with HTTP and MQTT on required network resources for IoT," in *Proc. Int. Conf. Control, Electron., Renew. Energy Commun. (ICCEREC)*, Sep. 2016, pp. 1–6.
[54] J. E. Luzuriaga, M. Perez, P. Boronat, J. C. Cano, C. Calafate, and P. Manzoni, "A comparative evaluation of AMQP and MQTT protocols over unstable and mobile networks," in *Proc. 12th Annu. IEEE Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2015, pp. 931–936.
[55] D. Thangavel, X. Ma, A. Valera, H.-X. Tan, and C. K.-Y. Tan, "Performance evaluation of MQTT and CoAP via a common middleware," in *Proc. IEEE 9th Int. Conf. Intell. Sensors, Sensor Netw. Inf. Process. (ISSNIP)*, Apr. 2014, pp. 1–6.
[56] M. Massa, S. Lovati, G. Franceschina, E. D'Alema, S. Marzorati, S. Mazza, M. Cattaneo, G. Selvaggi, A. Amato, A. Michelini, and P. Augliera, "ISMD, a web portal for real-time processing and dissemination of INGV strong-motion data," *Seismol. Res. Lett.*, vol. 85, no. 4, pp. 863–877, Jul. 2014.
[57] A. C. Valles, M. M. Ferrer, K. Poljansek, and I. Clark, Eds., "Science for disaster risk management 2020," Publications Office Eur. Union, Luxembourg, Tech. Rep. EUR 30183 EN, JRC114026, 2021. [Online]. Available: https://publications.jrc.ec.europa.eu/repository/handle/JRC114026, doi: 10.2760/571085.
[58] L. Chiaraluce, R. Di Stefano, E. Tinti, L. Scognamiglio, M. Michele, E. Casarotti, M. Cattaneo, P. De Gori, C. Chiarabba, G. Monachesi, A. Lombardi, L. Valoroso, D. Latorre, and S. Marzorati, "The 2016 central Italy seismic sequence: A first look at the mainshocks, aftershocks, and source models," *Seismol. Res. Lett.*, vol. 88, no. 3, pp. 757–771, May 2017.
[59] *Iris: SAC Data File Format*. Accessed: Mar. 30, 2020. [Online]. Available: http://www.adc1.iris.edu/files/sac-manual/manual/file_format.html
[60] D. Mills, J. Martin, J. Burbank, and W. Kasch. *Network Time Protocol Version 4: Protocol and Algorithms Specification*. Fremont, CA, USA: Internet Engineering Task Force (IETF), Jun. 2010. [Online]. Available: http://www.rfc-editor.org/rfc/rfc5905.txt
[61] *Iris Supported Software: Ringserver*. Accessed: Oct. 24, 2022. [Online]. Available: https://seiscode.iris.washington.edu/projects/ringserver
[62] *Incorporated Research Institutions for Seismology*. Accessed: Oct. 24, 2022. [Online]. Available: http://www.iris.edu
[63] R. A. Light, "Mosquitto: Server and client implementation of the MQTT protocol," *J. Open Source Softw.*, vol. 2, no. 13, p. 265, May 2017.
[64] M. Beyreuther, R. Barsch, L. Krischer, T. Megies, Y. Behr, and J. Wassermann, "ObsPy: A Python toolbox for seismology," *Seismol. Res. Lett.*, vol. 81, no. 3, pp. 530–533, May 2010.
[65] *Eclipse Paho Pyhthon Client*. Accessed: Oct. 24, 2022. [Online]. Available: https://www.eclipse.org/paho/clients/python/
[66] S. Behnel, L. Fiege, and G. Muhl, "On quality-of-service and publish-subscribe," in *Proc. 26th IEEE Int. Conf. Distrib. Comput. Syst. Workshops (ICDCSW)*, Jul. 2006, p. 20.
[67] M. Bagagli, "Filterpicker," Zenodo, CERN, Eur. Org. Nucl. Res., IT Dept., Digit. Repositories Section, Geneva, Switzerland, Tech. Rep. 6511985, Nov. 2019, doi: 10.5281/zenodo.3609025.
[68] A. T. Ringler and J. R. Evans, "A quick SEED tutorial," *Seismol. Res. Lett.*, vol. 86, no. 6, pp. 1717–1725, Nov. 2015.

PAOLA PIERLEONI received the master's degree in electronic engineering and the Ph.D. degree in electrical engineering from Università Politecnica delle Marche, in 1991 and 1995, respectively. In 1991, she joined the Department of Information Engineering, Università Politecnica delle Marche, where she is currently an Assistant Professor of telecommunications. Her main research interests include network protocols, wireless sensor networks, the Internet of Things, signal processing, and embedded devices development.

ROBERTO CONCETTI received the master's degree in computer engineering and the Ph.D. degree in informatics, management, and automatics engineering from Università Politecnica delle Marche, Italy, in 2016 and 2021, respectively. He is currently a High School Teacher with IISS Carlo Urbani, Porto Sant'Elpidio, Italy. His current research interests include computer networks, cloud computing for the Internet of Things, and their applications in earthquake early warning systems.

SIMONE MARZORATI received the master's degree in environmental science and the Ph.D. degree in geological sciences from the University of Milano–Bicocca, in 2001 and 2007, respectively. Since 2001, he has been with Istituto Nazionale di Geofisica e Vulcanologia, where he is currently a Researcher. His main research interests include seismic network monitoring, seismic signal processing, rapid strong motion assessment, and seismic data quality.

ALBERTO BELLI received the master's degree in telecommunications engineering and the Ph.D. degree in biomedical, electronic, and telecommunications engineering from Università Politecnica delle Marche, Italy, in 2012 and 2016, respectively. From 2016 to 2018, he was a Research Fellow with the Department of Information Engineering (DII), Università Politecnica delle Marche, where he has been a Staff Scientist, since 2019. His research interests include sensor networks for the Internet of Things solution, data fusion algorithms for array sensors, and wearable devices applications.

LORENZO PALMA (Member, IEEE) received the master's degree (cum laude) in electronic engineering and the Ph.D. degree in information engineering from Università Politecnica delle Marche, Italy, in 2012 and 2017, respectively. He is currently an Assistant Professor (RTD-B) with Università Politecnica delle Marche. His main research interests include the IoT, embedded artificial intelligence, wireless sensors networks, embedded systems, communication networks, and internet of medical things.

Open Access funding provided by 'Università Politecnica delle Marche' within the CRUI CARE Agreement