MDPI

Article

"Saving Precious Seconds"—A Novel Approach to Implementing a Low-Cost Earthquake Early Warning System with Node-Level Detection and Alert Generation

Raj Prasanna [1,*], Chanthujan Chandrakumar [1], Rasika Nandana [2], Caroline Holden [3], Amal Punchihewa [4], Julia S. Becker [1], Seokho Jeong [5], Nandika Liyanage [6], Danuka Ravishan [7], Rangana Sampath [8] and Marion Lara Tan [1]

[1] Joint Centre for Disaster Research, Massey University, Wellington 6021, New Zealand; c.chandrakumar2@massey.ac.nz (C.C.); j.becker@massey.ac.nz (J.S.B.); m.l.tan@massey.ac.nz (M.L.T.)
[2] School of Engineering and Computer Science, Victoria University of Wellington, Wellington 6012, New Zealand; rasika.walakuluarachchige@vuw.ac.nz
[3] SeismoCity Ltd., Wellington 6023, New Zealand; caroline.holden@seismocity.co.nz
[4] ADP Consultancy, Palmerston North 4410, New Zealand; amal_yahu@yahoo.com.sg
[5] Department of Civil Engineering, Changwon National University, Changwon 51140, Korea; seokho.jeong@changwon.ac.kr
[6] ALSO Germany, 94315 Straubing, Germany; nandika.liyanage@gmail.com
[7] Synopsys, Colombo 00800, Sri Lanka; danukaravishan1@gmail.com
[8] Datacom, Wellington 6011, New Zealand; rangana@outlook.com
* Correspondence: r.prasanna@massey.ac.nz

Abstract: This paper presents findings from ongoing research that explores the ability to use Micro-Electromechanical Systems (MEMS)-based technologies and various digital communication protocols for earthquake early warning (EEW). The paper proposes a step-by-step guide to developing a unique EEW network architecture driven by a Software-Defined Wide Area Network (SD-WAN)-based hole-punching technology consisting of MEMS-based, low-cost accelerometers hosted by the general public. In contrast with most centralised cloud-based approaches, a node-level decentralised data-processing is used to generate warnings with the support of a modified Propagation of Local Undamped Motion (PLUM)-based EEW algorithm. With several hypothetical earthquake scenarios, experiments were conducted to evaluate the system latencies of the proposed decentralised EEW architecture and its performance was compared with traditional centralised EEW architecture. The results from sixty simulations show that the SD-WAN-based hole-punching architecture supported by the Transmission Control Protocol (TCP) creates the optimum alerting conditions. Furthermore, the results provide clear evidence to show that the decentralised EEW system architecture can outperform the centralised EEW architecture and can save valuable seconds when generating EEW, leading to a longer warning time for the end-user. This paper contributes to the EEW literature by proposing a novel EEW network architecture.

Keywords: earthquake early warning; software-defined network; PLUM algorithm; decentralised architecture; MEMS; citizen science

1. Introduction

With increasing population and urbanisation across the globe, large earthquakes have become a significant threat to human life and infrastructure, especially for places closer to active earthquake faults [1]. In this context, interest in issuing earthquake early warnings (EEWs) is increasing globally, and research has found significant benefits of having EEW systems to warn the public [2]. EEW can be a beneficial tool for warning people when an earthquake occurs. EEW can warn people in areas where the anticipated ground-shaking due to an earthquake could cause harm or destruction [3]. Depending on the size and depth

of the earthquake and the number and type of sensors used to build the EEW system, the warning window can range from a few seconds to tens of seconds. Research revealed that even a 20–30 s warning lead-time could allow people to take simple protective actions such as drop–cover–hold and mentally prepare themselves for an impending earthquake [4–6]. Even a couple of seconds could be extremely useful, as they can provide enough time for automated systems to initiate precautionary emergency measures, such as stopping trains to minimise potential derailment, the appropriate shutting-off of gas distribution valves to reduce fire risk, and the orderly switching-off of large, heavy machinery to minimise potential losses [1].

Ongoing technological innovations in earthquake-monitoring tools, telecommunication, earthquake-detection algorithms, and processing capabilities have created new opportunities to develop EEW systems and provide opportunities for further enhancements [7]. At present, EEW systems are operational in several countries and territories worldwide [8]. Although these systems are robust, implementing them can be complex and expensive. The ShakeAlert EEW system in the USA took 15 years to create and is currently deployed in only three states: California, Oregon, and Washington. The deployment costs nearly 100 million USD and requires nearly 39 million USD annually to operate [9]. The cost of deploying an EEW system includes the expenditure of deploying a large number of state-of-the-art seismometers across a vast land mass, which could be economically unviable for many countries and territories [10].

In contrast with expensive high-end solutions, low-cost alternative technological solutions are emerging to create cost-effective EEW systems. Low-cost solutions involve using Internet of Things (IoT) technologies driven by micro-electromechanical systems' (MEMS)-based sensors [11]. Multiple studies, such as those in Sichuan-Yunnan, California, and Taiwan [12–14], have shown the viability and capabilities of MEMS-based sensor networks to provide EEW. Furthermore, with technological advancements, MEMS-based sensors are now embedded with considerable computational and data-processing capabilities, creating opportunities to further innovate and improve EEW systems [11].

Traditionally, the computation and data-processing of EEW systems are carried out at a central location [15]. Data-processing at a central location provides a better control and consistency in data processing; however, it also creates significant bottlenecks and limitations [16]. EEW solutions across the world have recognised the limitations and challenges of centralised processing and invested in redundancy solutions for data transportation and processing [9]. For example, in the US, the Advanced National Seismic System (ANSS) is equipped with three centralised processing units [9]. However, creating such redundancies increase the costs associated with the EEW systems.

In contrast to traditional EEW systems, in this paper, we present a novel approach of using decentralised data processing to generate warnings using a low-cost, MEMS-based sensor network. The paper provides evidence of the performance of this novel approach using data from an experimental network established in Aotearoa New Zealand (NZ).

With the aim of proposing a network architecture that is suitable for an EEW system consisting of MEMS-based, low-cost, ground motion detection sensors and driven by decentralised processing, the remaining sections of this paper are structured to provide the EEW background, including the research gap that this paper tries to address (Section 2), the method used (Section 3), its implementation and results (Section 4), discussion (Section 5), and finally the conclusions of the study (Section 6).

2. Background on Earthquake Early Warning (EEW) Systems

Two basic concepts enable EEW systems. Firstly, information travelling over communication networks moves faster than seismic waves: P-waves (primary or pressure waves) and S-waves (secondary or shear waves) and, secondly, the S-waves produce the most damaging energy in earthquakes, and these arrive later than P-waves [15]. With these concepts, EEW systems use a network of sensors distributed in a geographical area to detect earthquakes and transmit information and warnings in real-time.

Several countries and territories (Japan, Taiwan, Mexico, West Coast USA, etc.) have already implemented EEW systems that can warn the public about oncoming earthquakes for up to tens of seconds before ground-shaking occurs [11]. However, at present, NZ does not have an official, nationwide EEW system [6]. GNS Science—NZ's geological survey, through the GeoNet project, is the official source of earthquake information [6]. GeoNet comprises high-end geophysical instruments, automated software applications, and skilled staff to provide rapid information about earthquakes [17]. GeoNet's current instrument density takes around 60 s to compute a robust earthquake location, and thus is only capable of disseminating information through its website and a mobile phone app after the earthquake occurs [6]. GeoNet, as of writing, is not issuing EEW to the public. However, several other parties are providing EEW services in NZ. In April 2021, Google launched an EEW service to Android-based smartphone users in NZ [18]. This technology uses MEMS sensors embedded in Android phones to detect earthquakes. However, very little information on the Android EEW service is available in the public domain. This is not considered an official emergency alert service in NZ [19]. Aside from Google, other commercial providers offer private in-house EEW services to clients, but these services are unavailable to the general public [6].

Despite the lack of an official EEW system that warns the public, a 2019 survey of the NZ public showed that more than 90% support the use of EEW in NZ [5]. However, the significantly high implementation and operating costs of a countrywide EEW system leads to the question of its economic viability to deploy such a system in a country such as NZ, which has a population of only about 5 million. Hence, there is a need to further explore the development of viable EEW options for NZ [5].

The cost-effectiveness, implementation and maintenance of an EEW system is technologically complex. For example, Japan's EEW system was relatively successful during the 2011 Tohoku earthquake (Mw 9.0); however, the system encountered problems with aftershocks and issued too many warnings, including false alarms [20]. Due to blackout and communication failures, the lack of data from seismometers in disaster areas contributed to the false alarms [20]. A dense network of seismometers is crucial when there are significant lateral variations in the intensity of ground shaking [21]. There are also several other non-technical challenges, such as the minimal time remaining to issue a warning after processing data, public engagement and trust, and a lack of understanding of people's behaviour during earthquakes [22]. From the above-mentioned challenges, the large costs associated with the implementation of EEW have motivated researchers to further explore the feasibility of implementing low-cost EEW solutions.

2.1. Current State-of-the-Art of Low-Cost EEW Solutions

The United States Geological Survey (USGS) classifies modern seismometers as a data acquisition system (DAS), including the seismic sensor, data acquisition unit, and supporting communication hardware. Based on their performance, DAS are categorised into Class A, B, C, and D instruments [23]. Class A instruments are near state-of-the-art seismometers, while MEMS-based sensors such as Raspberry Shake (RS) are identified as Class C instruments [23,24]. Low-cost EEW solutions, such as those using IoT technologies driven by Class C, MEMS-based sensors, are emerging to bridge the economic and technological gaps [25].

Over the last ten years, advances in MEMS-based technologies have driven down the costs of ground motion sensors [26], while traditional seismic sensors cost tens of thousands of dollars [27]. Declining sensor costs and internet ubiquity are the main drivers of IoT technology [27,28]. The ubiquity of internet connection allows for the implementation of interconnected networks. With the enhanced cloud services, a high amount of sensor data can now be processed to generate warnings and alerts. The growth in this technological advancement should make our cities safer [29].

Such IoT technologies include several low-cost, MEMS-based earthquake sensor devices such as RS [30], P-Alert [14], and Grillo [11,31], each of which is capable of providing complementary and alternative earthquake detection and warning solutions. These low-cost MEMS, with a maximum resolution of 16-bits, can detect earthquakes tens of kilometres away [13]. Dense networks of these low-cost Class-C MEMS accelerometers have been used in seismological investigations, structural health monitoring, and EEW applications [12,13,32]. The P-Alert network in Taiwan, for example, uses a dense array of MEMS accelerometers that form a network of mini-arrays to record from moderate to large strong-motion events [11,14]. P-Alert technology is also used to monitor the Himalayan fault in India and the Apennines Fault in Italy [33,34]. These sensors are vertically mounted on building walls to study how the sensor and structure interactions between the acquired acceleration data can aid in the production of high-quality shake maps. In addition to this, as one of the work packages of the TURNkey European Union project, the University of Iceland has implemented an EEW sensor network using low-cost, MEMS-based RS sensors, where the processing of ground motion data is conducted at a central location [35]. Similarly, the researchers who conducted an EEW project, namely, Community Seismic Network, have designed a sensor package consisting of a three-axis Class-C MEMS accelerometer; these sensor packages are located in buildings, and data processing is carried out centrally at a cloud-based server to provide warnings [36].

The MEMS-based sensor networks are emerging as a solution to several EEW challenges inherent in the architecture of EEW networks [37]. Two key challenges for EEW systems are: (i) when earthquakes occur at the edge or outside the seismic network, and (ii) where the sensor density of the network is lower, thus compromising the required azimuthal coverage [27]. In such situations, it is common for EEW systems to have poor earthquake location estimates and significant alert delays due to their restricted azimuthal coverage and the time needed for the wavefront to reach the required number of sensors and generate an alert [27]. The array-based approach consisting of MEMS-based accelerometers has demonstrated improvements in EEW capabilities; this can be seen in cases such as those in Southern California [13], Northern India [38], Central Italy [39], Taiwan [25,40] and the Sichuan-Yunnan Province [12].

Off-the-shelf sensors similar to those seen in smartphones are also available as scalable opportunistic sensor nodes [41]. Phone-based sensing is possible because almost all smartphones are equipped with MEMS-based accelerometers. These motion sensors can be programmed to work as seismometers, detecting the shaking generated by earthquakes. Although using smartphones is an interesting opportunity, it poses challenges in the production of reliable EEW signals [42]. Additionally, using smartphones only for EEW purposes will not be cost-effective, as smartphones come with powerful CPUs, plentiful memory, and other auxiliary sensors [43]. Lee and colleagues [43] proposed a stand-alone earthquake detection and alerting device for homes in South Korea that can send alerts to nearby devices (e.g., smartphones and TV). As an alternative to a smartphone-based network or stand-alone device, Boxberger and colleagues have proposed an innovative instrumental design called a multi-parameter wireless sensing system that utilises and incorporates off-the-shelf components to analyse ground motion data and issue EEW [44]. Another system, run by Grillo—an independent, private-sector EEW network—uses MEMS accelerometers operating as an IoT and cloud platform, successfully providing earthquake alerts in several high-intensity earthquakes in Mexico [11,31]. Further, in 2012, Fischer and colleagues [15] reported on the testing of a wireless decentralised WLAN-based mesh network for detecting earthquakes using custom-made, low-cost sensors [45]. This is the only research found in the literature which has clearly taken an approach to detect and process data at the node-level. In this project, the approach adopted for communication between sensors limits the distance between the sensor nodes, and is in the range of 200–1000 m. Although the authors claimed that the sensors are low-cost, there is no indication of the actual cost of developing a sensor. With the limitation that two sensing nodes have to be

installed very close to each other, this type of EEW system may require a significantly larger number of sensors in a real-life scenario, which may offset its benefits.

Other promising IoT-enabled solutions that are readily available to citizens include BRINCO and BRCK [46]. BRINCO is the first IoT-enabled alarm that can warn users in the personal-aware mode; it sends ground motion data to a private cloud-based data centre and assimilates the information with other seismic networks [46]. BRCK is a versatile IoT-enabled device for use in areas with poor infrastructure, as it can utilise 2G communication, which is ideal for deployment in disaster zones [46]. Such IoT-enabled solutions are integrated with a cloud service for back-end analytics using machine learning and artificial intelligence techniques [46]. While the above low-cost EEW approaches started to show promising results, the affordability of such systems has encouraged the development of systems with increased community participation and engagement.

There are initiatives exploring community and participatory seismic sensing (i.e., citizen science techniques) to improve seismology and EEW [47]. For example, a prototype MacBook-based EEW system called MacSeisApp utilises the sudden motion sensors in people's laptops to detect seismic activities; it utilises Apple's Push Notifications via a dedicated server to produce the earthquake notification [48]. The Quake-Catcher Network offers a similar architecture to Apple—a distributed computing seismic network that uses low-cost accelerometers that connect to laptops to record earthquakes [49,50]. Quake-Catcher sensors are installed in the premises of volunteer host participants (e.g., schools, homes, and businesses); these participant sensor hosts exist throughout the world in Chile, Mexico, Taiwan, New Zealand, and other countries [49,50]. Low-cost, MEMS-based networks such as Quake-Catcher could potentially be integrated as complementary systems to traditional seismic networks [50].

An innovative, real-time, citizen-engaged network of mobiles phones was implemented by Zambrano and colleagues [51], considering the coupling features of the geographical zone and time and spatial analysis with crowdsourced data. It offered a precise and customisable architecture for the improvement in the delivery of notifications in and around the epicentre, and demonstrated a reduction in the number of false alarms [51]. Further, in 2015, Minson and colleagues [52] constructed a smartphone-based EEW and conducted an evaluation through controlled tests using simulated data for an Mw 7.0 Hayward fault earthquake in California and actual data from the Mw 9.0 Tohoku earthquake. With embedded MEMS-based motion sensors, smartphones have become a potential tool for consideration in the development of crowdsourced EEW applications. An ongoing, low-cost, crowdsourced, smartphone app-based EEW solution is the MyShake Platform; since its release in February 2016, more than 300,000 people globally have downloaded the app, providing EEW networks in different parts of the world within a short period of time [53]. Similarly, a network of 82 smartphones fixated in buildings in Costa Rica proposed an EEW with a significantly lower cost than a scientific-grade network [41]. Another crowdsourced EEW system is the SeismoCloud App, which shows promise in delivering EEW to users in a region [54]. In addition, in April 2021, Google launched an earthquake alert service to its Android phone users in NZ and Greece [18]. For the Android alerting services, when a phone detects an earthquake, it sends data, including the location details, for processing at a centralised server, and confirms them with shaking detected from hundreds of phones [18].

The above-described EEW solutions incorporate low-cost alternatives by using off-the-shelf components and devices, while some of the solutions have incorporated community and participatory seismic sensing. These low-cost solutions showcased their capability to accomplish complex information-processing activities at the sensor node-level, providing an opportunity to create networks that are driven by robust and faster communications and network topologies [55,56].

2.2. Research Gap

Modern IoT networks often operate in a three-layer architecture consisting of the cloud layer, fog layer, and edge layer [57]. Generally, the cloud layer contains a collection of servers capable of high-performance processing and storage, whereas the fog layer is often identified as an intermediate layer capable of reducing the workload of the devices at the cloud and edge layers [57]. The edge layer often contains devices such as sensor nodes, which are capable of acquiring data. Figure 1 shows these layers as applied to three different architectures.
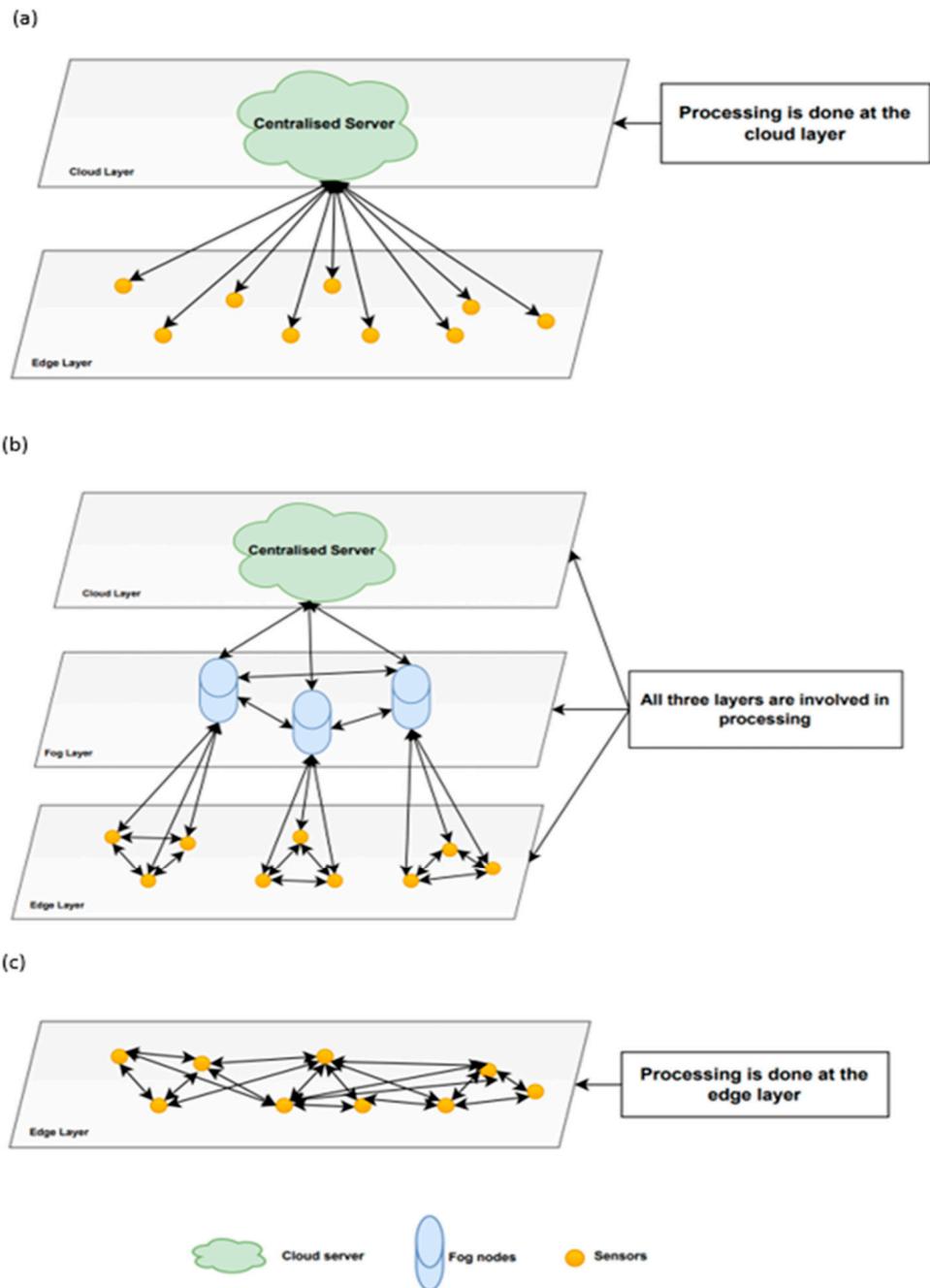
Figure 1. Illustrating IoT architectures and their layers: (**a**) centralised architecture, (**b**) typical decentralised architecture, and (**c**) proposed decentralised architecture.

As shown in Figure 1a, traditionally, IoT-based networks consist of a large number of sensor nodes at the edge layer, which transmit the sensed data to servers located at the cloud layer for processing and storage. In the literature, this type of IoT sensor network is defined as a centralised IoT network [58]. However, with the development of smart devices with enhanced data processing and storage capabilities for a lesser cost, data processing and storage have shifted from the cloud layer to the fog and the edge layers. This type of network is identified as a decentralised IoT network [58]. When it comes to decentralised IoT networks, the processing that occurs at the cloud and other layers can vary depending on the context of the application and the capabilities and capacities of the devices used at the fog and edge layers. For example, some networks use devices attached to both fog and edge layers, or devices attached to all three layers (Figure 1b), whereas some architectures are capable of accomplishing their entire processing or storage needs only at the edge-layer devices, such as sensor nodes (Figure 1c). Enhanced performance of both the fog and edge layer devices have created opportunities for decentralised processing and storage more realistic with the ability to achieve lower latencies, lower operational costs and increased network redundancy. However, when it comes to EEW networks, whether it is high-end sensor networks or low-cost MEMS-based networks, data processing in EEW systems was traditionally carried out at the cloud layer, and the warning sent out as push notifications to the end-users [41,59]. Data-processing at a central location provides the advantage of better control in the detection, collection and processing of data during a disaster and immediately afterwards [16]. However, processing data centrally also creates several technological bottlenecks and limitations. One of the main limitations is the risk of disruption in data collection to a central location due to the impacts on the telecommunication infrastructure after a big earthquake [60]. Data-processing centres, intermediate data-collection centres, and infrastructures to transport data may be severely disrupted or destroyed after an earthquake [61]. Unavailability of the central processing capability due to the loss of connectivity may hinder the issuing of time-critical warnings to end-users. EEW systems have recognised the limitations of centralised processing and, therefore, invested in redundancy solutions for data transportation and processing [9].

Modern-day, MEMS-based, ground motion detection sensors have significant processing capabilities, and they possess the ability to run ground motion algorithms and data-processing at the sensor node itself [11]. Improvements in technological capabilities have created an opportunity to explore how these sensors can be utilised to reduce the amount of centralised processing for ground motion data and the possibility of generating node-level alerts that are sharable among the sensor nodes and other connected devices. This approach could reduce the cost of earthquake solutions. A node-level approach may also increase the resilience of systems as it provides the possibility of alerting end-users at regional and local levels, despite infrastructure failures occurring in parts of the system. However, at present, most of the MEMS-based EEW systems either work as isolated private solutions or are inhibited by telecommunication latencies and centralised processing bottlenecks [11]. Except for the experimental EEW system based on custom-made sensors by Fischer and colleagues [15], there is limited evidence in the existing literature regarding detecting and issuing alerts in a more decentralised manner, where detection algorithms are run at the node-level of a MEMS-based sensor network to become self-configurable and self-healing. The approach proposed by Fischer and colleagues [15] seems capable of detecting earthquakes and processing data mostly at the node-level. However, there is no clear discussion regarding the robustness of the system, with only limited testing and evaluation having been completed. Further, they did not attempt to compare the performance of their approach with the traditional centralised EEW networks [15].

3. Methods

With the above-mentioned gaps in the EEW literature, we propose a fully decentralised, community-engaged, MEMS-based, EEW sensor network architecture. Similar to Figure 1c, the proposed architecture processes ground motion data exclusively at the edge layer using the sensors. This paper focuses on designing the above-defined EEW sensor network and discusses its step-by-step development. The proposed architecture fully operates at the edge layer, consisting of low-cost MEMS-based sensors capable of successfully running a reliable earthquake detection algorithm within the sensor node and securely exchanging ground motion and earthquake alert data with other nodes. These MEMS-based sensors are hosted and operated by individual citizens and community groups in NZ, where the sensors are installed in a private home, or a property owned by a particular community or a group.

The implementation of an appropriate and feasible community-based MEMS sensor network started with: (1) exploring different wide area network (WAN) topologies, followed by (2) selecting a software-defined WAN (SD-WAN) solution for a community-engaged EEW sensor network, (3) selecting appropriate sensors, and finally (4) selecting an appropriate EEW algorithm.

3.1. Explore Appropriate WAN Topology

Establishing secure communication links between the sensors installed in (1) different private local area networks, (2) supported by various internet service providers, and (3) located in different geographical locations is a key challenge in building a community-led low-cost sensor network. Therefore, it is important to investigate the available WAN technologies as possible solutions. Ideally, the WAN topology should support the building of a sensor network where the sensors are owned and hosted by the community members. The securely connected MEMS-based sensors communicate with each other to exchange data while installed at different geographical locations attached to private household computer networks. The following subsections explore potential WAN approaches that are suitable for implementation of the proposed EEW network.

3.1.1. Virtual Private Networks (VPN)

When it comes to exploring WAN approaches, Virtual Private Network (VPN) is one of the well-known, capable of creating secure networks, connecting devices in geographically dispersed locations attached to privately owned networks [62]. While VPNs are popular among business organisations, they are resource-intense and complex to set up and implement [63]. These issues are particularly relevant when designing and implementing citizen science or crowdsourced network, which consists of low-cost, MEMS-based sensors where end-users do not belong to a particular organisation or consortium. The end-users of this type of system are going to be members of the general public. However, they do not have any direct relationship or collaboration with each other, except that their host devices are connected and communicate with each other to form a network of ground-motion sensors to provide a public service by generating warnings. Therefore, VPN may not be suitable for building a community-engaged, MEMS-based EEW network, given the associated complexities in ownership and costs. However, another option to explore is the SD-WAN as an appropriate networking method to develop the proposed, community-engaged network.

3.1.2. Software-Defined WAN (SD-WAN)

SD-WAN is a virtual software-based WAN management approach. SD-WAN benefits include: (a) cost or price advantages with freedom of transportation across different communication infrastructure and technologies, (b) ability to enhance the performance of the applications and their agility, (c) appropriateness for software-as-a-service and cloud-based applications and (d) easy-to-operate, automated environment supported by cloud-based management [64].

Compared with VPN, SD-WAN virtualisation technology [65] can be considered more appropriate for citizen-engaged or crowdsourced networks. SD-WAN automatically repairs any outages occurring across the network nodes. Therefore, anyone can easily connect a sensor to the network, as SD-WAN offers "self-healing" capabilities. SD-WAN can also ensure automatic alignment while the network topology changes [66]. Due to SD-WAN's granular level of support and promising technology features, such as caching or application acceleration, devices located at any type of private network (homes, cafes, libraries, community centres, etc.) can readily maintain the connectivity. SD-WAN-based networks can readily allow for individuals or organisations to connect their devices and communicate using any internet connection as a specific communication infrastructure is not required. The flexibility offered by the SD-WAN-based networks makes it a more appropriate approach to implementing the proposed, community-engaged, MEMS-based network.

3.2. Selection of SD-WAN Solution for a Community-Engaged EEW Sensor Network

The proposed EEW network sensors are expected to be installed at community members' homes. Therefore, after its initial configuration, the sensors should easily connect to any household internet connection and switch networks; ideally, they should function as a plug-and-play. Therefore, this architecture should be designed with the ability to penetrate firewalls and a Network Address Translation (NAT), equipped with self-healing ability during a loss of connection, and be able to dynamically self-configure to manage varying conditions of the network address (as there is a possibility that sensors may change locations for multiple reasons). Ideally, the connection should be capable of being verified and encrypted to mitigate hackers breaking the network defences.

3.2.1. Hole Punching to Overcome Network Address Translation (NAT) Challenges

Network Address Translation (NAT) imposes common challenges in peer-to-peer (P2P) communication [64]. This is primarily because peer nodes may not be able to reach any form of globally valid IP address [64]. Although there are known NAT traversal techniques, there is minimal literature about them and their functionality. Further, there is only minimal information to justify their efficiency and effectiveness [64]. However, the technique commonly known as "hole punching" is considered one of the least complicated, yet more robust and effective NAT traversal techniques, where it simply creates a tunnel for reliable communication between two communicating entities, irrespective of the communication protocol being used. According to Reference [64], about 82% of the NAT techniques support hole punching to work with User Datagram Protocol (UDP), while about 64% support hole punching to work with Transmission Control Protocol (TCP). The use of NAT applications has become crucial for various peer-to-peer applications such as online gaming and voice over IP (VOIP) protocols. Devices attached to networks with public IP addresses should communicate with each other very easily. Additionally, the clients with private addresses could be able to connect to a public server without any difficulty if they can initiate a connection while behind a router or firewall. However, for hole-punching to operate properly, it is essential to form direct communication between two devices installed behind routers or firewalls that use NAT.

Ford and colleagues initially explored the use and application of hole-punching [67], and it has gained considerable attention in the peer-to-peer application communities. Furthermore, the specifications of several experimental level protocols, including STUN, ICE, and Teredo, have documented the key features of the UDP-based hole-punching techniques [67]. However, there are no published works that thoroughly analyse the advantages and disadvantages of hole-punching for multi-level NAT.

3.2.2. Selection of Appropriate UDP Hole-Punching SD-WAN Solution

We considered several popular hole-punching solutions, including WireGuard, Openvpn, Nebula, ZeroTier and Tailscale, to identify the most appropriate to build the community EEW network [68]. In this process, we considered factors such as scalability, flexibility,

cost and security. Having considered the pros and cons of the available solutions with regard to NAT-related challenges, we selected ZeroTier as the most appropriate SD-WAN hole-punching solution to implement the intended, community-engaged, MEMS-based EEW network. According to Goethals and colleagues [68], the response time, failure rate, packet loss, and latency of ZeroTier are in the acceptable range to build a proposed community-engaged SD-WAN solution. ZeroTier is capable of reducing the network management complexities by combining the strengths of both the VPN and SD-WAN approaches. ZeroTier's software solution allows for devices, applications, and services to be connected simply and securely, regardless of their physical location [69]. This can be used in various use cases, including VPN, multi/hybrid-cloud, SD-WAN, peer-to-peer networking, and IoT remote access, allowing all these things to be achieved with a single system, leading to vastly reduced complexity and associated costs.

ZeroTier's UDP hole-punching mechanism allows for connecting devices to communicate with the use of any type of data packet including TCP, UDP, etc. Basically, it captures data packets from the sender and transmits them to the receiving end through the transmission tunnel created with the UDP hole-punching mechanism. However, similar to any other solution, it comes with its own limitations. Like any other proprietary hole-punching technology solution in the market, it comes with a cost. However, ZeroTier is more affordable than most similar tools available on the market. It is especially appropriate for the experimental type of network as it offers a free service for networks of fewer than 50 devices. Although ZeroTier ensures scalable end-to-end security using 256-bit encryption, it can create vulnerabilities, especially with the type of devices used for the proposed work and hence needs further attention regarding security measures.

The decision to use ZeroTier as an appropriate SD-WAN solution to implement in the proposed network has provided the ability to assign a unique IP address for each sensor when it is connected to the EEW network. As the MEMS-based sensors are to be installed in people's homes in a citizen science environment, having individual IP addresses for each sensor will help connect sensors to the proposed, highly flexible and ad-hoc sensor network without a dedicated network infrastructure.

3.3. Selection of Suitable Sensors for the Proposed MEMS-Based EEW Network

Having selected the ZeroTier as an appropriate SD-WAN solution to build the proposed EEW sensor network, the next crucial step is to decide on the type of MEMS-based ground motion sensor or sensors to implement the proposed sensor network. In this process, we evaluated several low-cost MEMS-based ground motion sensors that are available on the market. The selection criteria of MEMS sensors include node-level data-processing ability, the memory of the sensor, accessibility, accuracy of the sensor data and price of the sensor. These selection factors are crucial as the proposed network not only captures the ground motion data from the sensors but also processes the data within the sensors, as opposed to other traditional EEW systems where sensors are only used to capture the data. The sensors evaluated for selection include popular off-the-shelf MEMS-based ground motion sensors on the market, such as those from P-Alert [14], Canterbury Seismic Instruments (CSI) [70], Grillo [11,31], and RS [30].

Having compared the sensors against the above-described criteria, we selected the RS sensor to build our experimental network. Its openness to access and relatively superior processing ability led us to select the RS sensor. Comparatively, other sensors provide limited direct access to the sensor and data and the limited processing capacity at the sensor node, which must be resolved before connecting to the proposed network. At present, we are in the process of engaging with sensor manufacturers who have developed other popular, low-cost, MEMS-based sensors in the market with the aim of resolving some of the above-mentioned limitations. It is expected that removing the above-described limitations will create future opportunities to integrate multiple types of MEMS-based sensors into the proposed network rather than depending solely on RS sensors.

Several different types of RS sensors are currently available on the market, namely RS1D, RS3D, RS4D and RSBOOM, RSSHAKE&BOOM. Even though the listed RS sensors are equipped with different sensing capabilities, all of them are powered by Raspberry Pi 3 Model B with a Broadcom BCM2837 $4 \times$ ARM Cortex-A53 (1.2 GHz) processor with 1GB LPDDR2 memory. We implemented our experimental RS EEW network by adding different types of RS sensors but predominantly the RS4D, which consists of a geophone and triaxial C-class ground-motion accelerometer.

3.4. Selection of an Appropriate EEW Algorithm

Having identified the appropriate SD-WAN networking approach, followed by the type of MEMS-based sensor, the next crucial step is to identify an appropriate EEW algorithm that can successfully be implemented within the RS sensor node, which is constrained by its processor's computational and processing capabilities. With the aim of identifying the best-suited detection algorithm, several popular EEW algorithms were compared and evaluated. The complexity of the algorithm and the level of the computational capacity of the sensors are two main factors that were considered during the selection of the detection algorithm.

At present, a variety of EEW approaches have been used across different parts of the world [11]. These approaches can broadly be classified into three categories: (i) the source characterisation, (ii) ground-motion-based EEW and the (iii) on-site prediction approaches [59,71,72]. These three approaches have several strengths and weaknesses depending on the EEW implementation environment [55].

There are notable drawbacks to consider when selecting an EEW method based on source characterisation. These drawbacks include missing earthquake detection during extreme seismic activities, underprediction of large strong-motion earthquakes with finite faults, and overprediction of large, multiple, simultaneous earthquakes [73]. When it comes to on-site prediction approaches, predicting the intensity of the S-wave with the P-wave's intensity using a single station can lead to inaccurate results [11]. In contrast, ground-motion-based EEW approaches showed promising and accurate results [71]. One of the newer ground motion-based algorithms: the PLUM algorithm by Kodera, has become popular due to its robustness, lightweight design and easy-to-implement nature [71]. The PLUM algorithm has already been implemented in Japan as part of their EEW method [71]. EEW researchers in the West Coast USA are also studying the PLUM algorithm to integrate it into their sensing and alerting system [74]. Considering these implementations, supported by their robustness and ability to perform well in resource-constrained environments, our research has chosen the PLUM algorithm as the most appropriate EEW algorithm to implement the proposed, decentralised, MEMS-based EEW architecture.

4. Implementation and Results

4.1. Implementation of a Raspberry Shake (RS) Sensor Network

We have deployed nearly 25 sensors across five regions in NZ's North Island: Auckland, Rotorua, Palmerston North, Wairarapa, and Wellington. While our network currently consists of different types of RS sensors, the experiments and evaluations conducted in this paper used RS4D sensors. Figure 2 shows an example of the proposed network with RS4D sensors (S1, S2, . . . Sn), where sensors are installed at the homes of the general public attached to the home local area network. In this arrangement, data processing entirely occurs at the sensor node, and the communication takes place directly between the sensors without any centralised servers.
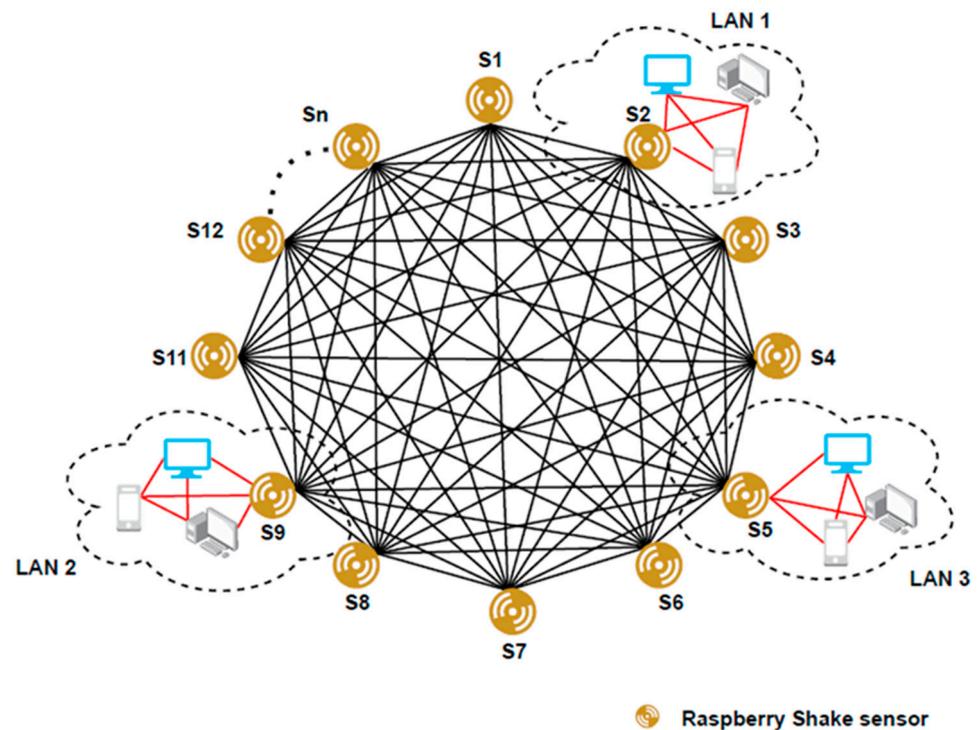
Figure 2. The proposed decentralised architecture consists of Raspberry Shake sensors installed in different Local Area Networks, which are connected to each other via the internet.

4.1.1. Implementation of Appropriate Security Measures for RS Sensor Network

Having selected an RS-based EEW sensor network implemented on a ZeroTier SD-WAN platform, making the sensor network secure is considered the next most important step in the process of building the proposed EEW network. Security becomes paramount, especially because the proposed network consists of RS sensors located at the homes and premises of the general public and attached to private home networks. Even though ZeroTier provides strong end-to-end data security between the low-cost, MEMS-based devices, the proposed node-level processing architecture could still create opportunities for potential security breaches.

Such security breaches can become harmful since sensors are located in different geographical locations, attached to different home-based private networks. It was found that most of the potential security breaches that could be anticipated in the proposed architecture may primarily occur by directly or indirectly accessing the sensor through outside networks [75]. Successful login to a sensor by a third party could allow a user to (a) gain admin access [75], (b) find IP addresses of other sensors through brute-forcing [76], (c) access the source files, (d) remote login into another sensor [77], and (e) sneak out to an outside network [78]. Figure 3 shows the interconnectivity of the above-mentioned threats.

In terms of security measures, several protection mechanisms were implemented at each sensor to increase security at each level in the above diagram. Direct login access was restricted by introducing security configurations to the user account in each sensor. In addition, all possible IO ports, such as USB, serial, i2c, HDMI, etc., were disabled. Furthermore, remote login to the sensors was restricted so that it can only be carried out through the admin's centralised server. With that, remote login to the sensor from the centralised server is only permitted with a Secure Shell (ssh) key, which is pre-generated and saved inside both server and sensor such that each sensor could have a unique ssh key. In addition, incoming ping requests to all sensors were restricted to reduce the risk of automated hacking tools. Due to security breaches in open network ports, all the unused open network ports were disabled by configuring a network firewall. Moreover, we reduced the number of incorrect login attempts and banned the user IP for a considerable amount of

time in case of multiple incorrect login attempts. To protect the source code, we encrypted all the files used for communication purposes with AES-256 encryption [79], which makes it harder to decrypt the source code.
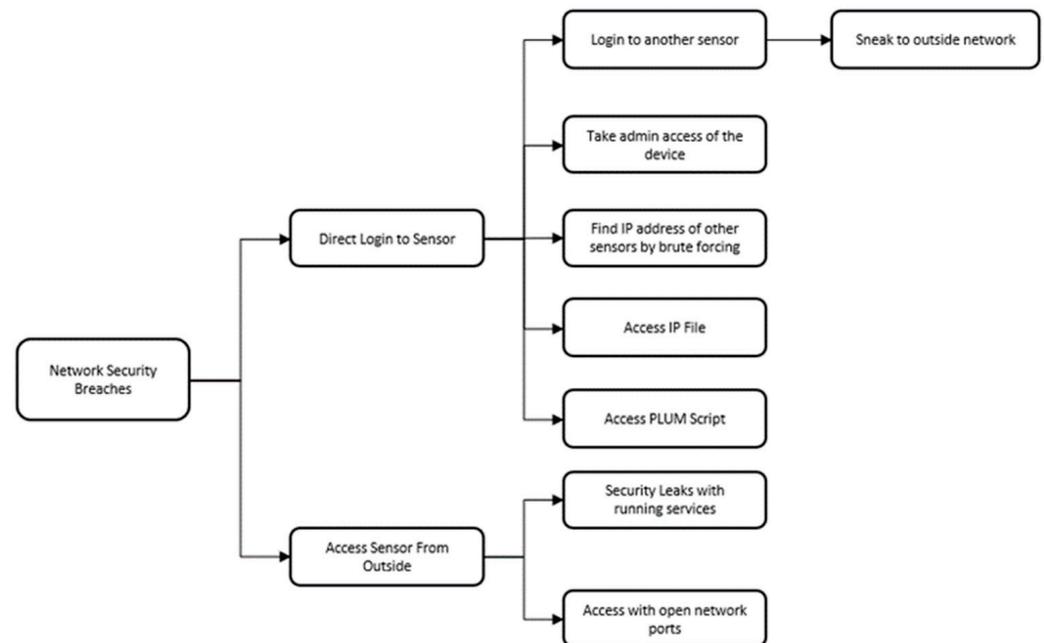
Figure 3. Different types of possible security breaches in the proposed network architecture.

To ensure the security of the proposed community-engaged architecture, we installed the above-mentioned security measures to selected RS4D sensors. After having installed the software to enhance the security, the vulnerability of the network to security breaches was tested using a separate RS4D sensor environment dedicated to testing the performance of newly installed security features. These tests were conducted before the sensors were connected to the proposed EEW sensor network. As a first step, we used a network vulnerability scanner (Nmap) [80] and a password cracker (brute force attack generator) [81] to assess the vulnerability of the network and ensure a proper configuration. The vulnerability scanner was only able to discover the expected open network ports of the RS4D sensors from all the available network ports (1000 ports). Further, the password cracker tool was not able to log in from one sensor to another. These outcomes can be considered a confirmation of the enhanced security of the network due to the newly implemented security measures.

4.2. Implementation of Modified PLUM EEW Algorithm within RS Sensor Environment

Having introduced measures to secure the node-level sensor environment running on a ZeroTier SD-WAN solution, the next crucial step is to implement the PLUM algorithm within the RS sensor nodes to detect the earthquakes.

When implemented with the RS4D sensor, the PLUM algorithm should be able to use the real-time seismic data captured by the sensors to predict the seismic intensities at the given prediction points within an area of a 30 km radius [71].

As shown in Figure 4, the prediction process for a given location (red star) takes place in a circle with a 30 km radius around an observation data point (blue dot). While the PLUM algorithm operates with a single operating point, this approach was subsequently criticised for the possibility of false or missed alerts and to terminate the propagation of the alert [82]. This is particularly relevant for the proposed type of EEW network, consisting of low-cost sensors installed in the homes of the general public, which are often in noisy environments compared to the high-end EEW networks consisting of sensors installed in much more noise-controlled environments. Therefore, it is crucial to include extra observation points in the system to avoid false or missed alerts and terminate the

propagation of a false alert [82]. Accordingly, to minimise the anticipated false alarms by having only a single observation station, we made a few modifications to Kodera's original PLUM algorithm. In our modified approach, we only use the PLUM approach to predict the seismic intensity of a station using the observed real-time intensity at an observation station. Rather than taking the maximum observed seismic intensity concept to predict the station intensity [71], we are directly assigned the real-time seismic intensity of the observation station to the prediction station. In this process, to reduce the number of false alerts, we introduced a two-station trigger concept to the PLUM algorithm, as proposed by Cochran and colleagues [74]. This modified PLUM algorithm triggering occurs when the predicted seismic intensity exceeds a predefined threshold at a particular sensor. Then, the triggered sensor will check whether any of the neighbouring sensors has experienced a ground motion intensity above the same threshold within a defined waiting time. An alert will only be issued if these conditions are met. Otherwise, the system will terminate the alert. As reconfirmed in a more recent EEW study in the US [82], this modified PLUM approach with a two-station trigger concept is expected to reduce the number of false alerts and increase the accuracy of the alert generation.
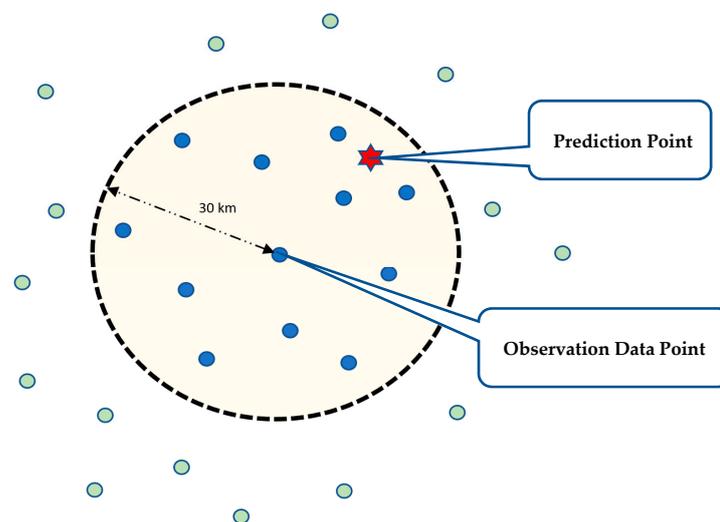
Figure 4. A scenario illustrating how the prediction process of the PLUM ground motion detection algorithm occurs within a circular region of 30 km radius (The red star represents a prediction point and blue dots are the potential observation data points whereas green dots are observation points outside the 30 km range and not involved in the detection process in this particular scenario).

4.3. Defining and Calculating System Latency for the Proposed EEW Architecture

As shown in Equation (1), The system latency of the proposed EEW network ($\delta t_{sys\_latency}$), can be defined as the sum of three components δt_{detect} (detection time), $\delta t_{transmission}$ (transmission time) and $\delta t_{s\text{-wave\_travel}}$ (S-wave travel time).

Equation (1) proposed a formula for EEW system latency:

$$\delta t_{sys\_latency} = \delta t_{detect} + \delta t_{transmission} + \delta t_{s-wave\_travel} \tag{1}$$

4.3.1. Transmission Delay ($\delta t_{transmission}$)

This research defines the transmission delay ($\delta t_{transmission}$) as the time taken to transmit the data between two sensors within an area of a 30 km radius, and it is only dependent on telecommunication factors; it is independent of the earthquake characteristics. Furthermore, the transmission delay ($\delta t_{transmission}$) was measured, along with different communication protocols to identify which protocol could minimise the transmission delay.

4.3.2. Detection Time (δt_{detect})

Detection time(δt_{detect}) is the time taken for a particular detection algorithm to calculate and issue an alert of an earthquake when seismic intensity at a sensor goes above the predefined shaking intensity threshold [41]. Therefore, δt_{detect} depends on the type of the algorithm and its complexity. Furthermore, δt_{detect} also depends on the specifications of the ground motion detection sensor itself (in this research, RS). This includes the speed of the processor, size of the RAM, the capacity and the speed of the storage, where these factors determine the speed of execution of the algorithm at the node-level.

4.3.3. S-wave Travel Time ($\delta t_{s\text{-}wave\_travel}$)

With the multi-station trigger concept, S-wave travel time ($\delta t_{s\text{-}wave\_travel}$) is defined as the time taken for the S-wave to travel from the first observation sensor to the farthest observation sensor from the epicentre. For different earthquakes, the S-wave travel time will differ according to the sensor geolocations and the number of sensors used to trigger the alert (e.g., two sensors for our approach). As mentioned in the PLUM algorithm section, using the low-cost EEW sensor network proposed in this study, the use of two observation stations to trigger an alert is essential to reduce the number of false alerts [82]. Due to the two-station trigger concept, even though the first sensor in the network detects the earthquake in a fraction of time, it needs to wait for confirmation from the 2nd observation sensor. This makes the S-wave travel time between the 1st and the 2nd observation sensor ($\delta t_{s\text{-}wave\_travel}$) a significant component of determining the system latency. For our experiments, we assumed that the earthquakes are shallow and the generated S-waves propagate horizontally as a plain wavefront with a constant speed of 3 km/s [71].

4.3.4. System Latency Calculation Scenarios

To test the performance of the proposed decentralised EEW sensor architecture as well as to obtain its $\delta t_{sys\_latency}$, we calculated δt_{detect}, $\delta t_{s\text{-}wave\_travel}$ and $\delta t_{transmission}$ separately. As the first step, we calculated the transmission delay ($\delta t_{transmission}$), followed by the detection time (δt_{detect}), and finally the S-wave travel time ($\delta t_{s\text{-}wave\_travel}$).

For the latency calculations, we deployed a ground motion data simulator inside each RS ground motion detection sensor, capable of accurately simulating the ground-shaking captured by the RS MEMS accelerometers. In this process, we created six hypothetical shallow earthquake scenarios with their corresponding S-wave arrival from various azimuthal directions towards the installed sensors in Wellington. This paper presents the results obtained from the hypothetical earthquake scenarios that originated from the regions that were anticipated to create ground shaking in the Wellington region. One of the six scenarios was from the direction of Hawke's Bay region, another one from the Tūrangi region, three from the Cook Strait, and the final one from the South Wairarapa region.

In terms of installing sensors, even though we planned to utilise a higher number of sensors located in the Wellington region, our sensor installation efforts in the community were disrupted due to the COVID-19 lockdown. This led to a limit on the number of sensors that were available for the experiments. At the time of conducting the latency experiment, as shown in Figure 5, there were only five sensors available in the Wellington region, which was considered sufficient to conduct the experiments.

As the next step, for each sensor, time-stamped ground-motion datasets with the calculated S-wave arrival time due to a particular earthquake scenario were programmed into the data simulator installed within the sensor. Thereafter, the pre-programmed data simulator in each sensor were triggered at the exact time when the S-wave (with the velocity of 3 km/s) of each earthquake scenario was expected to reach that particular sensor. For the simulations, we assumed that the earthquake waves travelled as circular waves [41]. For the hypothetical earthquake dataset, we used a ground motion dataset captured by our RS sensor network. Each earthquake scenario was repeated ten times, leading to a total of 60 simulations. After running ten simulations for each scenario, the range of the values obtained for the system latency for all the investigated protocols remained negligible and

was in the milliseconds range. Therefore, we decided that it is reasonable to report the average findings obtained for the latency figures for each scenario.
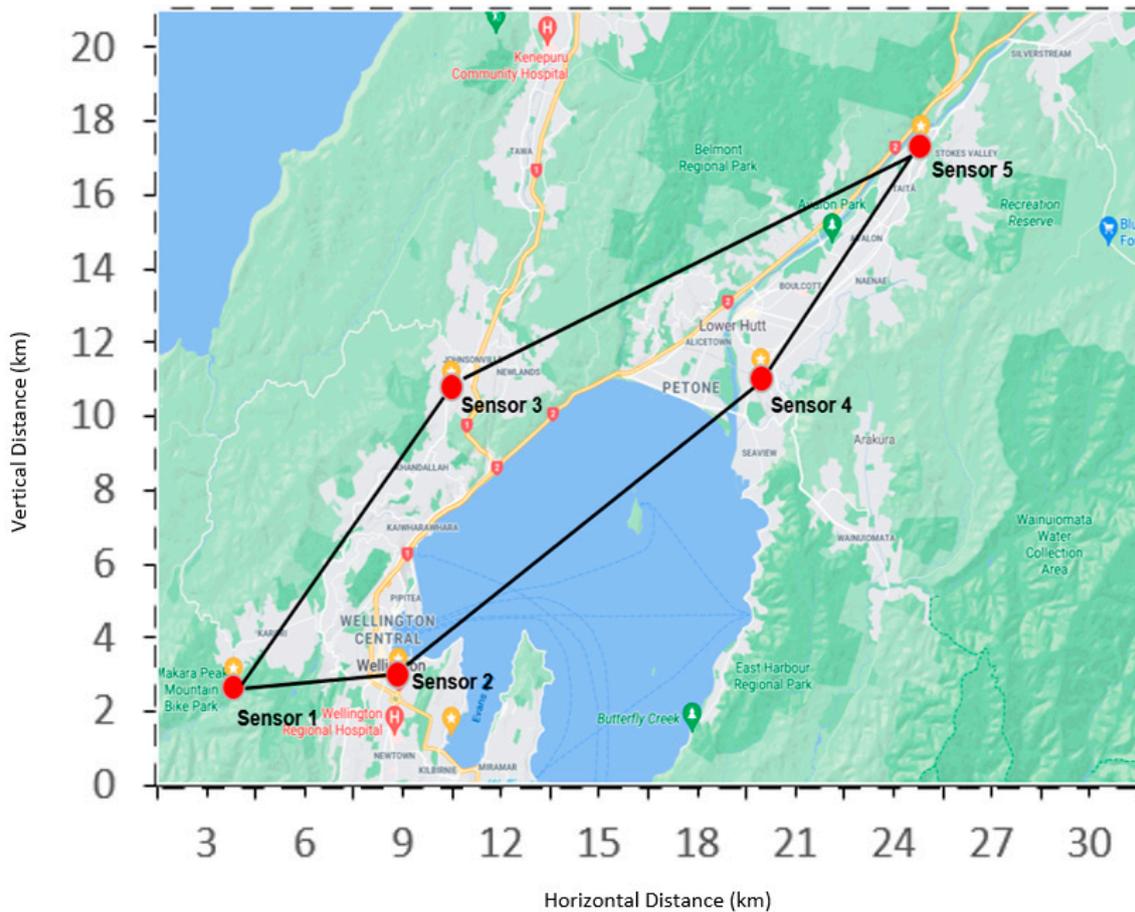

Figure 5. Installed Raspberry Shake sensors in the Wellington region.

Calculating the Transmission Delay of the Network ($\delta t_{\text{transmission}}$)

With the hypothetical simulated environment, we analysed and compared different standard communication protocols that support the implementation of IoT-based networks to select the protocol that provides the minimum value for the $\delta t_{\text{transmission}}$. This is important, as having a minimum value for the transmission delay is desirable for a time-critical application such as EEW.

At present, several popular communication protocols support the implementation of IoT-based networks [83]. Among them, Message Queuing Telemetry Transport (MQTT) is an application layer protocol based on TCP/IP with a publish and subscribe data transfer mechanism [84]. In comparison, both the TCP and UDP are standard communication protocols without any application layer [85]. These three protocols were evaluated to find the most suitable primary communication protocol for the proposed EEW network. By evaluating the standard communication protocols, TCP and UDP, we defined an application layer where client–server communication can take place.

Having recognised the potential use of the above-mentioned protocols for communication, we compared the transmission delay ($\delta t_{\text{transmission}}$) obtained for each of the protocols. During this comparison, TCP, UDP, and MQTT protocols were tested for the end-to-end transmission delay between two sensors for a cycle of 24 h.

Figure 6a shows the average transmission delay recorded for each hour for a 24-h period for the UDP and TCP protocols, while Figure 6b shows this for the MQTT. When comparing the average transmission delay for a day for all three protocols, UDP outperformed the TCP

and MQTT, with an average $\delta t_{\text{transmission}}$ of 16.10 ms, whereas TCP and MQTT reported an average of 51.38 ms and 613.51 ms, respectively.
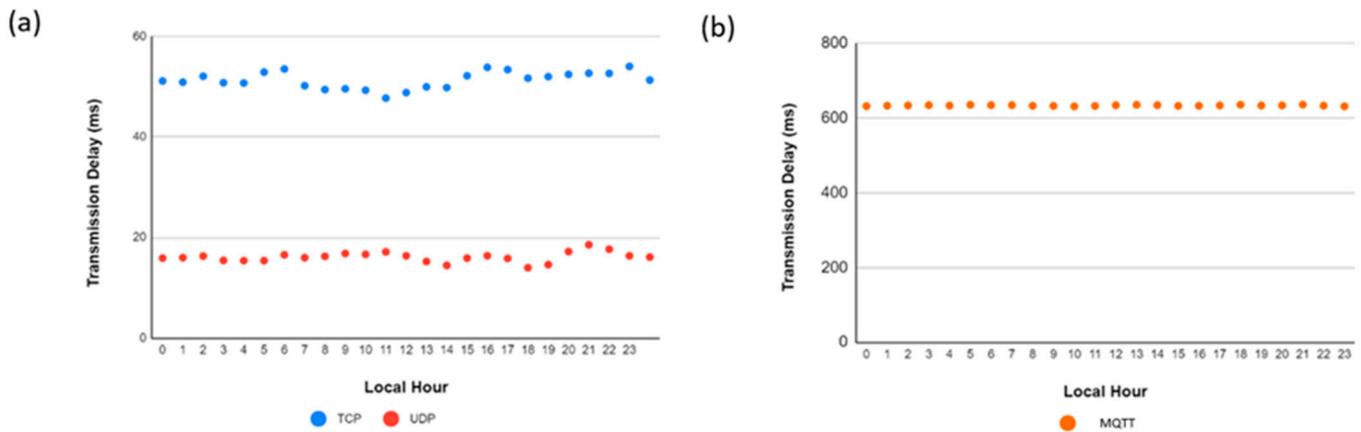
Figure 6. (**a**) Hourly distribution of the transmission delay during a 24-h period for TCP (blue dots) and UDP (red dots). (**b**) Hourly distribution of the transmission delay during a 24-h period for MQTT (orange dots).

It is evident that the $\delta t_{\text{transmission}}$ of UDP is much smaller than that of TCP and MQTT, but the reliability of the data delivery of each protocol should be considered before selecting the best communication protocol for the proposed EEW network. Therefore as the next step, the reliability of each communication protocol is analysed.

According to the communication protocol characteristics, it is evident that TCP has its own benefits, such as error checking mechanism and handshaking process between the sensors, which ensure reliable data communication compared to the UDP, which has none of the above characteristics.

When selecting the most appropriate communication protocol, it is evident that MQTT is the least appropriate due to its high $\delta t_{\text{transmission}}$, so it is not suitable for the proposed EEW architecture. As the next step, UDP and TCP were compared to identify the most appropriate. Even though UDP protocol resulted in a $\delta t_{\text{transmission}}$ which is one-third of the TCP's $\delta t_{\text{transmission}}$, UDP does not guarantee reliable data delivery, which can be considered a significant limitation for a time-critical application. It was also observed that the performance difference between the protocols is in the millisecond range, and hence can be considered as only creating a negligible impact on the system latency.

Having considered findings from the evaluation process of the protocols (MQTT, UDP and TCP), TCP was selected as the best communication protocol for the proposed EEW network to conduct the communication between the sensors.

Calculating the Detection Time (δt_{detect}) of the Network

The detection time (δt_{detect}) for the proposed EEW architecture was calculated in the RS4D sensor environment by implementing a modified PLUM-based earthquake detection algorithm. In this experimental environment, the calculated δt_{detect} varied between 0.10 and 0.19 s for the six selected hypothetical earthquake scenarios (Table 1). We selected the PLUM-based approach due to its simplicity and easy-to-implement nature [71]. Additionally, PLUM does not calculate the source parameters of the earthquake, which makes it comparatively faster than most other EQ detection approaches [11]. More importantly, it requires less processing power. Therefore, the obtained δt_{detect} values show that the RS4D type of low-cost sensor, with its Raspberry Pi 3 Model B processor, is easily able to provide the level of processing power required to run a PLUM-based detection algorithm.

Table 1. Reported average latencies for the six hypothetical earthquake scenarios with the decentralised architecture.

Hypothetical Scenarios *	Decentralised Processing Using TCP (in Seconds)			
	δt_{detect}	$\delta t_{transmission}$	$\delta t_{s\text{-}wave\_travel}$	$\delta t_{sys\_latency}$
Scenario 1	0.10	0.05	2.50	2.65
Scenario 2	0.13	0.05	2.70	2.88
Scenario 3	0.19	0.05	1.00	1.24
Scenario 4	0.17	0.05	1.20	1.42
Scenario 5	0.17	0.05	1.10	1.32
Scenario 6	0.19	0.05	1.60	1.84

* See Appendix A for illustrations of the scenarios with corresponding azimuthal directions.

Calculating the S-Wave Travel Time ($\delta t_{s\text{-}wave\_travel}$) of the Network

When it comes to S-wave travel time ($\delta t_{s\text{-}wave\_travel}$) between the first and the second observation sensors, the travel time differs according to the earthquake scenario, since it is primarily dependent on the respective sensor locations, in relation to the direction of the S-wave. Our findings from the six earthquake scenarios presented in this paper confirm the above observation as the S-wave travel time ($\delta t_{s\text{-}wave\_travel}$) between the first and second sensors varied between 1.0 and 2.7 s (Table 1). Furthermore, it is evident from the findings that there is a clear correlation between $\delta t_{s\text{-}wave\_travel}$ and the density of the sensor distribution in the S-wave travel direction. When the S-wave is directed towards the areas with lesser sensor distribution, it resulted in a higher $\delta t_{s\text{-}wave\_travel}$ value, whereas a lower $\delta t_{s\text{-}wave\_travel}$ value occurred in the direction of higher sensor distribution.

Overall, simulation experiments conducted with the node-level processing approach resulted in a system latency ($\delta t_{sys\_latency}$) between 1.24 and 2.88 s (Table 1).

Comparison of System Latencies for the Decentralised and Centralised EEW Architectures

In addition to conducting experiments at the node-level with decentralised architecture, we also implemented a centralised network architecture connecting the same set of sensors used for the node-level processing to make a comparison (Figure 7). Unlike the decentralised architecture, the centralised processing architecture consists of an additional computer that acts as a central server to perform all the detection algorithm processing. Sensors in the centralised architecture forward the ground-motion data they captured directly to this dedicated server, at which point the detection algorithm is executed. After the execution, the computer will disseminate the alert data to the appropriate sensors.

Although there is not much evidence available in the literature about the specifications used for centralised processing servers, we found that most of the implementations published with centralised processing used the Amazon Web Services (AWS) as their centralised data processor [9,41]. However, the specifications of the implemented AWS server were not described in any of the centralised processing literature. Thus, guided by the above literature, for our decentralised vs. centralised comparison, we used the Amazon Web Services (AWS) virtual machine as our centralised server to connect the RS4D sensors with the configuration of t2. micro instance, 1GB RAM and 8GB memory.

The experimental simulations with the centralised architecture were conducted using two standard communication protocols: TCP and MQTT.
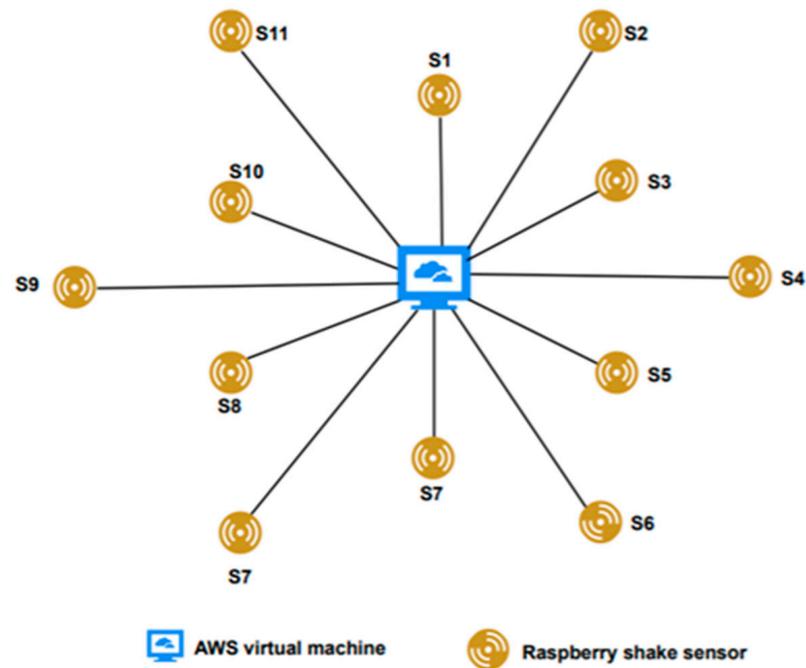
Figure 7. Centralised processing architecture using the AWS virtual machine.

Although we selected TCP as the best communication protocol to implement the decentralised EEW architecture, when calculating the latency for the centralised architecture, we compared the performance of the centralised architecture with the use of both TCP and MQTT. This is because, traditionally, most of the centralised EEW architectures primarily use MQTT as their communication protocol, but not TCP. As shown in Table 2, the results obtained from the comparison experiments clearly show that the TCP-based decentralised EEW architecture proposed in this research, implemented on a ZeroTier SD-WAN platform, outperforms the traditional centralised EEW architecture that operates with either MQTT or TCP.
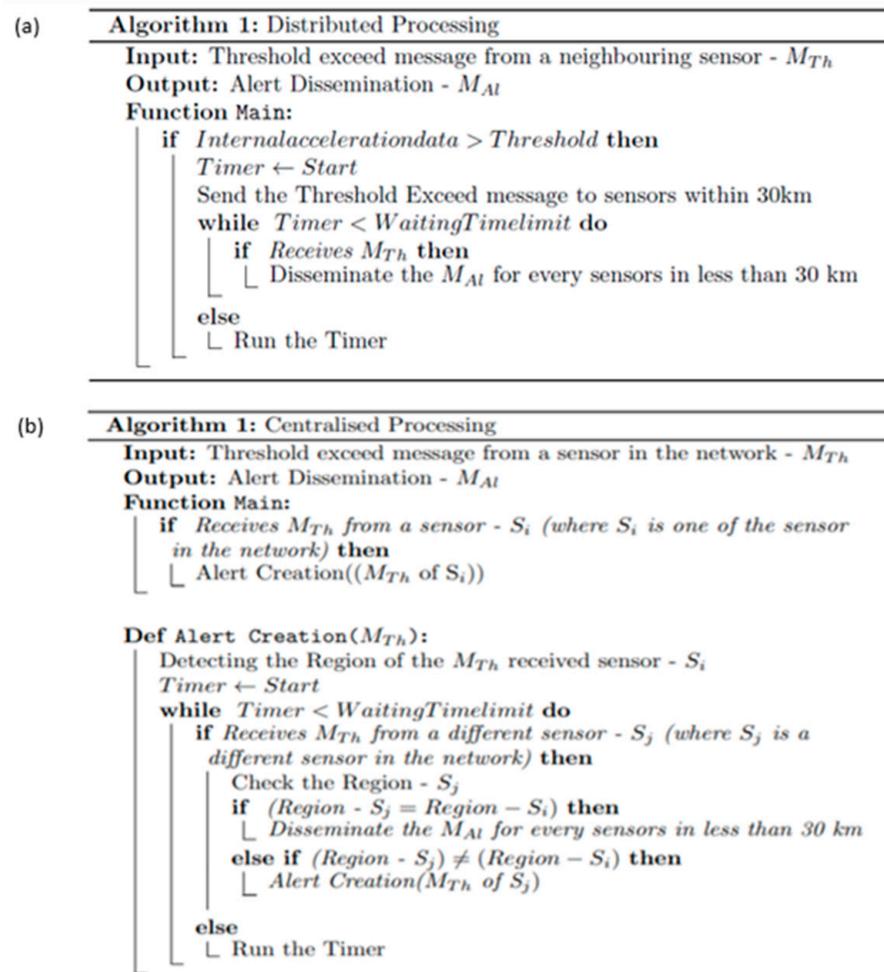
Table 2. Reported latencies for the six earthquake scenarios with three different architectures.

Hypothetical Scenarios	System Latency for Centralised Processing (in Seconds)		System Latency for Decentralised Processing Using TCP (in Seconds)
	MQTT	TCP	
Scenario 1	4.15	2.94	2.65
Scenario 2	4.57	3.13	2.88
Scenario 3	3.24	1.48	1.24
Scenario 4	3.51	1.69	1.42
Scenario 5	3.32	1.64	1.32
Scenario 6	3.84	2.08	1.84

As shown in Table 2, the results also show that, when running the simulations for the six earthquake scenarios, the centralised processing model with TCP as the communication protocol performed better compared to MQTT. This is evident for the centralised architecture with a TCP reported system latency ($\delta t_{sys\_latency}$) in the range of 1.48–3.13 s, whereas, when using the MQTT, the system latency ($\delta t_{sys\_latency}$) varied between 3.24 and 4.57 s.

We also observed that, when using the centralised processing approach, the system latency increased with the number of sensors in the network. In a real-time scenario, a typical nationwide EEW network may consist of hundreds of sensors and, during an actual

earthquake, with the use of the PLUM approach, the system has to deal with a significant number of circular regions with a 30 km radius. In such a situation, the system has to identify the region of the detection sensor accurately, followed by its neighbouring sensors. In a centralised arrangement, this processing needs to be carried out in a single dedicated server, and a high number of sensors attached to a number of circular regions will lead to an increase in the complexity of the processing software. The increased complexity of the software is evident when comparing the pseudocodes of the decentralised architecture (Figure 8a) and centralised architecture (Figure 8b). It can be clearly seen that, as shown in Figure 8b, for the centralised processing architecture, it is essential to include two additional **if** conditional blocks to identify the region and the neighbouring sensors of a particular detection sensor.

(a)

Algorithm 1: Distributed Processing

Input: Threshold exceed message from a neighbouring sensor - M_{Th}
Output: Alert Dissemination - M_{Al}
Function Main:
  if $Internal\,acceleration\,data > Threshold$ **then**
    $Timer \leftarrow Start$
    Send the Threshold Exceed message to sensors within 30km
    while $Timer < WaitingTimelimit$ **do**
      if $Receives\ M_{Th}$ **then**
        Disseminate the M_{Al} for every sensors in less than 30 km
  else
    Run the Timer

(b)

Algorithm 1: Centralised Processing

Input: Threshold exceed message from a sensor in the network - M_{Th}
Output: Alert Dissemination - M_{Al}
Function Main:
  if $Receives\ M_{Th}$ from a sensor - S_i (where S_i is one of the sensor in the network) **then**
    Alert Creation((M_{Th} of S_i))

Def Alert Creation(M_{Th}):
  Detecting the Region of the M_{Th} received sensor - S_i
  $Timer \leftarrow Start$
  while $Timer < WaitingTimelimit$ **do**
    if $Receives\ M_{Th}$ from a different sensor - S_j (where S_j is a different sensor in the network) **then**
      Check the Region - S_j
      if $(Region - S_j = Region - S_i)$ **then**
        Disseminate the M_{Al} for every sensors in less than 30 km
      else if $(Region - S_j) \neq (Region - S_i)$ **then**
        Alert Creation(M_{Th} of S_j)
    else
      Run the Timer

Figure 8. (**a**) Pseudocode for the decentralised processing algorithm. (**b**) Pseudocode for the centralised processing algorithm.

With this added complexity, the respective pseudocodes show that the processing time rises along with the number of sensors in the EEW network. Therefore, in a real-life scenario, when an EEW system carries a high number of sensors located across a vast geographic region, a system that operates with centralised processing, identifying the region of a sensor and its neighbouring sensors will become significantly time-consuming. In contrast, when using the decentralised approach, the system latency does not depend on the number of sensors outside the radius of its 30 km region; each sensor will only communicate with sensors within the 30 km radius.

To check the validity of the above observation, in addition to the five sensors installed in the Wellington region, we introduced several additional sensors installed in different regions outside Wellington. We recalculated the system latency with the centralised processing network approach (Figure 9). As shown in Figure 9, these additional sensors were installed in four geographically dispersed regions on the North Island of NZ.
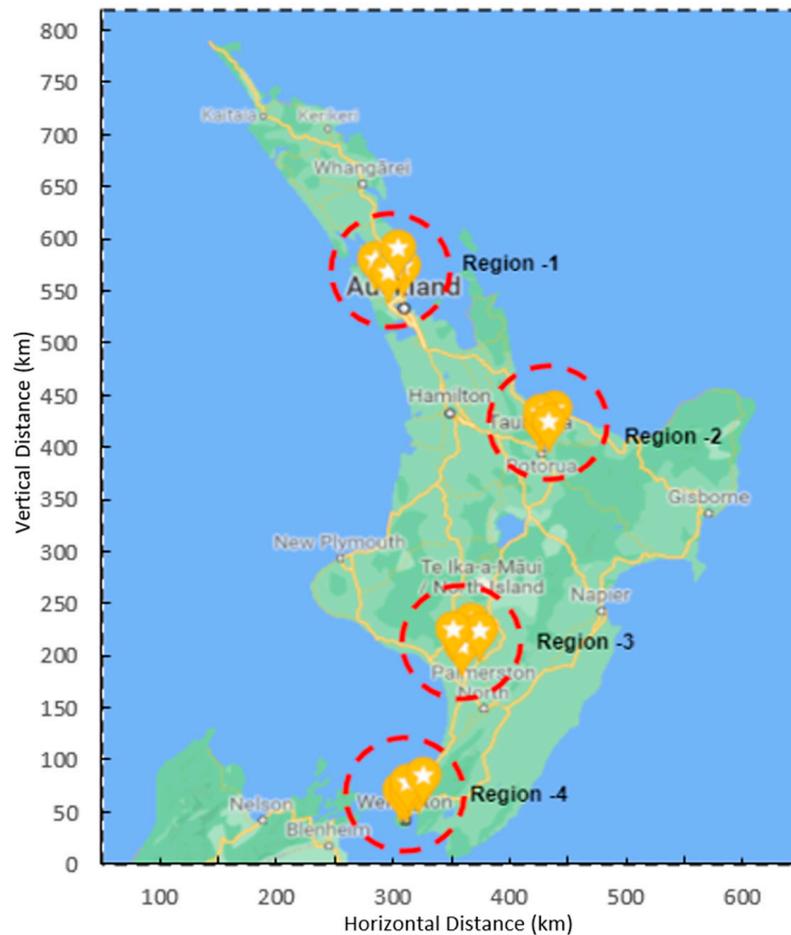

Figure 9. Sensors installed in different regions of New Zealand.

As discussed above, our proposed approach is mainly based on disseminating the alert in a 30 km radius. The introduction of more sensors in a large area, as shown in Figure 9, increased the processing time at the central server. Introducing additional sensors creates an essential need to identify the sensor's region and its neighbouring sensors during a particular earthquake event.

The results obtained from this multiple region scenario show that the system latency varied from 1.52 to 3.2 s for TCP. This was earlier reported as 1.48 to 3.13 s when operating with only a single, 30-km-radius circular region (Table 2). Despite the above findings having reported a marginal increase in latency, the above experimental results provide evidence to show that the system latency of the centralised architecture continues to increase with each additional sensor that is added into the system. Therefore, we can argue that the system latency of an EEW system with centralised architecture can increase in a situation where the network consists of a significantly high number of sensors dispersed across multiple 30-km regions.

Calculation of Packet Loss

In addition to calculating the system latency, we further investigated the packet loss for the three-sensor network approaches (decentralised processing with TCP, centralised processing with TCP and centralised processing with MQTT) to compare the system latency. These investigations reported significant packet loss when we ran the experiments in a centralised architecture environment where the MQTT was used as the primary communication protocol (Figure 10). In contrast, the reported packet loss was zero for both the decentralised and centralised architecture when we used the TCP as the communication protocol. Although the MQTT is built on top of TCP, the main reason for the packet loss is the packet drop in the centralised broker. We used the free version of the PAHO MQTT centralised broker, where a considerable amount of data packets was dropped due to the heavy traffic in the server [86].
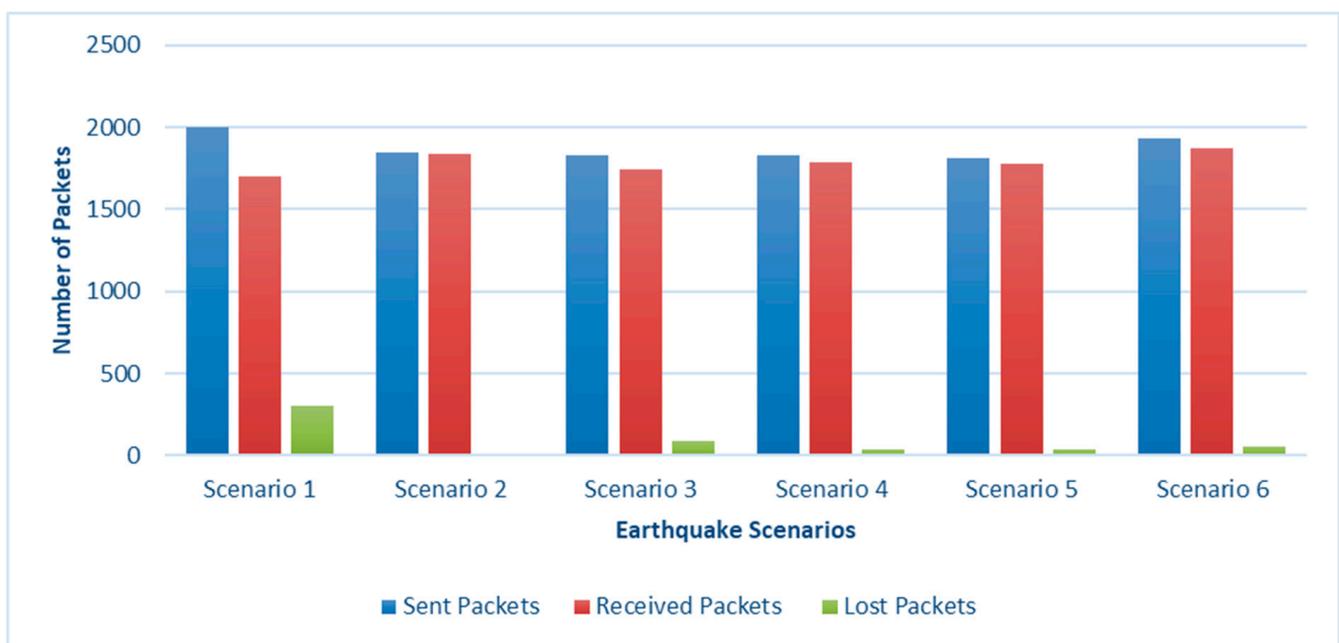
Figure 10. Lost data packets in centralised processing with MQTT as the communication protocol.

5. Discussion

Unique to any of the previously conducted EEW research across the globe, the research presented in this paper introduces a comprehensive sensor network architecture from scratch, with the specifications of the essential components needed to construct a low-cost, MEMS-based EEW system. Mostly, the previously published literature on EEW systems primarily focused only on discussions of system latency and the accuracy of the network architecture [8,11,41]. In contrast, this paper has investigated all the components and steps required to implement an EEW system and compared them with the existing approaches. In addition, the decentralised EEW sensor network architecture proposed in this paper shows that the detection of earthquakes and processing of ground-motion data can successfully be implemented at the sensor node. This research explored and implemented an experimental EEW system by employing 100% decentralised processing compared to the currently available EEW approaches based on centralised processing [8,11,41,87]. Even though Fischer and colleagues proposed a decentralised EEW approach, there are no clear findings of the robustness of their system [15]. Further, this paper also demonstrates that using a lightweight and easy-to-implement algorithm such as PLUM can be considered an ideal EEW approach, that is suitable for implementation in resource-constrained environments such as low-cost, RS MEMS-based sensors.

Our work demonstrates that a low-cost, MEMS-based sensor network with decentralised processing can be used to produce EEW alerts to the public with a minimal cost compared to both high-end EEW systems such as California's ShakeAlert and low-cost systems such as Costa-Rica's ASTUTI [41]. Further, it should be noted that, with the decentralised processing, the proposed EEW architecture outperforms a system with centralised processing; therefore, there will be no additional costs in implementing a centralised middleware server. The major proportion of the cost of our proposed EEW solution is allocated for purchasing the MEMS-based RS sensors. Furthermore, the annual running cost of the proposed network primarily consists of the internet usage of the sensors. Implemented as a community-engaged EEW solution, the public usually absorbs the internet charges.

Most of the network architectures constructed for EEW systems in the past were mainly focused on centralised processing rather than node-level processing. From the results of our proposed decentralised processing approach, it is clear that it outperforms the other proposed approaches worldwide [11]. The data transmission delay between the sensors plays a significant role in the system latency of the EEW system. Our results show that the standard communication protocols UDP and TCP outperform the commonly used communication protocol MQTT. Even though the UDP outperforms TCP by a minimum value of approximately 35 ms, it is always advisable to use TCP as the communication protocol for time-critical applications due to its higher reliability compared to UDP. To compare and evaluate the performance of the communication protocols, along with the proposed SD-WAN architecture, we implemented a centralised processing architecture by adding an AWS virtual machine to the network. From the results obtained from the latency calculations, we can see that the system latency of our proposed decentralised processing outperforms the MQTT-based centralised processing approach by a considerable value of approximately 2 s. It is also evident that the packet loss when using MQTT is a significant drawback for a time-critical application, where packet losses cannot be accepted.

The results show that the system latency of the centralised processing increases with the number of sensors in the network since there is only one processing unit for the complete network, compared to decentralised processing, where each sensor processes the algorithm for the sensors within the 30 km radius. Furthermore, the additional processing blocks for the centralised server, such as identifying the area of the particular sensor and the neighbouring sensors in an earthquake event, will also add to the delay. In comparison, our approach will not have such a delay as the sensors only directly communicate with other sensors in the 30-km-radius area. Our results also show that the decentralised processing approach significantly reduces latency by deploying more sensors in the network, which will shrink the travel time of the S-wave between the two neighbouring sensors [33]. Furthermore, for a larger network with a considerable number of sensors, it should be noted that the processing time of the algorithm with the centralised processing approach can only be reduced by improving the processing power of the centralised server, which will eventually raise the cost of the centralised architecture. On the other hand, the cost of the decentralised EEW architectures is becoming more affordable because MEMS-based sensors are getting cheaper, while their processing power is increasing rapidly.

Regarding system latencies, the above discussion clearly shows that the implementation of the TCP-based decentralised processing architecture outperforms other centralised processing architectures implemented around the world.

In addition, redundancy should be considered in case of a failure in the EEW network architecture. In our approach, the redundancy is mainly dependent on the density of the sensors in a 30 km radius area, since we are processing the algorithm at the sensor node. Therefore, the failure of a single or multiple sensors may not cause any major network failure since the remaining sensors in a particular area will continue to process the data and detect earthquakes. On the contrary, in the centralised processing approach, failure to connect to the centralised server or failure of the centralised server itself will collapse the functionality of the entire network, and lead to failure to detect an earthquake. The EEW

sensor network architecture proposed in this paper is less prone to failures compared to an architecture driven by centralised processing.

In addition to that, most of the EEW approaches found in the previous literature were implemented using the MQTT-based centralised processing. It is proven that the TCP-based centralised processing outperforms the MQTT-based centralised processing approach. Even though implementing a TCP-based centralised processing architecture requires the inclusion of a software-defined network at the node-level to identify the sensors uniquely, we can propose TCP as a better choice when reducing the system latency of the network, compared to MQTT, for a centralised EEW network.

Furthermore, in this research, we investigated the potential security-related risks and identified the potential security breaches that can be anticipated in the proposed type of community-engaged EEW sensor network environment. To mitigate the identified vulnerabilities, we implemented appropriate measures to secure the proposed EEW sensor network, which runs on the ZeroTier SD-WAN platform. At present, there is hardly any EEW literature that addresses potential security risks or provides solutions to mitigate such risks.

In addition to the above factors, the proposed architecture can be easily scaled, implemented, and exported to develop a sensor network by simply provisioning low-cost sensors, installing them in people's homes, and implementing decentralised processing at the node-level. Six of the selected hypothetical earthquake scenarios were triggered by S-waves. The scenarios suggest that a low-cost, MEMS-based, decentralised processing network could achieve the fastest theoretical EEW performance compared with the other centralised EEW networks, especially with the anticipated futuristic improvements in low-cost sensors and processing algorithms. These types of low-cost sensors could also support the implementation of hybrid networks with the aim of enhancing and complementing existing EEW networks consisting of expensive Class A type seismographs. Our approach to ground-motion detection with an appropriate alerting mechanism has demonstrated an effective EEW solution, where alerts may arrive early, allowing the system end-users to carry out simple protective actions, such as drop-cover and hold.

Limitations and Future Work

While this paper provided evidence that an EEW system can be implemented using MEMS-based, low-cost sensors without any centralised processing, we identified several limitations that need further investigation. The inherent limitations of the PLUM approach constrain our proposed EEW system. While the PLUM algorithm is considered a more robust approach to detect seismic intensity, it limits the warning time to a maximum of ~10 s. Regarding the further use of S-Wave to detect the intensity of shaking, PLUM approach makes it unsuitable for providing a meaningful warning to the areas near the epicentre. To minimise the inherent limitations of the PLUM algorithm in the next phase of our research, we intend to investigate the feasibility of predicting the S-wave shaking intensities using the P-waves, which can eventually considerably increase the warning time. Additionally, we are looking into different algorithms, which could predict the shaking intensity beyond the 30-km radius defined by the PLUM algorithm. Further, our team is exploring the opportunities to implement a community-engaged sensor network comprising different types of low-cost sensors, rather than using a single type of sensor. The successful implementation of an EEW network with multiple sensor types will result in an eco-system of community-engaged, MEMS-based, EEW sensor networks.

As described in Section 4.1.1, even though we have conducted a preliminary testing of the performance of the embedded additional measures to enhance the security of the proposed MEMS-based, decentralised sensor network, to provide a more accurate judgment, security enhancements must be tested when exposed to real-world threats when operating under real-world scenarios. We intend to carry out such an in-depth testing of the implemented security enhancements as one of the future activities in the ongoing research. Furthermore, we intend to develop a centralised alert service that could detect and notify

users of security breaches in the sensor network and is capable of automating actions upon the suspicious activity (e.g., automatically disconnecting the breached and vulnerable sensors from the network). This feature will automatically remove the vulnerable sensor nodes from the network, which will eventually increase the security of the network.

6. Conclusions

We have demonstrated that the PLUM-based EEW algorithm can be implemented using a network of low-cost, MEMS-based sensors, providing accurate operational EEW at a lower cost compared with scientific-grade seismographs. Furthermore, we investigated the overall system latency of the proposed EEW system and its components in the proposed network. From the outcomes of the transmission delay, along with different standard communication protocols, we can confirm that the use of TCP as a communication protocol running on an appropriate SD-WAN solution can reduce the transmission delay, regardless of the type of processing architecture (centralised or decentralised). In terms of the detection time, by introducing the decentralised processing architecture, we showed that our results outperform the commonly implemented centralised processing architectures. It should be noted that the detection time of our proposed decentralised processing network does not vary with the number of sensors in the network. Thus, it should show approximately the same results as the nationwide EEW system; however, when it comes to the centralised processing, the algorithm's detection time tends to increase with the number of sensors. Furthermore, the packet loss in the decentralised processing architecture is negligible compared to the centralised architecture.

In this paper, we investigated the feasibility of implementing an EEW network that processes the detection algorithm at node-level rather than in a centralised processor. We also presented a step-by-step guide to building an EEW network with low-cost, MEMS-based sensors. This research provides clear evidence to confirm that the proposed type of EEW system can successfully be implemented by choosing the appropriate detection sensors and algorithms. Therefore, this paper can be considered as providing a comprehensive guide to construct an EEW sensor network with decentralised processing and can be used as a benchmark, which is beneficial in building similar networks in the future. Furthermore, the proposed concept of a decentralised, low-cost sensor network architecture can be used to implement community-engaged warning applications in the other disaster domains, such as developing low-cost warnings for bushfires.

Author Contributions: Conceptualization, R.P., C.C., R.N., C.H.,A.P., J.S.B., S.J., D.R. and R.S.; Methodology, R.P., C.C., R.N., C.H., A.P., N.L. and D.R.; software, C.C., R.N., N.L., D.R. and R.S.; formal analysis, R.P. and C.C.; investigation, R.P., C.H., A.P., J.S.B., N.L. and R.S.; data curation, R.N. and N.L.; writing—original draft preparation, R.P and C.C.; writing—review and editing, R.P., C.C., R.N., C.H., A.P., J.S.B., S.J., N.L., D.R. and M.L.T.; visualization, C.C. and R.N.; project administration, R.P. and M.L.T.; funding acquisition, R.P., C.C., C.H., A.P., J.S.B., S.J., N.L. and M.L.T. All authors have read and agreed to the published version of the manuscript.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

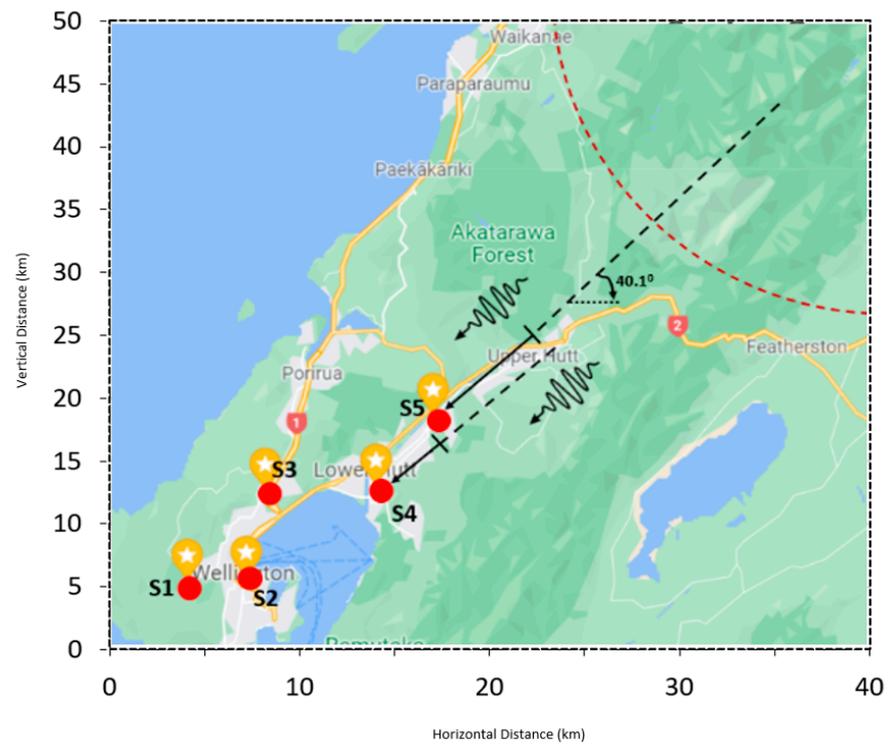## Appendix A. Map Illustration of the Latency Test Scenarios
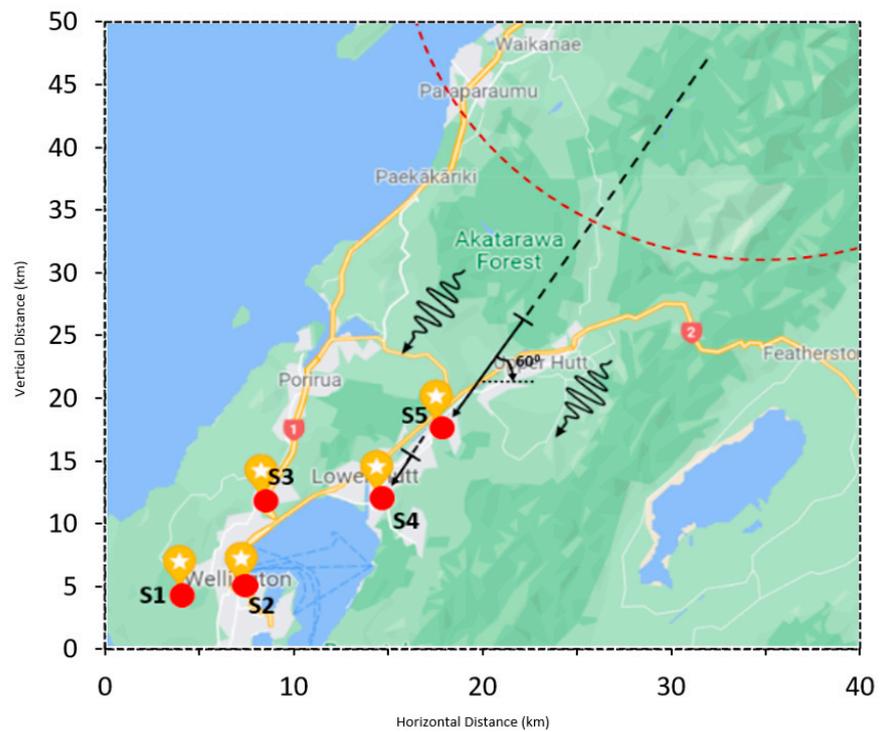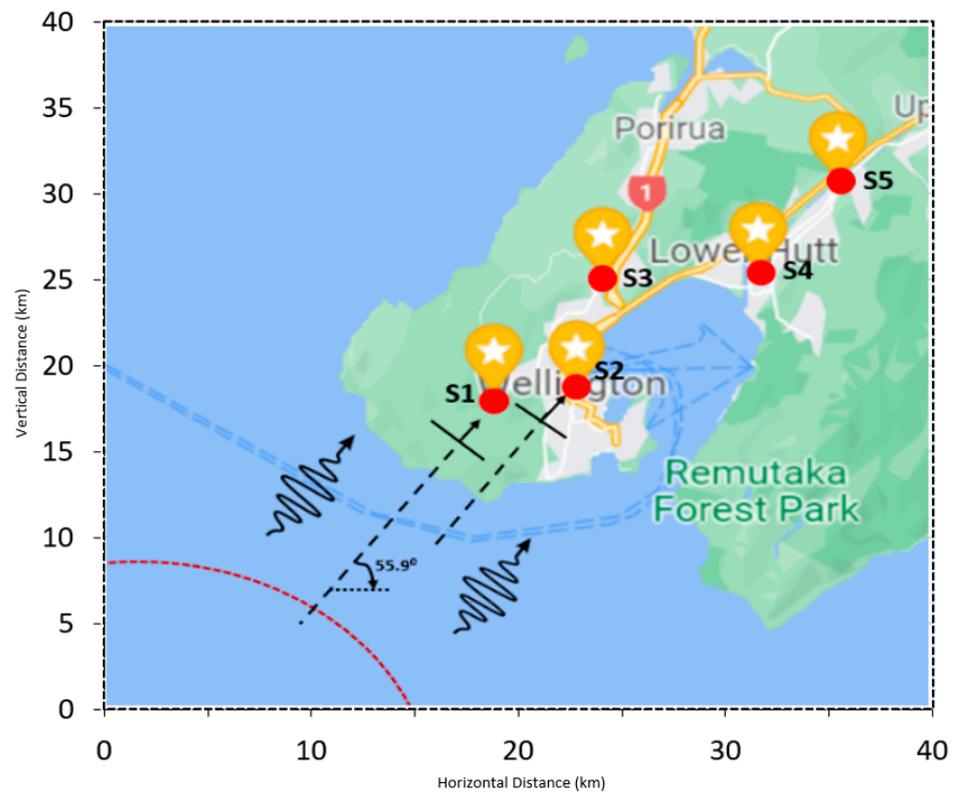

Figure A1. Scenario 1.

Figure A2. Scenario 2.
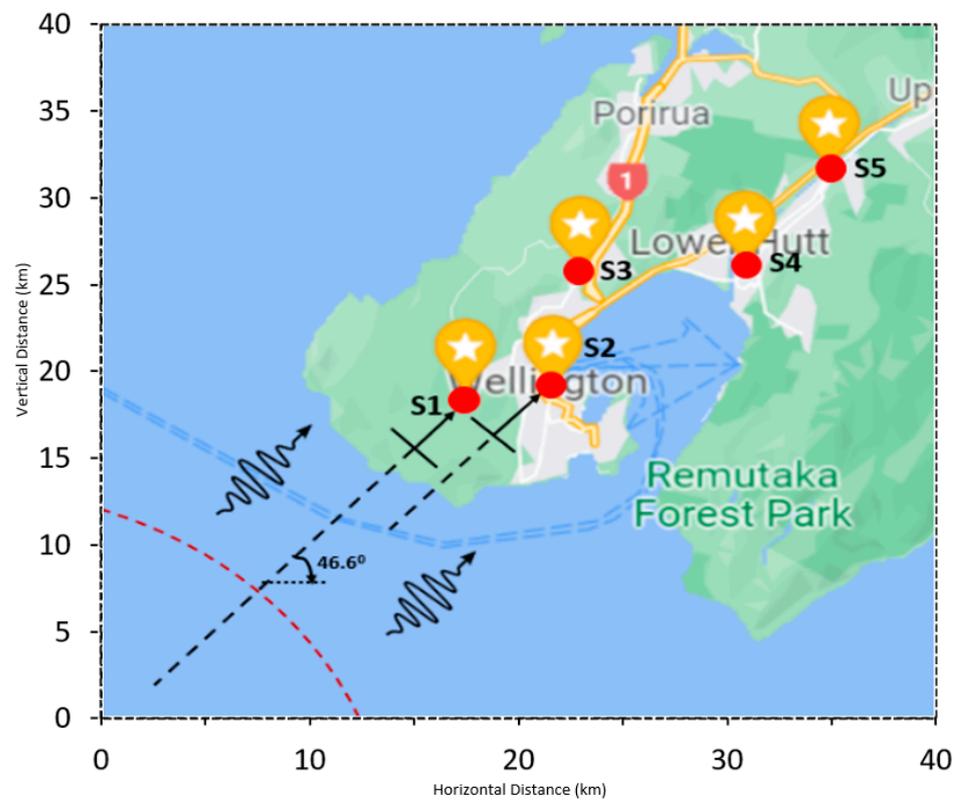
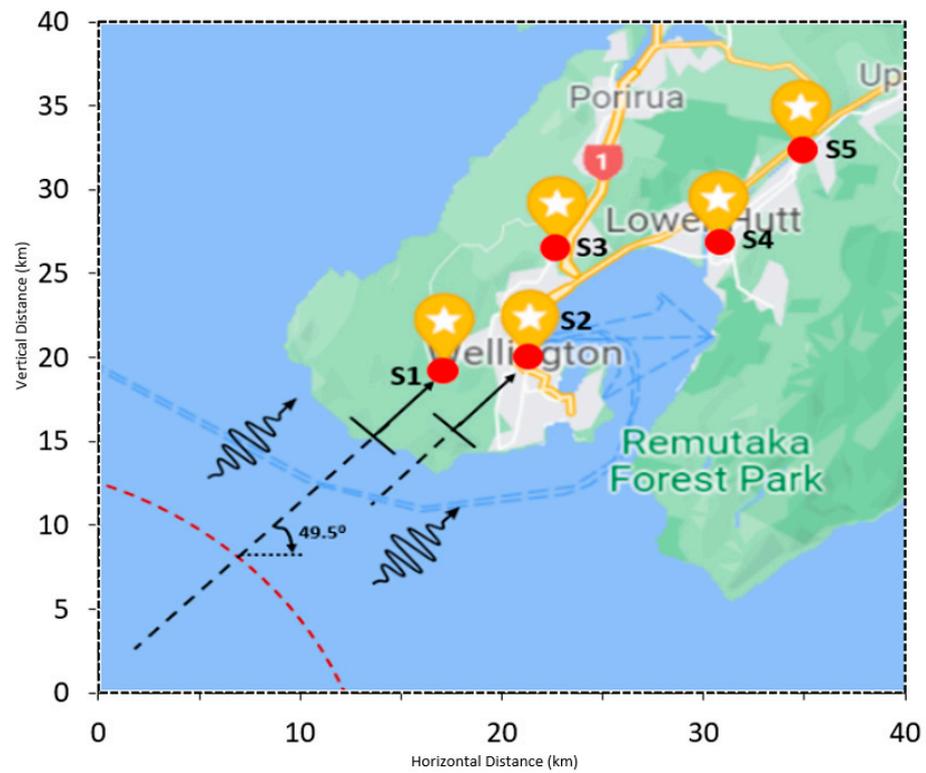**Figure A3.** Scenario 3.

Figure A4. Scenario 4.
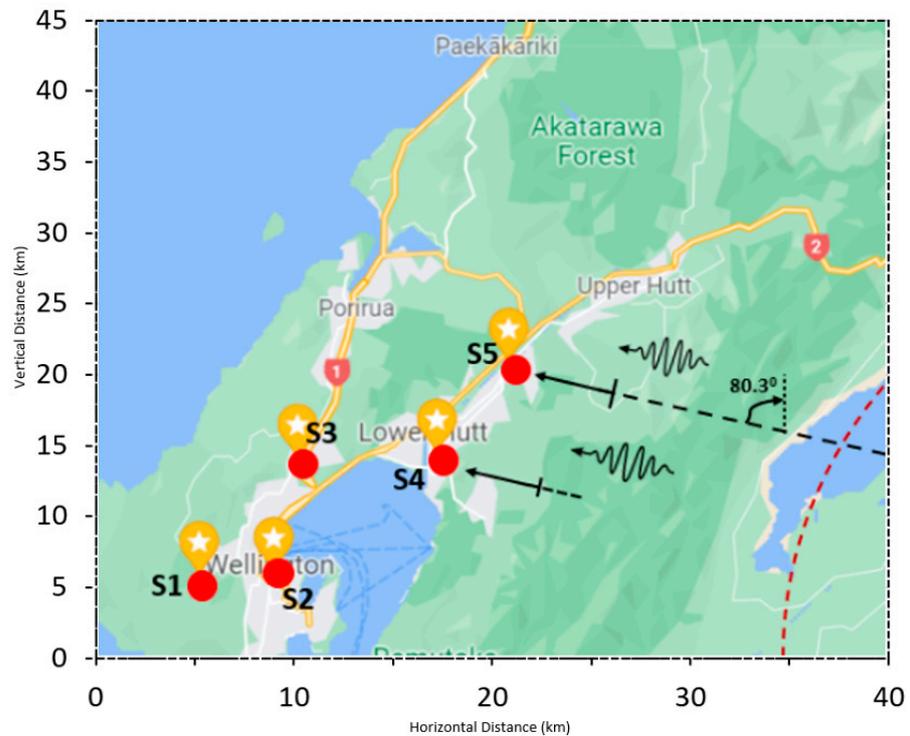
Figure A5. Scenario 5.

Figure A6. Scenario 6.

The following link provides access to the GitHub repository containing source code that can be used to implement the EEW sensor network architecture proposed in this paper: https://github.com/rs-networking/Decentralised_Processing_PLUM_V1.0.git (accessed on 20 August 2021)

References

1. Strauss, J.A.; Allen, R. Benefits and Costs of Earthquake Early Warning. *Seism. Res. Lett.* **2016**, *87*, 765–772. [CrossRef]
2. Adhikari, M.; Paton, D.; Johnston, D.; Prasanna, R.; McColl, S.T. Modelling predictors of earthquake hazard preparedness in Nepal. *Procedia Eng.* **2018**, *212*, 910–917. [CrossRef]
3. Wu, Y.-M.; Zhao, L. Magnitude estimation using the first three seconds P-wave amplitude in earthquake early warning. *Geophys. Res. Lett.* **2006**, *33*, 4–7. [CrossRef]
4. Nakayachi, K.; Becker, J.S.; Potter, S.H.; Dixon, M. Residents' Reactions to Earthquake Early Warnings in Japan. *Risk Anal.* **2019**, *39*, 1723–1740. [CrossRef]
5. Becker, J.S.; Potter, S.H.; Vinnell, L.J.; Nakayachi, K.; McBride, S.K.; Johnston, D.M. Earthquake early warning in Aotearoa New Zealand: A survey of public perspectives to guide warning system development. *Humanit. Soc. Sci. Commun.* **2020**, *7*, 1–12. [CrossRef]
6. Becker, J.S.; Potter, S.H.; Prasanna, R.; Tan, M.L.; Payne, B.A.; Holden, C.; Horspool, N.; Smith, R.; Johnston, D.M. Scoping the potential for earthquake early warning in Aotearoa New Zealand: A sectoral analysis of perceived benefits and challenges. *Int. J. Disaster Risk Reduct.* **2020**, *51*, 101765. [CrossRef]
7. Kanamori, H.; Hauksson, E.; Heaton, T. Real-time seismology and earthquake hazard mitigation. *Nature* **1997**, *390*, 461–464. [CrossRef]
8. Wang, Y.; Li, S.; Song, J. Threshold-based evolutionary magnitude estimation for an earthquake early warning system in the Sichuan–Yunnan region, China. *Sci. Rep.* **2020**, *10*, 201055. [CrossRef]
9. Given, D.D.; Cochran, E.S.; Heaton, T.H.; Hauksson, E.; Allen, R.M.; Hellweg, P.; Vidale, J.; Bodin, P. Technical Implementation Plan for the ShakeAlert Production System-An Earthquake Early Warning System for the West Coast of the United States. 2014. Available online: www.scec.org/terashake (accessed on 20 August 2021).
10. Teymourian, J.; Moinfar, A.A.; Naderzadeh, A. Strengthening of existing buildings against earthquake with consideration of economic constraints in developing countries. In Proceedings of the 13th World Conference on Earthquake Engineering, Vancouver, BC, Canada, 1–6 August 2004; pp. 1–12.
11. Allen, R.M.; Melgar, D. Earthquake Early Warning: Advances, Scientific Challenges, and Societal Needs. *Annu. Rev. Earth Planet. Sci.* **2019**, *47*, 361–388. [CrossRef]
12. Peng, C.; Jiang, P.; Chen, Q.; Ma, Q.; Yang, J. Performance Evaluation of a Dense MEMS-Based Seismic Sensor Array Deployed in the Sichuan-Yunnan Border Region for Earthquake Early Warning. *Micromachines* **2019**, *10*, 735. [CrossRef]
13. Nof, R.N.; Chung, A.I.; Rademacher, H.; Dengler, L.; Allen, R.M. MEMS Accelerometer Mini-Array (MAMA): A Low-Cost Implementation for Earthquake Early Warning Enhancement. *Earthq. Spectra* **2019**, *35*, 21–38. [CrossRef]
14. Wu, Y.; Mittal, H.; Huang, T.; Yang, B.M.; Jan, J.; Chen, S.K. Performance of a Low-Cost Earthquake Early Warning System (P-Alert) and Shake Map Production during the 2018 Mw 6.4 Hualien, Taiwan, Earthquake. *Seism. Res. Lett.* **2019**, *90*, 19–29. [CrossRef]
15. Fischer, J.; Redlich, J.-P.; Zschau, J.; Milkereit, C.; Picozzi, M.; Fleming, K.; Brumbulli, M.; Lichtblau, B.; Eveslage, I. A wireless mesh sensing network for early warning. *J. Netw. Comput. Appl.* **2012**, *35*, 538–547. [CrossRef]
16. Aizu, I. The role of ICT during the disaster—A story of how Internet and other information and communication services could or could not help relief operations at the Great East Japan Earthquake. *Glob. Inf. Soc. Watch Rep.* **2011**, *2011*, 1–7.
17. GeoNet. GeoNet Home. Available online: https://www.geonet.org.nz/ (accessed on 20 August 2021).
18. Voosen, P. New Google effort uses cellphones to detect earthquakes. *Science* **2021**, *48*, 101721. [CrossRef]
19. McDonald, L. New Google Trial Will Send Earthquake Alerts to Android Phones in New Zealand. Available online: https://www.stuff.co.nz/technology/apps/124969803/new-google-trial-will-send-earthquake-alerts-to-android-phones-in-new-zealand (accessed on 12 November 2021).
20. Hoshiba, M.; Iwakiri, K. Initial 30 seconds of the 2011 off the Pacific coast of Tohoku Earthquake (Mw 9.0)—amplitude and τ c for magnitude estimation for Earthquake Early Warning? *Earth, Planets Space* **2011**, *63*, 553–557. [CrossRef]
21. Field, E.H.; Hough, S.E. The Variability of PSV Response Spectra across a Dense Array Deployed during the Northridge Aftershock Sequence. *Earthq. Spectra* **1997**, *13*, 243–257. [CrossRef]
22. Tan, M.L.; Prasanna, R.; Becker, J.; Brown, A.; Stock, K.; Kenney, C.; Lambie, E.; Johnston, D.; De Alwis, D. Outlook for earthquake early warning for Aotearoa New Zealand: Insights from initiating a community-of-practice. In Proceedings of the 2021 Technical Conference for the New Zealand Society for Earthquake Engineering, Christchurch, New Zealand, 14–16 April 2021; pp. 1–8.
23. Santos, J.; Catapang, A.N.; Reyta, E.D. Understanding the Fundamentals of Earthquake Signal Sensing Networks. *AnalogDialogue* **2019**, *53*, 3.
24. Raspberry Shake. Specifications for: Raspberry Shake RS4D. 2021. Available online: https://manual.raspberryshake.org/_downloads/SpecificationsforRaspberryShake4DMEMSV4.pdf (accessed on 12 November 2021).
25. Wu, Y.; Chen, D.-Y.; Lin, T.; Hsieh, C.; Chin, T.; Chang, W.; Li, W.; Ker, S. A High-Density Seismic Network for Earthquake Early Warning in Taiwan Based on Low Cost Sensors. *Seism. Res. Lett.* **2013**, *84*, 1048–1054. [CrossRef]
26. Babatain, W.; Bhattacharjee, S.; Hussain, A.M.; Hussain, M.M. Acceleration Sensors: Sensing Mechanisms, Emerging Fabrication Strategies, Materials, and Applications. *ACS Appl. Electron. Mater.* **2021**, *3*, 504–531. [CrossRef]
27. Cochran, E.S. To catch a quake. *Nat. Commun.* **2018**, *9*, 2508. [CrossRef] [PubMed]

28. Ben, R.A.; Mellouli, S. Smart Cities in the Era of Artificial Intelligence and Internet of Things: Promises and Challenges. *Public Adm. Inf. Technol.* **2021**, *37*, 259–288.

29. Jayawardene, V.; Huggins, T.J.; Prasanna, R.; Fakhruddin, B. The role of data and information quality during disaster response decision-making. *Prog. Disaster Sci.* **2021**, *12*, 100202. [CrossRef]

30. Anthony, R.E.; Ringler, A.T.; Wilson, D.C.; Wolin, E. Do Low-Cost Seismographs Perform Well Enough for Your Network? An Overview of Laboratory Tests and Field Observations of the OSOP Raspberry Shake 4D. *Seism. Res. Lett.* **2018**, *90*, 219–228. [CrossRef]

31. Melgar, D.; Ruiz-Angulo, A.; Pérez-Campos, X.; Crowell, B.W.; Xu, X.; Cabral-Cano, E.; Brudzinski, M.; Rodriguez-Abreu, L. Energetic Rupture and Tsunamigenesis during the 2020 Mw 7.4 La Crucecita, Mexico Earthquake. *Seism. Res. Lett.* **2021**, *92*, 140–150. [CrossRef]

32. Franchi, F.; Graziosi, F.; Marotta, A.; Rinaldi, C. Structural Health Monitoring over 5G uRLLC Network. *Lecture Notes in Civil Engineering* **2021**, *127*, 60–68.

33. Mittal, H.; Wu, Y.-M.; Sharma, M.L.; Yang, B.M.; Gupta, S. Testing the performance of earthquake early warning system in northern India. *Acta Geophys.* **2018**, *67*, 59–75. [CrossRef]

34. Zollo, A.; Colombelli, S.; Elia, L.; Emolo, A.; Festa, G.; Iannaccone, G.; Martino, C.; Gasparini, P. An Integrated Regional and On-Site Earthquake Early Warning System for Southern Italy: Concepts, Methodologies and Performances. In *Early Warning for Geological Disasters*; Wenzel, F., Zschau, J., Eds.; Springer: Berlin/Heidelberg, Germany, 2014; pp. 117–137.

35. TurnKey Earthquake Early Warning. 2020. Available online: https://earthquake-turnkey.eu/the-project/ (accessed on 21 November 2021).

36. Clayton, R.W.; Heaton, T.; Kohler, M.; Chandy, M.; Guy, R.; Bunn, J. Community Seismic Network: A Dense Array to Sense Earthquake Strong Motion. *Seism. Res. Lett.* **2015**, *86*, 1354–1363. [CrossRef]

37. Wu, Y.-M.; Mittal, H. A Review on the Development of Earthquake Warning System Using Low-Cost Sensors in Taiwan. *Sensors* **2021**, *21*, 7649. [CrossRef]

38. Chamoli, B.P.; Kumar, A.; Chen, D.-Y.; Gairola, A.; Jakka, R.S.; Pandey, B.; Kumar, P.; Rathore, G. A Prototype Earthquake Early Warning System for Northern India. *J. Earthq. Eng.* **2021**, *25*, 2455–2473. [CrossRef]

39. Pierleoni, P.; Marzorati, S.; Ladina, C.; Raggiunto, S.; Belli, A.; Palma, L.; Cattaneo, M.; Valenti, S. Performance Evaluation of a Low-Cost Sensing Unit for Seismic Applications: Field Testing During Seismic Events of 2016-2017 in Central Italy. *IEEE Sens. J.* **2018**, *18*, 6644–6659. [CrossRef]

40. Chen, D.-Y.; Hsiao, N.-C.; Wu, Y.-M. The Earthworm Based Earthquake Alarm Reporting System in Taiwan. *Bull. Seism. Soc. Am.* **2015**, *105*, 568–579. [CrossRef]

41. Brooks, B.A.; Protti, M.; Ericksen, T.; Bunn, J.; Vega, F.; Cochran, E.S.; Duncan, C.; Avery, J.; Minson, S.E.; Chaves, E.; et al. Robust Earthquake Early Warning at a Fraction of the Cost: ASTUTI Costa Rica. *AGU Adv.* **2021**, *2*, e2021AV000407. [CrossRef]

42. Zambrano, A.; Perez, I.; Palau, C.; Esteve, M. Quake detection system using smartphone-based wireless sensor network for early warning. In Proceedings of the 2014 IEEE International Conference on Pervasive Computing and Communication Workshops (PERCOM WORKSHOPS), Budapest, Hungary, 24–28 March 2014; pp. 297–302. [CrossRef]

43. Lee, J.; Kim, J.-S.; Choi, S.; Kwon, Y.-W. A Smart Device Using Low-Cost Sensors to Detect Earthquakes. In Proceedings of the 2019 IEEE International Conference on Big Data and Smart Computing (BigComp), Kyoto, Japan, 27 February–2 March 2019; pp. 1–4. [CrossRef]

44. Boxberger, T.; Fleming, K.; Pittore, M.; Parolai, S.; Pilz, M.; Mikulla, S. The Multi-Parameter Wireless Sensing System (MPwise): Its Description and Application to Earthquake Risk Mitigation. *Sensors* **2017**, *17*, 2400. [CrossRef] [PubMed]

45. Fleming, K.; Picozzi, M.; Milkereit, C.; Kuhnlenz, F.; Lichtblau, B.; Fischer, J.; Zulfikar, C.; Ozel, O.; Groups, T.S.A.E.W. The Self-organizing Seismic Early Warning Information Network (SOSEWIN). *Seism. Res. Lett.* **2009**, *80*, 755–771. [CrossRef]

46. Ray, P.P.; Mukherjee, M.; Shu, L. Internet of Things for Disaster Management: State-of-the-Art and Prospects. *IEEE Access* **2017**, *5*, 18818–18835. [CrossRef]

47. Faulkner, M.; Olson, M.; Chandy, R.; Krause, J.; Chandy, K.M.; Krause, A. Demo abstract, the next big one: Detecting earthquakes and other rare events from community-based sensors. In Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks, Chicago, IL, USA, 12–14 April 2011; pp. 121–122.

48. Mehrazarin, S.; Tang, B.; Leyba, K.; Han, J.; Beheshti, M. A MacBook based earthquake early warning system. In Proceedings of the 2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), San Francisco, CA, USA, 10–14 April 2016; 2016, pp. 1029–1030. [CrossRef]

49. Chung, A.I.; Lawrence, J.F.; Christensen, C. Evaluating the integrability of the quake-catcher network (QCN). In Proceedings of the 10th International Conference on Information Systems for Crisis Response and Management, Baden-Baden, Germany, 12–15 May 2013; pp. 386–390.

50. Cochran, E.S.; Lawrence, J.F.; Kaiser, A.; Fry, B.; Chung, A.; Christensen, C. Comparison between low-cost and traditional MEMS accelerometers: A case study from the M7.1 Darfield, New Zealand, aftershock deployment. *Ann. Geophys.* **2012**, *54*, 728–737. [CrossRef]

51. Vizuete, A.M.Z.; Llopis, I.P.; Palau, C.; Domingo, M.E. Distributed Sensor System for Earthquake Early Warning Based on the Massive Use of Low Cost Accelerometers. *IEEE Lat. Am. Trans.* **2015**, *13*, 291–298. [CrossRef]

52. Minson, S.E.; Brooks, B.A.; Glennie, C.L.; Murray, J.R.; Langbein, J.O.; Owen, S.E.; Heaton, T.H.; Iannucci, R.A.; Hauser, D.L. Crowdsourced earthquake early warning. *Sci. Adv.* **2015**, *1*, e1500036. [CrossRef]
53. Allen, R.M.; Kong, Q.; Martin-Short, R. The MyShake Platform: A Global Vision for Earthquake Early Warning. *Pure Appl. Geophys.* **2020**, *177*, 1699–1712. [CrossRef]
54. Panizzi, E. The SeismoCloud App: Your smartphone as a seismometer. In Proceedings of the Workshop on Advanced Visual Interfaces AVI, Bari, Italy, 7–10 June 2016; pp. 336–337. [CrossRef]
55. Anderson, R. Early Warning for Geological Disasters: Scientific Methods and Current Practice. *Environ. Eng. Geosci.* **2014**, *20*, 404. [CrossRef]
56. Lee, J.; Khan, I.; Choi, S.; Kwon, Y.-W. A Smart IoT Device for Detecting and Responding to Earthquakes. *Electronics* **2019**, *8*, 1546. [CrossRef]
57. Firouzi, F.; Farahani, B.; Marinšek, A. The convergence and interplay of edge, fog, and cloud in the AI-driven Internet of Things (IoT). *Inf. Syst.* **2021**, 101840. [CrossRef]
58. Santamaria, A.F.; Raimondo, P.; Tropea, M.; De Rango, F.; Aiello, C. An IoT Surveillance System Based on a Decentralised Architecture. *Sensors* **2019**, *19*, 1469. [CrossRef]
59. Böse, M.; Allen, R.; Brown, H.; Gua, G.; Fischer, M.; Hauksson, E.; Heaten, T.; Hellweg, M.; Liukis, M.; Neuhauser, D.; et al. CISN ShakeAlert: An Earthquake Early Warning Demonstration System for California. In *Early Warning for Geological Disasters*; Wenzel, F., Zschau J, Eds.; Springer: Berlin/Heidelberg, Germany, 2014; pp. 49–69.
60. Kobayashi, M. Experience of infrastructure damage caused by the Great East Japan Earthquake and countermeasures against future disasters. *IEEE Commun. Mag.* **2014**, *52*, 23–29. [CrossRef]
61. Prasanna, R.; Yang, L.; King, M.; Huggins, T.J. Information systems architecture for fire emergency response. *J. Enterp.* **2017**, *30*, 605–624. [CrossRef]
62. Lewis, M. *Comparing, Designing, and Deploying VPNs (Networking Technology)*; Cisco Press: Indianapolis, IN, USA, 2006.
63. Hauser, F.; Haberle, M.; Schmidt, M.; Menth, M. P4-IPsec: Site-to-Site and Host-to-Site VPN With IPsec in P4-Based SDN. *IEEE Access* **2020**, *8*, 139567–139586. [CrossRef]
64. Sadasivam, S. Multiprotocol Label Switching (MPLS). Available online: https://medium.com/@senthilsivam/mpls-39f3f0a778aa (accessed on 25 June 2021).
65. Rafique, W.; Qi, L.; Yaqoob, I.; Imran, M.; Rasool, R.U.; Dou, W. Complementing IoT Services Through Software Defined Networking and Edge Computing: A Comprehensive Survey. *IEEE Commun. Surv. Tutorials* **2020**, *22*, 1761–1804. [CrossRef]
66. Gabriel, D.; Barmettler, L. SD-WAN Topology Viewer. Term Project. Ph.D. Thesis, Eastern Switzerland University of Applied Sciences, St. Gallen, Switzerland, 18 December 2020.
67. Ford, B.; Srisuresh, P.; Kegel, D. Peer-to-peer communication across network address translators. In Proceedings of the USENIX Annual Technical Conference, Anaheim, CA, USA, 10–15 April 2005; pp. 179–192.
68. Goethals, T.; Kerkhove, D.; Volckaert, B.; De Turck, F. Scalability evaluation of VPN technologies for secure container networking. In Proceedings of the 15th International Conference on Network and Service Management (CNSM), Halifax, NS, Canada, 21–25 October 2019. [CrossRef]
69. Ierymenko, A.; Limberg, G.; Henry, J. Protocol design whitepaper: ZeroTier documentation. Available online: https://docs.zerotier.com/zerotier/manual/ (accessed on 18 December 2021).
70. Damiano, L. CSI Home Page: General. Available online: https://www.csi.net.nz/ (accessed on 20 June 2021).
71. Kodera, Y.; Yamada, Y.; Hirano, K.; Tamaribuchi, K.; Adachi, S.; Hayashimoto, N.; Morimoto, M.; Nakamura, M.; Hoshiba, M. The Propagation of Local Undamped Motion (PLUM) Method: A Simple and Robust Seismic Wavefield Estimation Approach for Earthquake Early Warning. *Bull. Seism. Soc. Am.* **2018**, *108*, 983–1003. [CrossRef]
72. Wu, Y.-M. Rapid Assessment of Damage Potential of Earthquakes in Taiwan from the Beginning of P Waves. *Bull. Seism. Soc. Am.* **2005**, *95*, 1181–1185. [CrossRef]
73. Hoshiba, M.; Iwakiri, K.; Hayashimoto, N.; Shimoyama, T.; Hirano, K.; Yamada, Y.; Ishigaki, Y.; Kikuta, H. Outline of the 2011 off the Pacific coast of Tohoku Earthquake (Mw 9.0)—Earthquake Early Warning and observed seismic intensity? *Earth, Planets Space* **2011**, *63*, 547–551. [CrossRef]
74. Cochran, E.S.; Bunn, J.; Minson, S.E.; Baltay, A.S.; Kilb, D.; Kodera, Y.; Hoshiba, M. Event Detection Performance of the PLUM Earthquake Early Warning Algorithm in Southern California. *Bull. Seism. Soc. Am.* **2019**, *109*, 1524–1541. [CrossRef]
75. Denis, M.; Zena, C.; Hayajneh, T. Penetration testing: Concepts, attack methods, and defense strategies. In Proceedings of the 2016 IEEE Long Island Systems, Applications and Technology Conference (LISAT), Farmingdale, NY, USA, 29 April 2016; pp. 1–6. [CrossRef]
76. Cao, P.M.; Wu, Y.; Banerjee, S.S.; Azoff, J.; Withers, A.; Kalbarczyk, Z.T.; Iyer, R.K. Caudit: Continuous auditing of SSH servers to mitigate brute-force attacks. In Proceedings of the 16th USENIX Conference on Networked Systems Design and Implementation, Boston, MA, USA, 26–28 February 2019; pp. 667–682.
77. Provos, N.; Honeyman, P. ScanSSH—Scanning the internet for SSH servers. In Proceedings of the LISA 15th Systems Administration Conference, San Diego, CA, USA, 2–7 December 2001; pp. 25–30.
78. Ndichu, S.; McOyowo, S.; Okoyo, H.; Wekesa, C. A Remote Access Security Model based on Vulnerability Management. *Int. J. Inf. Technol. Comput. Sci.* **2020**, *12*, 38–51. [CrossRef]
79. Rayarapu, A.; Krishna, N.V.; Mundhra, D. Securing Files Using AES Algorithm. *Int. J. Comput. Sci. Inf. Technol.* **2013**, *4*, 433–435.

80. Wang, Y.; Yang, J. Ethical Hacking and Network Defense: Choose Your Best Network Vulnerability Scanning Tool. In Proceedings of the 2017 31st International Conference on Advanced Information Networking and Applications Workshops (WAINA), Taipei, Taiwan, 27–29 March 2017; pp. 110–113. [CrossRef]

81. Wack, J.P.; Tracy, M.C.; Souppaya, M.P. Guideline on network security testing. In *Guideline on Network Security Testing*; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2003. [CrossRef]

82. Kilb, D.; Bunn, J.J.; Saunders, J.K.; Cochran, E.S.; Minson, S.E.; Baltay, A.; O'Rourke, C.T.; Hoshiba, M.; Kodera, Y. The PLUM Earthquake Early Warning Algorithm: A Retrospective Case Study of West Coast, USA, Data. *J. Geophys. Res. Solid Earth* **2021**, *126*, e2020JB021053. [CrossRef]

83. Firdous, S.N.; Baig, Z.; Valli, C.; Ibrahim, A. Modelling and Evaluation of Malicious Attacks against the IoT MQTT Protocol. In Proceedings of the 2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Exeter, UK, 21–23 June 2017; pp. 748–755. [CrossRef]

84. Yassein, M.B.; Shatnawi, M.Q.; Aljwarneh, S.; Al-Hatmi, R. Internet of Things: Survey and open issues of MQTT protocol. In Proceedings of the 2017 International Conference on Engineering & MIS (ICEMIS), Monastir, Tunisia, 8–10 May 2017. [CrossRef]

85. Kumar, S.; Rai, S. Survey on Transport Layer Protocols: TCP & UDP. *Int. J. Comput. Appl.* **2012**, *46*, 7.

86. Sadeq, A.S.; Hassan, R.; Al-Rawi, S.S.; Jubair, A.M.; Aman, A.H.M. A Qos Approach For Internet Of Things (Iot) Environment Using Mqtt Protocol. In Proceedings of the 2019 International Conference on Cybersecurity (ICoCSec), Negeri Sembilan, Malaysia, 25–26 September 2019; pp. 59–63. [CrossRef]

87. Bossu, R.; Finazzi, F.; Steed, R.; Fallou, L.; Bondár, I. Shaking in 5 seconds!' A Voluntary Smartphone-based Earthquake Early Warning System. *Seismol. Soc. Am.* **2022**, *93*, 137–148.